

# El-Hajjam - Application de Gestion de Salon de Coiffure

Cahier des Charges

**Abderrahmane AHLALLAY**

Développeur MERN stack

14 Novembre 2025

---

## Table des matières

---

<b>1</b>	<b>Informations du projet</b>	<b>7</b>
<b>2</b>	<b>Objectif du projet</b>	<b>7</b>
2.1	Scope du Projet . . . . .	7
2.2	Problématique . . . . .	7
<b>3</b>	<b>Compétences visées</b>	<b>8</b>
3.1	Gestion de projet . . . . .	8
3.2	Analyse et résolution de problèmes . . . . .	8
3.3	Communication et partage . . . . .	8
3.4	Environnement professionnel . . . . .	8
3.5	Développement technique . . . . .	8
<b>4</b>	<b>Getting Started</b>	<b>9</b>
4.1	Prérequis . . . . .	9
4.2	Installation . . . . .	9
4.3	Commandes disponibles . . . . .	10
<b>5</b>	<b>Spécifications fonctionnelles</b>	<b>10</b>
5.1	Légende des Acteurs . . . . .	10
5.2	Module 1 : Authentification et Gestion des Utilisateurs . . . . .	10
5.2.1	US-001 : Authentification et Connexion Coiffeur . . . . .	10
5.2.2	US-002 : Création de Compte Coiffeur par l'Admin . . . . .	10
5.2.3	US-003 : Modification des Informations Personnelles d'un Coiffeur . . . . .	11
5.2.4	US-004 : Attribution et Gestion des Priviléges Coiffeur . . . . .	11
5.3	Module 2 : Gestion Clientèle . . . . .	11
5.3.1	Fonctionnalités Administrateur . . . . .	11
5.3.2	Fonctionnalités Coiffeur . . . . .	11
5.4	Module 3 : Gestion des Rendez-vous . . . . .	11
5.4.1	Fonctionnalités Administrateur . . . . .	12
5.4.2	Fonctionnalités Coiffeur . . . . .	12
5.5	Module 4 : Gestion du Personnel . . . . .	12
5.5.1	Fonctionnalités Administrateur . . . . .	13
5.6	Module 5 : Services et Prestations . . . . .	13
5.6.1	Fonctionnalités Administrateur . . . . .	13
5.7	Module 6 : Gestion de Stock . . . . .	13
5.7.1	Matières premières . . . . .	13
5.7.2	Produits finis . . . . .	14
5.8	Module 7 : Facturation et Paiements . . . . .	14
5.8.1	Fonctionnalités Administrateur . . . . .	14
5.9	Module 8 : Reporting et Analytics . . . . .	14
5.9.1	Fonctionnalités Administrateur . . . . .	14
5.10	Module 9 : Administration et Audit . . . . .	15
5.10.1	Fonctionnalités Administrateur . . . . .	15
5.11	Module 10 : Gestion des Congés . . . . .	15
5.11.1	Fonctionnalités Coiffeur . . . . .	15
5.11.2	Fonctionnalités Administrateur . . . . .	15

<b>6 Priorisation des Fonctionnalités</b>	<b>16</b>
6.1 Must Have (Critique - MVP) . . . . .	16
6.2 Should Have (Important) . . . . .	17
6.3 Could Have (Souhaitable) . . . . .	18
<b>7 Critères d'Acceptation</b>	<b>18</b>
7.1 Module 1 : Authentification et Gestion des Utilisateurs . . . . .	18
7.1.1 US-001 : Authentification et Connexion Coiffeur . . . . .	18
7.1.2 US-002 : Création de Compte Coiffeur par l'Admin . . . . .	19
7.1.3 US-003 : Modification des Informations Personnelles d'un Coiffeur . . . . .	19
7.1.4 US-004 : Attribution et Gestion des Priviléges Coiffeur . . . . .	19
7.2 Module 2 : Gestion Clientèle . . . . .	19
7.2.1 Ajouter un nouveau client . . . . .	19
7.2.2 Rechercher un client . . . . .	19
7.2.3 Voir l'historique des rendez-vous d'un client . . . . .	20
7.3 Module 3 : Gestion des Rendez-vous . . . . .	20
7.3.1 Créer un nouveau rendez-vous . . . . .	20
7.3.2 Voir le calendrier des rendez-vous . . . . .	20
7.4 Module 4 : Gestion du Personnel . . . . .	20
7.4.1 Créer le compte d'un membre du personnel . . . . .	20
7.4.2 Gérer les horaires de travail . . . . .	20
7.5 Module 5 : Services et Prestations . . . . .	21
7.5.1 Créer un nouveau service . . . . .	21
7.5.2 Mettre à jour les informations d'un service . . . . .	21
7.6 Module 6 : Gestion de Stock . . . . .	21
7.6.1 Ajouter une matière première . . . . .	21
7.6.2 Enregistrer une entrée de stock . . . . .	21
7.7 Module 7 : Facturation et Paiements . . . . .	21
7.7.1 Créer une facture pour un rendez-vous . . . . .	21
7.7.2 Enregistrer un paiement . . . . .	22
7.8 Module 8 : Reporting et Analytics . . . . .	22
7.8.1 Voir les statistiques de fréquentation . . . . .	22
7.8.2 Voir les services les plus demandés . . . . .	22
7.9 Module 9 : Administration et Audit . . . . .	22
7.9.1 Voir le journal des opérations . . . . .	22
7.10 Module 10 : Gestion des Congés . . . . .	22
7.10.1 Demander un congé (Coiffeur) . . . . .	22
7.10.2 Approuver une demande de congé (Admin) . . . . .	23
<b>8 Modèle de Données</b>	<b>23</b>
8.1 Schéma de Base de Données MongoDB . . . . .	23
8.1.1 Collection : users (Coiffeurs) . . . . .	23
8.1.2 Collection : operations (Opérations/Prestations) . . . . .	23
8.1.3 Collection : conges (Demandes de Congés) . . . . .	24
8.1.4 Collection : stockAlerts (Avertissements Stock) . . . . .	24
8.1.5 Collection : produits (Produits/Matières Premières) . . . . .	24
8.1.6 Collection : clients (Clients du Salon) . . . . .	25
8.1.7 Collection : rendezvous (Rendez-vous) . . . . .	25
8.1.8 Collection : services (Services et Prestations) . . . . .	26
8.1.9 Collection : factures (Factures) . . . . .	26
8.1.10 Collection : paiements (Paiements) . . . . .	27
8.2 Relations entre Collections . . . . .	27

8.3	Contraintes d'Intégrité . . . . .	27
<b>9</b>	<b>Matrice de Dépendances</b>	<b>28</b>
<b>10</b>	<b>Règles Métier</b>	<b>29</b>
10.1	Règles d'Authentification et Sécurité . . . . .	29
10.1.1	Politique de Mots de Passe . . . . .	29
10.1.2	Gestion des Sessions . . . . .	29
10.1.3	Tentatives de Connexion . . . . .	29
10.2	Règles de Gestion des Opérations . . . . .	30
10.2.1	Validation des Opérations . . . . .	30
10.2.2	Modification et Suppression . . . . .	30
10.2.3	Calcul des Totaux . . . . .	30
10.3	Règles de Gestion des Congés . . . . .	30
10.3.1	Validation des Dates . . . . .	30
10.3.2	Gestion des Conflits . . . . .	30
10.3.3	Statuts et Workflow . . . . .	30
10.4	Règles de Gestion des Stocks . . . . .	30
10.4.1	Seuils d'Alerte . . . . .	30
10.4.2	Validation des Avertissements . . . . .	31
10.5	Règles de Statistiques et Rapports . . . . .	31
10.5.1	Calcul des Statistiques . . . . .	31
10.5.2	Périodes de Calcul . . . . .	31
10.5.3	Export de Rapports . . . . .	31
10.6	Règles de Privilèges . . . . .	31
10.6.1	Attribution des Privilèges . . . . .	31
10.6.2	Hiérarchie des Privilèges . . . . .	31
<b>11</b>	<b>Architecture API REST</b>	<b>32</b>
11.1	Principe Général . . . . .	32
11.2	Documentation API . . . . .	32
11.3	Caractéristiques Générales de l'API . . . . .	32
11.3.1	Architecture . . . . .	32
11.3.2	Standards et Conventions . . . . .	32
11.3.3	Sécurité . . . . .	32
11.3.4	Modules API . . . . .	33
11.4	Livrables API . . . . .	33
<b>12</b>	<b>Exigences de Performance</b>	<b>33</b>
12.1	Temps de Réponse . . . . .	33
12.2	Capacité et Scalabilité . . . . .	33
12.3	Disponibilité . . . . .	34
12.4	Optimisation . . . . .	34
12.4.1	Cache . . . . .	34
12.4.2	Base de données . . . . .	34
12.4.3	Frontend . . . . .	34
12.5	Monitoring . . . . .	34
12.5.1	Métriques à surveiller . . . . .	34
12.5.2	Alertes . . . . .	34

<b>13 Politique de Sécurité</b>	<b>35</b>
13.1 Authentification et Autorisation . . . . .	35
13.1.1 JWT (JSON Web Tokens) . . . . .	35
13.1.2 Hashage des Mots de Passe . . . . .	35
13.1.3 Protection CSRF . . . . .	35
13.1.4 Protection XSS . . . . .	35
13.2 Chiffrement des Données . . . . .	35
13.2.1 Données en Transit . . . . .	35
13.2.2 Données au Repos . . . . .	35
13.3 Conformité RGPD . . . . .	35
13.3.1 Droit à l'Information . . . . .	35
13.3.2 Droit d'Accès . . . . .	36
13.3.3 Droit à l'Oubli . . . . .	36
13.3.4 Portabilité des Données . . . . .	36
13.3.5 Sécurité des Données . . . . .	36
13.4 Gestion des Sessions . . . . .	36
13.4.1 Sécurité des Sessions . . . . .	36
13.4.2 Headers de Sécurité . . . . .	36
13.5 Audit et Logging . . . . .	36
13.5.1 Événements à Logger . . . . .	36
13.5.2 Informations de Log . . . . .	37
13.5.3 Rétention des Logs . . . . .	37
<b>14 Stratégie de Tests</b>	<b>37</b>
14.1 Types de Tests . . . . .	37
14.1.1 Tests Unitaires (Backend) . . . . .	37
14.1.2 Tests Unitaires (Frontend) . . . . .	37
14.1.3 Tests d'Intégration . . . . .	37
14.1.4 Tests End-to-End (E2E) . . . . .	38
14.1.5 Tests de Performance . . . . .	38
14.1.6 Tests de Sécurité . . . . .	38
14.2 Scénarios de Tests Critiques . . . . .	38
14.2.1 US-001 : Authentification . . . . .	38
14.2.2 US-005 : Saisie d'Opérations . . . . .	39
14.2.3 US-009 : Consultation Opérations Admin . . . . .	39
14.3 Données de Test . . . . .	39
14.3.1 Jeu de données de base . . . . .	39
14.3.2 Données de charge . . . . .	39
14.4 Environnements de Test . . . . .	39
14.4.1 Développement . . . . .	39
14.4.2 Staging . . . . .	39
14.4.3 Production . . . . .	40
<b>15 Spécifications UI/UX</b>	<b>40</b>
15.1 Charte Graphique . . . . .	40
15.1.1 Couleurs Principales . . . . .	40
15.1.2 Typographie . . . . .	40
15.2 Responsive Design . . . . .	40
15.2.1 Breakpoints . . . . .	40
15.2.2 Adaptations Mobile . . . . .	40
15.3 Guidelines UX . . . . .	41
15.3.1 Navigation . . . . .	41

15.3.2 Feedback Utilisateur . . . . .	41
15.3.3 Formulaires . . . . .	41
15.3.4 Tableaux et Listes . . . . .	41
15.4 Wireframes Principaux . . . . .	41
15.4.1 Page de Connexion . . . . .	41
15.4.2 Dashboard Coiffeur . . . . .	42
15.4.3 Formulaire Saisie Opération . . . . .	42
15.5 Accessibilité (WCAG 2.1) . . . . .	42
15.5.1 Niveau de Conformité . . . . .	43
15.5.2 Exigences . . . . .	43
<b>16 Technologies et outils</b> . . . . .	<b>43</b>
16.1 Frontend . . . . .	43
16.2 Backend . . . . .	43
16.3 DevOps . . . . .	43
16.4 Qualité . . . . .	44
16.5 Tests . . . . .	44
16.6 Gestion de projet . . . . .	44
16.7 Internationalisation . . . . .	44
<b>17 Ordre de développement recommandé</b> . . . . .	<b>44</b>
<b>18 Écrans à développer</b> . . . . .	<b>45</b>
18.1 Authentification . . . . .	45
18.2 Gestion des Utilisateurs (Admin) . . . . .	45
18.3 Gestion des Opérations (Coiffeur) . . . . .	45
18.4 Statistiques et Rapports . . . . .	45
18.5 Gestion des Congés (Coiffeur) . . . . .	45
18.6 Gestion des Stocks (Coiffeur) . . . . .	45
18.7 Pages système . . . . .	45
<b>19 Modalités pédagogiques</b> . . . . .	<b>46</b>
<b>20 Modalités d'évaluation</b> . . . . .	<b>46</b>
<b>21 Livrables attendus</b> . . . . .	<b>46</b>
21.1 Code source . . . . .	46
21.2 Documentation technique . . . . .	46
21.2.1 Architecture . . . . .	47
21.2.2 Installation et configuration . . . . .	47
21.2.3 API . . . . .	47
21.2.4 Dépendances . . . . .	47
21.3 Dockerisation . . . . .	47
21.4 CI/CD . . . . .	47
21.5 Présentation . . . . .	47
<b>22 Critères de performance</b> . . . . .	<b>47</b>
22.1 Fonctionnalités . . . . .	47
22.2 Authentification et Sécurité . . . . .	48
22.3 Architecture . . . . .	48
22.4 React et Hooks . . . . .	48
22.5 Tests . . . . .	48
22.6 Gestion de projet . . . . .	48

---

22.7 Dockerisation . . . . .	48
22.8 CI/CD . . . . .	49
22.9 Qualité du code . . . . .	49
22.10 Internationalisation . . . . .	49
22.11 Documentation . . . . .	49
<b>23 Glossaire Technique</b>	<b>49</b>
23.1 Termes Généraux . . . . .	49
23.2 Authentification et Sécurité . . . . .	49
23.3 Technologies Frontend . . . . .	50
23.4 Technologies Backend . . . . .	50
23.5 Gestion de Données . . . . .	50
23.6 DevOps et Outils . . . . .	50
23.7 Tests . . . . .	50
23.8 Gestion de Projet . . . . .	51
23.9 Architecture . . . . .	51
23.10 Internationalisation . . . . .	51
<b>24 Notes importantes</b>	<b>51</b>
24.1 Phase Multi-Tenant (Future) . . . . .	51
24.2 Bonnes Pratiques . . . . .	51
24.3 Contraintes Techniques . . . . .	51

## 1 Informations du projet

### Informations Générales

- **Assigné par :** Zakaria Ziane
- **Développé par :** Abderrahmane AHLALLAY
- **Date de création :** 14 Novembre 2025
- **Version du cahier des charges :** Version 1
- **Type de projet :** Développement Fullstack avec MERN Stack
- **Contexte :** Création d'une application MERN Fullstack

## 2 Objectif du projet

### Objectifs Principaux

Développer une application complète de gestion pour un salon de coiffure permettant :

- **Gestion des coiffeurs :** création, modification, attribution de priviléges
- **Suivi des opérations :** saisie et suivi des prestations effectuées par chaque coiffeur
- **Statistiques et rapports :** analyse de la performance du salon et des coiffeurs
- **Phase future :** généralisation multi-tenant (chaque salon peut créer son espace)

L'application doit être **performante, sécurisée, scalable** et préparée pour une évolution vers un modèle multi-tenant.

### 2.1 Scope du Projet

#### Version V1 - Application Interne

Cette première version (V1) est une **application interne de gestion** destinée aux administrateurs et coiffeurs du salon pour gérer efficacement les opérations quotidiennes.

**Scope Futur :** Une interface publique sera développée dans une phase ultérieure (Phase 4), permettant aux clients de consulter les produits et services, commander et prendre des rendez-vous en ligne.

### 2.2 Problématique

Dans le contexte web actuel, on recherche des solutions pour développer des applications web performantes et intuitives. Celles-ci doivent être à la fois simples à administrer et à déployer. Comment peut-on allier efficacement le rendu côté serveur, propre à la logique traditionnelle, à la flexibilité et la réactivité offertes par JavaScript et Node.js ? La stack **MERN** (MongoDB, Express.js, React.js, Node.js) semble être une réponse prometteuse à ce défi.

### 3 Compétences visées

---

Ce projet vise à développer les compétences suivantes (toutes au **Niveau 3**) :

#### 3.1 Gestion de projet

- **C1.** Planifier le travail à effectuer individuellement
- **C2.** Contribuer au pilotage de l'organisation du travail individuel et collectif
- **C13.** Contribuer à la gestion d'un projet

#### 3.2 Analyse et résolution de problèmes

- **C3.** Définir le périmètre d'un problème rencontré en adoptant une démarche inductive
- **C4.** Rechercher de façon méthodique une ou des solutions au problème rencontré

#### 3.3 Communication et partage

- **C5.** Partager la solution adoptée en utilisant les moyens de partage de connaissance ou de documentation disponibles
- **C6.** Présenter un travail réalisé en synthétisant ses résultats, sa démarche
- **C8.** Interagir dans un contexte professionnel de façon respectueuse et constructive
- **C9.** Faciliter un temps de travail collectif en assurant une communication constructive entre les participants

#### 3.4 Environnement professionnel

- **C7.** Se familiariser avec les codes et la culture propres à son environnement professionnel
- **C10.** Installer et configurer son environnement de travail en fonction du projet

#### 3.5 Développement technique

- **C11.** Développer des interfaces utilisateur
- **C12.** Développer des composants métier
- **C14.** Analyser les besoins et maquetter une application
- **C15.** Définir l'architecture logicielle d'une application
- **C16.** Concevoir et mettre en place une base de données
- **C17.** Développer des composants d'accès aux données
- **C18.** Préparer et exécuter les plans de tests
- **C19.** Préparer et documenter le déploiement d'une application
- **C20.** Contribuer à la mise en production dans une démarche DevOps

## 4 Getting Started

---

### 4.1 Prérequis

Avant de commencer, assurez-vous d'avoir installé les outils suivants :

- **Node.js** (version 18.x ou supérieure)
- **npm** ou **yarn** (gestionnaire de paquets)
- **Git** (contrôle de version)
- **Docker** et **Docker Compose** (pour la conteneurisation)
- **MongoDB** (localement ou via MongoDB Atlas)
- **Un éditeur de code** (VS Code recommandé)

### 4.2 Installation

#### 1. Cloner le dépôt

```
1 git clone https://github.com/AHLALLAY/el-hajjam.git
2 cd el-hajjam
```

#### 2. Installer les dépendances

```
1 # Backend
2 cd backend
3 npm install
4
5 # Frontend
6 cd ../frontend
7 npm install
```

#### 3. Configurer les variables d'environnement

```
1 # Backend
2 cp backend/.env.example backend/.env
3 # diter le fichier .env avec vos configurations
4
5 # Frontend
6 cp frontend/.env.example frontend/.env
7 # diter le fichier .env avec vos configurations
```

#### 4. Lancer l'application en mode développement

```
1 # Backend
2 cd backend
3 npm run dev
4
5 # Frontend (dans un autre terminal)
6 cd frontend
7 npm run dev
```

#### 5. Lancer avec Docker

```
1 docker-compose up -d
```

### 4.3 Commandes disponibles

- `npm run dev` : Lance le serveur de développement
- `npm run build` : Compile l'application pour la production
- `npm run start` : Lance l'application en mode production
- `npm run test` : Lance les tests unitaires
- `npm run test:e2e` : Lance les tests end-to-end
- `npm run lint` : Vérifie le code avec ESLint
- `npm run format` : Formate le code avec Prettier

## 5 Spécifications fonctionnelles

### 5.1 Légende des Acteurs

Les acteurs dans les user stories sont identifiés par des couleurs distinctes :

#### Acteur : Coiffeur

**Description :** Utilisateur principal qui effectue les prestations et saisit ses opérations  
**Couleur d'identification :** Bleu

#### Acteur : Administrateur

**Description :** Gestionnaire du salon qui administre les coiffeurs et consulte les rapports  
**Couleur d'identification :** Violet

### 5.2 Module 1 : Authentification et Gestion des Utilisateurs

#### 5.2.1 US-001 : Authentification et Connexion Coiffeur

##### US-001

###### Acteur : Coiffeur

Je veux me connecter avec mon email et mot de passe  
Afin d'accéder à mon espace de travail

#### 5.2.2 US-002 : Création de Compte Coiffeur par l'Admin

##### US-002

###### Acteur : Administrateur

Je veux créer un compte pour un coiffeur avec email et mot de passe  
Afin de permettre au coiffeur d'accéder à l'application

### 5.2.3 US-003 : Modification des Informations Personnelles d'un Coiffeur

**US-003**

**Acteur :** Administrateur

Je veux modifier les informations personnelles d'un coiffeur

Afin de maintenir les données à jour

### 5.2.4 US-004 : Attribution et Gestion des Privilèges Coiffeur

**US-004**

**Acteur :** Administrateur

Je veux donner des privilèges à un coiffeur

Afin de contrôler les permissions d'accès aux fonctionnalités

## 5.3 Module 2 : Gestion Clientèle

### 5.3.1 Fonctionnalités Administrateur

**Module 2 - Gestion Clientèle**

**Acteur :** Administrateur

- Ajouter un nouveau client
- Voir la liste de tous les clients
- Rechercher un client par nom ou numéro de téléphone
- Mettre à jour les informations d'un client
- Voir l'historique complet des rendez-vous d'un client
- Voir les préférences d'un client
- Ajouter des notes sur un client

### 5.3.2 Fonctionnalités Coiffeur

**Module 2 - Gestion Clientèle**

**Acteur :** Coiffeur

- Voir les informations d'un client
- Consulter l'historique des services d'un client
- Ajouter des notes sur un client

## 5.4 Module 3 : Gestion des Rendez-vous

#### 5.4.1 Fonctionnalités Administrateur

##### Module 3 - Gestion des Rendez-vous

###### Acteur : Administrateur

- Créer un nouveau rendez-vous
- Voir tous les rendez-vous du jour
- Voir le calendrier des rendez-vous
- Modifier un rendez-vous existant
- Annuler un rendez-vous
- Confirmer un rendez-vous
- Voir les rendez-vous par coiffeur
- Voir les rendez-vous par client
- Vérifier la disponibilité d'un coiffeur à une heure donnée
- Voir les créneaux disponibles
- Gérer les horaires d'ouverture du salon

#### 5.4.2 Fonctionnalités Coiffeur

##### Module 3 - Gestion des Rendez-vous

###### Acteur : Coiffeur

- Voir mes rendez-vous du jour
- Voir mon planning hebdomadaire
- Voir les détails d'un rendez-vous
- Marquer un rendez-vous comme terminé
- Annuler un rendez-vous

#### 5.5 Module 4 : Gestion du Personnel

### 5.5.1 Fonctionnalités Administrateur

#### Module 4 - Gestion du Personnel

##### Acteur : Administrateur uniquement

- Créer le compte d'un membre du personnel
- Mettre à jour les informations d'un membre du personnel
- Supprimer un membre du personnel
- Voir la liste de tous les membres du personnel
- Gérer les horaires de travail de chaque membre du personnel
- Voir les performances de chaque membre du personnel

## 5.6 Module 5 : Services et Prestations

### 5.6.1 Fonctionnalités Administrateur

#### Module 5 - Services et Prestations

##### Acteur : Administrateur uniquement

- Créer un nouveau service (coupe, coloration, etc.)
- Voir la liste de tous les services disponibles
- Mettre à jour les informations d'un service (nom, durée, prix)
- Supprimer un service
- Définir la durée estimée d'un service
- Définir le prix d'un service

## 5.7 Module 6 : Gestion de Stock

### 5.7.1 Matières premières

#### Module 6 - Matières Premières

##### Acteur : Coiffeur / Administrateur

- Ajouter une matière première
- Voir la liste des matières premières
- Mettre à jour les informations d'une matière première
- Voir le stock disponible de chaque matière première
- Être alerté lorsque le stock d'une matière première est faible
- Enregistrer une entrée de stock (réception)
- Enregistrer une sortie de stock (utilisation)

### 5.7.2 Produits finis

#### Module 6 - Produits Finis

##### Acteur : Coiffeur / Administrateur

- Ajouter un produit au stock
- Voir la liste de tous les produits
- Mettre à jour les informations d'un produit de stock
- Voir le stock disponible de chaque produit
- Être alerté lorsque le stock d'un produit est faible
- Voir l'historique des mouvements de stock

## 5.8 Module 7 : Facturation et Paiements

### 5.8.1 Fonctionnalités Administrateur

#### Module 7 - Facturation et Paiements

##### Acteur : Administrateur uniquement

- Créer une facture pour un rendez-vous
- Voir la liste des factures
- Enregistrer un paiement
- Voir les factures impayées
- Générer un reçu de paiement
- Voir l'historique des paiements d'un client
- Voir les statistiques de revenus par période

## 5.9 Module 8 : Reporting et Analytics

### 5.9.1 Fonctionnalités Administrateur

#### Module 8 - Reporting et Analytics

##### Acteur : Administrateur uniquement

- Voir les statistiques de fréquentation
- Voir les revenus par période (jour, semaine, mois)
- Voir les services les plus demandés
- Voir les performances de chaque coiffeur
- Voir les heures de pointe
- Exporter les rapports en format PDF ou Excel

## 5.10 Module 9 : Administration et Audit

### 5.10.1 Fonctionnalités Administrateur

#### Module 9 - Administration et Audit

##### Acteur : Administrateur uniquement

- Voir le journal des opérations
- Voir le journal des opérations filtré par membre du personnel
- Voir le journal des opérations filtré par membre du personnel et par mois
- Voir les statistiques

## 5.11 Module 10 : Gestion des Congés

### 5.11.1 Fonctionnalités Coiffeur

#### Module 10 - Gestion des Congés

##### Acteur : Coiffeur

- Demander un congé (dates de début et fin, type de congé, motif)
- Voir mes demandes de congé (en attente, approuvées, refusées)
- Voir les détails d'une de mes demandes de congé
- Modifier une demande de congé en attente
- Annuler une demande de congé en attente
- Voir mes congés approuvés dans mon calendrier

### 5.11.2 Fonctionnalités Administrateur

#### Module 10 - Gestion des Congés

##### Acteur : Administrateur uniquement

- Voir toutes les demandes de congé du personnel
- Approuver une demande de congé
- Refuser une demande de congé
- Voir le calendrier des congés de tout le personnel
- Filtrer les congés par membre du personnel ou par période
- Voir les statistiques des congés (congés pris, restants, etc.)

## 6 Priorisation des Fonctionnalités

Les fonctionnalités sont classées selon la méthode MoSCoW (Must Have, Should Have, Could Have, Won't Have) :

### 6.1 Must Have (Critique - MVP)

#### Must Have - Fonctionnalités Critiques

##### Module 1 : Authentification et Sécurité

- Création de compte administrateur
- Connexion (administrateur et coiffeur)
- Réinitialisation du mot de passe

##### Module 2 : Gestion Clientèle (base)

- Ajouter un nouveau client
- Voir la liste de tous les clients
- Rechercher un client

##### Module 3 : Gestion des Rendez-vous (base)

- Créer un nouveau rendez-vous
- Voir tous les rendez-vous du jour
- Voir le calendrier des rendez-vous

## 6.2 Should Have (Important)

### Should Have - Fonctionnalités Importantes

#### Module 1 : Authentification

- Modification des informations personnelles
- Attribution et gestion des privilèges

#### Module 2 : Gestion Clientèle

- Mettre à jour les informations d'un client
- Voir l'historique des rendez-vous d'un client
- Ajouter des notes sur un client

#### Module 3 : Gestion des Rendez-vous

- Modifier un rendez-vous existant
- Annuler un rendez-vous
- Vérifier la disponibilité d'un coiffeur

#### Module 4 : Gestion du Personnel

- Créer le compte d'un membre du personnel
- Voir la liste de tous les membres du personnel
- Gérer les horaires de travail

#### Module 5 : Services et Prestations

- Créer un nouveau service
- Voir la liste de tous les services disponibles
- Mettre à jour les informations d'un service

#### Module 7 : Facturation et Paiements (base)

- Créer une facture pour un rendez-vous
- Enregistrer un paiement
- Voir la liste des factures

### 6.3 Could Have (Souhaitable)

Could Have - Fonctionnalités Souhaitables
<b>Module 6 : Gestion de Stock</b> <ul style="list-style-type: none"><li>— Gestion complète des stocks (matières premières et produits finis)</li><li>— Alertes automatiques de stock faible</li><li>— Historique des mouvements de stock</li></ul>
<b>Module 8 : Reporting et Analytics</b> <ul style="list-style-type: none"><li>— Statistiques de fréquentation</li><li>— Services les plus demandés</li><li>— Heures de pointe</li><li>— Export des rapports en PDF/Excel</li></ul>
<b>Module 9 : Administration et Audit</b> <ul style="list-style-type: none"><li>— Journal des opérations</li><li>— Filtrage par membre du personnel</li><li>— Statistiques d'audit</li></ul>
<b>Module 10 : Gestion des Congés</b> <ul style="list-style-type: none"><li>— Demande de congés par les coiffeurs</li><li>— Approbation/refus par l'admin</li><li>— Calendrier des congés</li><li>— Statistiques des congés</li></ul>
<b>Fonctionnalités avancées</b> <ul style="list-style-type: none"><li>— Envoi de rapports par email</li><li>— Comparaison avec les périodes précédentes</li><li>— Historique des modifications des profils</li><li>— Notifications en temps réel</li></ul>

## 7 Critères d'Acceptation

### 7.1 Module 1 : Authentification et Gestion des Utilisateurs

#### 7.1.1 US-001 : Authentification et Connexion Coiffeur

- Le coiffeur peut saisir son email et mot de passe sur une page de connexion
- En cas d'erreur (email invalide, mot de passe incorrect), un message clair est affiché
- Après connexion réussie, le coiffeur est redirigé vers son dashboard personnel
- La session reste active pendant une durée définie (ex : 24h)

- Possibilité de cocher "Se souvenir de moi" pour prolonger la session
- En cas de mot de passe oublié, possibilité de réinitialisation via email

#### 7.1.2 US-002 : Création de Compte Coiffeur par l'Admin

- L'admin accède à un formulaire de création de compte coiffeur
- L'admin peut saisir : nom, prénom, email, mot de passe, nombre de salon (1 par défaut), téléphone (optionnel)
- L'email doit être valide et unique dans le système
- Le mot de passe respecte les critères de sécurité (minimum 8 caractères, majuscule, chiffre)
- Un email de bienvenue avec les identifiants est envoyé au coiffeur
- Le compte est créé avec le statut "actif" par défaut
- L'admin peut voir la liste des coiffeurs créés

#### 7.1.3 US-003 : Modification des Informations Personnelles d'un Coiffeur

- L'admin peut accéder au profil d'un coiffeur depuis la liste
- L'admin peut modifier : nom, prénom, email, téléphone, adresse, date de naissance
- L'email modifié doit être valide et unique (si différent de l'ancien)
- Les modifications sont sauvegardées après confirmation
- Un message de succès confirme la mise à jour
- Les modifications sont visibles immédiatement dans le profil

#### 7.1.4 US-004 : Attribution et Gestion des Privilèges Coiffeur

- L'admin peut accéder à la gestion des privilèges d'un coiffeur
- L'admin peut attribuer différents types de privilèges (gestion caisse, clients, rapports, etc.)
- Les privilèges peuvent être activés/désactivés via des cases à cocher
- Les privilèges sont appliqués immédiatement après sauvegarde
- Le coiffeur voit uniquement les fonctionnalités autorisées dans son interface

### 7.2 Module 2 : Gestion Clientèle

#### 7.2.1 Ajouter un nouveau client

- L'admin peut accéder à un formulaire de création de client
- L'admin peut saisir : nom, prénom, téléphone, email (optionnel), adresse (optionnel)
- Le téléphone doit être valide et unique dans le système
- Le client est créé et visible immédiatement dans la liste
- Un message de confirmation confirme la création

#### 7.2.2 Rechercher un client

- L'admin peut rechercher un client par nom ou numéro de téléphone
- La recherche fonctionne en temps réel pendant la saisie

- Les résultats sont affichés instantanément
- Un message s'affiche si aucun résultat n'est trouvé

### 7.2.3 Voir l'historique des rendez-vous d'un client

- L'admin peut accéder à l'historique complet depuis le profil du client
- Tous les rendez-vous du client sont affichés par ordre chronologique
- Chaque rendez-vous affiche : date, heure, coiffeur, service, statut
- Possibilité de filtrer par période

## 7.3 Module 3 : Gestion des Rendez-vous

### 7.3.1 Crée un nouveau rendez-vous

- L'admin peut accéder à un formulaire de création de rendez-vous
- L'admin peut sélectionner : client, coiffeur, service, date, heure
- Le système vérifie automatiquement la disponibilité du coiffeur
- Un message d'erreur s'affiche si le créneau n'est pas disponible
- Le rendez-vous est créé et visible dans le calendrier
- Le client et le coiffeur reçoivent une notification

### 7.3.2 Voir le calendrier des rendez-vous

- L'admin peut voir tous les rendez-vous dans une vue calendrier
- Possibilité de filtrer par coiffeur, par jour, semaine ou mois
- Les rendez-vous sont affichés avec les informations essentielles
- Clic sur un rendez-vous pour voir les détails complets

## 7.4 Module 4 : Gestion du Personnel

### 7.4.1 Crée le compte d'un membre du personnel

- L'admin peut créer un compte avec : nom, prénom, email, mot de passe, téléphone
- L'email doit être valide et unique
- Le compte est créé avec le rôle "coiffeur" par défaut
- Un email de bienvenue est envoyé avec les identifiants
- Le nouveau membre apparaît dans la liste du personnel

### 7.4.2 Gérer les horaires de travail

- L'admin peut définir les horaires de travail pour chaque membre
- Possibilité de définir des horaires différents par jour de la semaine
- Les horaires sont pris en compte pour la vérification de disponibilité
- Les modifications sont appliquées immédiatement

## 7.5 Module 5 : Services et Prestations

### 7.5.1 Créer un nouveau service

- L'admin peut créer un service avec : nom, durée estimée, prix
- Le nom du service doit être unique
- La durée est exprimée en minutes
- Le prix doit être supérieur à 0
- Le service est disponible immédiatement pour la création de rendez-vous

### 7.5.2 Mettre à jour les informations d'un service

- L'admin peut modifier tous les champs d'un service existant
- Les modifications sont répercutées sur les futurs rendez-vous
- Les rendez-vous passés conservent les anciennes informations
- Un message de confirmation confirme la mise à jour

## 7.6 Module 6 : Gestion de Stock

### 7.6.1 Ajouter une matière première

- L'admin peut ajouter une matière première avec : nom, catégorie, quantité, unité, seuil d'alerte
- Le nom doit être unique
- La quantité initiale peut être définie lors de la création
- Le seuil d'alerte permet de déclencher des notifications automatiques
- La matière première apparaît dans la liste du stock

### 7.6.2 Enregistrer une entrée de stock

- L'admin peut enregistrer une réception de stock
- Saisie de : produit, quantité, date de réception, fournisseur (optionnel)
- La quantité est ajoutée au stock existant
- L'historique des mouvements est enregistré
- Un message de confirmation confirme l'enregistrement

## 7.7 Module 7 : Facturation et Paiements

### 7.7.1 Créeer une facture pour un rendez-vous

- L'admin peut créer une facture depuis un rendez-vous terminé
- La facture inclut automatiquement : client, service, prix, date
- Possibilité d'ajouter des services supplémentaires
- Le total est calculé automatiquement
- La facture est générée avec un numéro unique

- Un PDF de la facture peut être téléchargé

### 7.7.2 Enregistrer un paiement

- L'admin peut enregistrer un paiement pour une facture
- Saisie de : montant, mode de paiement, date
- Le montant peut être partiel ou total
- Le statut de la facture est mis à jour automatiquement
- Un reçu peut être généré après paiement

## 7.8 Module 8 : Reporting et Analytics

### 7.8.1 Voir les statistiques de fréquentation

- L'admin peut voir les statistiques de fréquentation du salon
- Affichage par période : jour, semaine, mois, année
- Graphiques visuels (courbes, barres) pour l'évolution
- Comparaison avec les périodes précédentes
- Export possible en PDF ou Excel

### 7.8.2 Voir les services les plus demandés

- L'admin peut voir un classement des services par popularité
- Affichage du nombre de rendez-vous par service
- Graphique en camembert pour visualisation
- Filtrage par période possible

## 7.9 Module 9 : Administration et Audit

### 7.9.1 Voir le journal des opérations

- L'admin peut consulter le journal de toutes les opérations
- Affichage de : utilisateur, action, date/heure, détails
- Possibilité de filtrer par membre du personnel
- Possibilité de filtrer par période (mois)
- Export possible pour archivage

## 7.10 Module 10 : Gestion des Congés

### 7.10.1 Demander un congé (Coiffeur)

- Le coiffeur peut créer une demande de congé
- Saisie de : type (régulier/exceptionnel), dates de début et fin, motif
- Validation des dates (début < fin, pas de dates passées)
- Vérification automatique des conflits avec autres congés

- La demande est enregistrée avec le statut "en attente"
- L'admin reçoit une notification

### 7.10.2 Approuver une demande de congé (Admin)

- L'admin peut voir toutes les demandes de congé en attente
- L'admin peut approuver ou refuser une demande
- Possibilité d'ajouter un commentaire lors de la décision
- Le coiffeur reçoit une notification du changement de statut
- Les congés approuvés apparaissent dans le calendrier

## 8 Modèle de Données

### 8.1 Schéma de Base de Données MongoDB

#### 8.1.1 Collection : users (Coiffeurs)

```

1  {
2      "_id": "ObjectId",
3      "email": "string (unique, required, indexed)",
4      "password": "string (hashed, required)",
5      "firstName": "string (required)",
6      "lastName": "string (required)",
7      "phone": "string (optional)",
8      "address": "string (optional)",
9      "dateOfBirth": "Date (optional)",
10     "role": "string (enum: ['coiffeur', 'admin'], default: 'coiffeur')",
11     "status": "string (enum: ['actif', 'inactif', 'suspendu'], default: 'actif')",
12     "privileges": {
13         "gestionCaisse": "boolean (default: false)",
14         "gestionClients": "boolean (default: false)",
15         "gestionRapports": "boolean (default: false)",
16         "gestionStocks": "boolean (default: false)"
17     },
18     "createdAt": "Date (default: Date.now)",
19     "updatedAt": "Date (default: Date.now)",
20     "lastLogin": "Date (optional)",
21     "salonId": "ObjectId (optional, pour multi-tenant)"
22 }
```

#### Index recommandés :

- Index unique sur `email`
- Index sur `role` et `status` pour les requêtes de filtrage
- Index sur `salonId` pour le multi-tenant

#### 8.1.2 Collection : operations (Opérations/Prestations)

```

1  {
2      "_id": "ObjectId",
3      "coiffeurId": "ObjectId (required, ref: users, indexed)",
```

```

4   "typeTravail": "string (required, enum: ['coupe', 'coloration', 'brushing',
5     'soin', 'autre'])",
6   "tarif": "number (required, min: 0)",
7   "pourboire": "number (default: 0, min: 0)",
8   "total": "number (calculated: tarif + pourboire)",
9   "dateOperation": "Date (required, indexed)",
10  "notes": "string (optional, maxLength: 500)",
11  "createdAt": "Date (default: Date.now)",
12  "updatedAt": "Date (default: Date.now)",
13  "salonId": "ObjectId (optional, pour multi-tenant)"
14 }

```

#### Index recommandés :

- Index composé sur coiffeurId et dateOperation (desc) pour les requêtes d'historique
- Index sur dateOperation pour les filtres par période
- Index sur typeTravail pour les statistiques

#### 8.1.3 Collection : conges (Demandes de Congés)

```

1 {
2   "_id": "ObjectId",
3   "coiffeurId": "ObjectId (required, ref: users, indexed)",
4   "type": "string (required, enum: ['regulier', 'exceptionnel'])",
5   "dateDebut": "Date (required, indexed)",
6   "dateFin": "Date (required, indexed)",
7   "motif": "string (optional, maxLength: 500)",
8   "statut": "string (enum: ['en_attente', 'approuve', 'refuse'], default:
9     'en_attente')",
10  "approuvePar": "ObjectId (optional, ref: users)",
11  "dateApprobation": "Date (optional)",
12  "commentaireAdmin": "string (optional)",
13  "createdAt": "Date (default: Date.now)",
14  "updatedAt": "Date (default: Date.now)",
15  "salonId": "ObjectId (optional, pour multi-tenant)"
}

```

#### 8.1.4 Collection : stockAlerts (Avertissements Stock)

```

1 {
2   "_id": "ObjectId",
3   "coiffeurId": "ObjectId (required, ref: users, indexed)",
4   "produitId": "ObjectId (optional, ref: produits)",
5   "nomProduit": "string (required)",
6   "niveauStock": "string (required, enum: ['epuise', 'presque_epuise'])",
7   "notes": "string (optional, maxLength: 500)",
8   "statut": "string (enum: ['nouveau', 'en_cours', 'resolu'], default:
9     'nouveau')",
10  "resoluPar": "ObjectId (optional, ref: users)",
11  "dateResolution": "Date (optional)",
12  "createdAt": "Date (default: Date.now)",
13  "updatedAt": "Date (default: Date.now)",
14  "salonId": "ObjectId (optional, pour multi-tenant)"
}

```

#### 8.1.5 Collection : produits (Produits/Matières Premières)

```

1 {

```

```

2   "_id": "ObjectId",
3   "nom": "string (required, unique, indexed)",
4   "categorie": "string (required, enum: ['shampooing', 'coloration', 'soin',
5     'accessoire', 'autre'])",
6   "quantiteStock": "number (default: 0, min: 0)",
7   "seuilAlerte": "number (default: 5, min: 0)",
8   "unite": "string (required, enum: ['unite', 'litre', 'kg', 'ml'])",
9   "fournisseur": "string (optional)",
10  "prixUnitaire": "number (optional, min: 0)",
11  "createdAt": "Date (default: Date.now)",
12  "updatedAt": "Date (default: Date.now)",
13  "salonId": "ObjectId (optional, pour multi-tenant)"
14 }
```

**Index recommandés :**

- Index unique sur nom
- Index sur categorie pour les filtres
- Index sur quantiteStock pour les alertes automatiques

**8.1.6 Collection : clients (Clients du Salon)**

```

1 {
2   "_id": "ObjectId",
3   "firstName": "string (required)",
4   "lastName": "string (required)",
5   "phone": "string (required, unique, indexed)",
6   "email": "string (optional)",
7   "address": "string (optional)",
8   "preferences": {
9     "coiffeurPrefere": "ObjectId (optional, ref: users)",
10    "servicesPrefere": ["ObjectId (optional, ref: services)"],
11    "notes": "string (optional, maxLength: 1000)"
12  },
13  "notes": ["string (optional)"],
14  "createdAt": "Date (default: Date.now)",
15  "updatedAt": "Date (default: Date.now)",
16  "salonId": "ObjectId (optional, pour multi-tenant)"
17 }
```

**Index recommandés :**

- Index unique sur phone
- Index sur lastName et firstName pour la recherche
- Index sur email si présent

**8.1.7 Collection : rendezvous (Rendez-vous)**

```

1 {
2   "_id": "ObjectId",
3   "clientId": "ObjectId (required, ref: clients, indexed)",
4   "coiffeurId": "ObjectId (required, ref: users, indexed)",
5   "serviceId": "ObjectId (required, ref: services, indexed)",
6   "dateRendezVous": "Date (required, indexed)",
7   "heureDebut": "string (required, format: 'HH:mm')",
8   "heureFin": "string (required, format: 'HH:mm')",
9   "statut": "string (enum: ['confirme', 'en_attente', 'annule', 'termine']),
10    default: 'en_attente')",
11   "notes": "string (optional, maxLength: 500)",
```

```

11   "createdAt": "Date (default: Date.now)",
12   "updatedAt": "Date (default: Date.now)",
13   "salonId": "ObjectId (optional, pour multi-tenant)"
14 }
```

#### Index recommandés :

- Index composé sur coiffeurId et dateRendezVous
- Index composé sur clientId et dateRendezVous
- Index sur statut pour les filtres
- Index sur dateRendezVous pour les requêtes par période

### 8.1.8 Collection : services (Services et Prestations)

```

1 {
2   "_id": "ObjectId",
3   "nom": "string (required, unique, indexed)",
4   "description": "string (optional, maxLength: 500)",
5   "dureeEstimee": "number (required, min: 1, unit: minutes)",
6   "prix": "number (required, min: 0)",
7   "categorie": "string (required, enum: ['coupe', 'coloration', 'brushing',
8     'soin', 'autre'])",
9   "actif": "boolean (default: true)",
10  "createdAt": "Date (default: Date.now)",
11  "updatedAt": "Date (default: Date.now)",
12  "salonId": "ObjectId (optional, pour multi-tenant)"
```

#### Index recommandés :

- Index unique sur nom
- Index sur categorie pour les filtres
- Index sur actif pour les requêtes de services disponibles

### 8.1.9 Collection : factures (Factures)

```

1 {
2   "_id": "ObjectId",
3   "numeroFacture": "string (required, unique, indexed)",
4   "clientId": "ObjectId (required, ref: clients, indexed)",
5   "rendezVousId": "ObjectId (required, ref: rendezvous, indexed)",
6   "services": [
7     {
8       "serviceId": "ObjectId (required, ref: services)",
9       "nom": "string (required)",
10      "prix": "number (required, min: 0)",
11      "quantite": "number (default: 1, min: 1)"
12    },
13    "sousTotal": "number (required, min: 0)",
14    "remise": "number (default: 0, min: 0)",
15    "total": "number (required, calculated: sousTotal - remise)",
16    "statut": "string (enum: ['impayee', 'partiellement_payee', 'payee'],
17      default: 'impayee')",
18    "dateEmission": "Date (required, default: Date.now)",
19    "dateEcheance": "Date (optional)",
20    "createdAt": "Date (default: Date.now)",
21    "updatedAt": "Date (default: Date.now)",
22    "salonId": "ObjectId (optional, pour multi-tenant)"
```

**Index recommandés :**

- Index unique sur `numeroFacture`
- Index sur `clientId` et `dateEmission` pour l'historique
- Index sur `statut` pour les filtres de factures impayées
- Index sur `rendezVousId` pour la liaison avec les rendez-vous

**8.1.10 Collection : paiements (Paiements)**

```

1 {
2   "_id": "ObjectId",
3   "factureId": "ObjectId (required, ref: factures, indexed)",
4   "montant": "number (required, min: 0)",
5   "modePaiement": "string (required, enum: ['especes', 'carte', 'cheque',
6     'virement', 'autre'])",
7   "datePaiement": "Date (required, default: Date.now, indexed)",
8   "reference": "string (optional)",
9   "notes": "string (optional, maxLength: 500)",
10  "createdAt": "Date (default: Date.now)",
11  "salonId": "ObjectId (optional, pour multi-tenant)"
}

```

**Index recommandés :**

- Index sur `factureId` pour l'historique des paiements d'une facture
- Index sur `datePaiement` pour les rapports financiers

**8.2 Relations entre Collections**

- `users (1) < (N) operations`
- `users (1) < (N) rendezvous`
- `users (1) < (N) conges`
- `users (1) < (N) stockAlerts`
- `clients (1) < (N) rendezvous`
- `clients (1) < (N) factures`
- `services (1) < (N) rendezvous`
- `rendezvous (1) < (N) factures`
- `factures (1) < (N) paiements`
- `produits (1) < (N) stockAlerts`

**8.3 Contraintes d'Intégrité****Référentielle :**

- `operations.coiffeurId` doit référencer un `users._id` existant
- `rendezvous.clientId` doit référencer un `clients._id` existant
- `rendezvous.coiffeurId` doit référencer un `users._id` existant
- `rendezvous.serviceId` doit référencer un `services._id` existant
- `factures.clientId` doit référencer un `clients._id` existant

- `factures.rendezVousId` doit référencer un `rendezvous._id` existant
- `paiements.factureId` doit référencer un `factures._id` existant
- `conges.coiffeurId` doit référencer un `users._id` existant
- `stockAlerts.coiffeurId` doit référencer un `users._id` existant
- `stockAlerts.produitId` doit référencer un `produits._id` existant (si fourni)

#### Domaine :

- `users.email` : format email valide, unique
- `users.password` : minimum 8 caractères, au moins 1 majuscule et 1 chiffre
- `clients.phone` : format valide, unique
- `rendezvous.dateRendezVous` :  $\geq$  Date actuelle (pas de dates passées)
- `rendezvous.heureDebut` :  $<$  `heureFin`
- `services.prix` :  $\geq 0$
- `services.dureeEstimee` :  $\geq 1$  minute
- `factures.total` :  $\geq 0$
- `paiements.montant` :  $\geq 0$  et  $\leq$  montant restant de la facture
- `operations.tarif` :  $\geq 0$
- `operations.pourboire` :  $\geq 0$
- `conges.dateDebut` :  $<$  `dateFin`
- `conges.dateDebut` :  $\geq$  Date actuelle (pas de dates passées)

#### Business :

- Un coiffeur ne peut pas avoir deux rendez-vous qui se chevauchent
- Un coiffeur ne peut pas avoir deux congés qui se chevauchent
- Une opération ne peut être modifiée que par son créateur
- Un coiffeur ne peut pas supprimer une opération de plus de 30 jours
- Un rendez-vous ne peut être créé que si le coiffeur est disponible
- Une facture ne peut être créée que pour un rendez-vous terminé
- Le total des paiements d'une facture ne peut pas dépasser le montant total de la facture
- Un service ne peut être supprimé que s'il n'est pas utilisé dans des rendez-vous futurs

## 9 Matrice de Dépendances

Cette matrice indique les dépendances entre les modules pour planifier le développement :

Module	Dépend de	Description
<b>Module 1</b> : Authentification	-	Module de base, aucune dépendance
<b>Module 2</b> : Gestion Clientèle	Module 1	Nécessite l'authentification pour gérer les clients

Module	Dépend de	Description
<b>Module 3</b> : Gestion des Rendez-vous	Module 1, Module 2, Module 4, Module 5	Nécessite authentification, clients, personnel et services
<b>Module 4</b> : Gestion du Personnel	Module 1	Nécessite l'authentification pour créer des comptes
<b>Module 5</b> : Services et Prestations	Module 1	Nécessite l'authentification (admin uniquement)
<b>Module 6</b> : Gestion de Stock	Module 1	Nécessite l'authentification pour gérer le stock
<b>Module 7</b> : Facturation et Paiements	Module 1, Module 3	Nécessite authentification et rendez-vous
<b>Module 8</b> : Reporting et Analytics	Module 1, Module 3, Module 7	Nécessite authentification, rendez-vous et facturation
<b>Module 9</b> : Administration et Audit	Module 1	Nécessite l'authentification admin
<b>Module 10</b> : Gestion des Congés	Module 1, Module 4	Nécessite authentification et gestion du personnel

## 10 Règles Métier

---

### 10.1 Règles d'Authentification et Sécurité

#### 10.1.1 Politique de Mots de Passe

- Minimum 8 caractères
- Au moins 1 majuscule
- Au moins 1 chiffre
- Au moins 1 caractère spécial (recommandé)
- Expiration : 90 jours (optionnel pour MVP)
- Historique : les 5 derniers mots de passe ne peuvent pas être réutilisés

#### 10.1.2 Gestion des Sessions

- Durée de session par défaut : 24 heures
- Avec "Se souvenir de moi" : 7 jours
- Refresh token : renouvellement automatique si activité dans les 30 dernières minutes
- Invalidation : déconnexion automatique après inactivité de 2 heures

#### 10.1.3 Tentatives de Connexion

- Maximum 5 tentatives échouées
- Blocage temporaire : 15 minutes après 5 échecs
- Notification par email après 3 tentatives échouées

## 10.2 Règles de Gestion des Opérations

### 10.2.1 Validation des Opérations

- Le tarif doit être  $> 0$
- Le pourboire doit être  $\geq 0$
- La date d'opération ne peut pas être dans le futur (max : date/heure actuelle)
- La date d'opération ne peut pas être antérieure à 1 an

### 10.2.2 Modification et Suppression

- Un coiffeur ne peut modifier/supprimer que ses propres opérations
- Modification possible uniquement dans les 30 jours suivant la création
- Suppression possible uniquement dans les 7 jours suivant la création
- Après ces délais, seule l'admin peut modifier/supprimer (avec justification)

### 10.2.3 Calcul des Totaux

- Total = tarif + pourboire
- Calcul automatique côté serveur (non confiance côté client)

## 10.3 Règles de Gestion des Congés

### 10.3.1 Validation des Dates

- Date de début < date de fin
- Pas de dates passées autorisées
- Durée maximale : 30 jours consécutifs
- Délai de préavis minimum : 7 jours pour congés réguliers

### 10.3.2 Gestion des Conflits

- Vérification automatique des chevauchements avec autres congés du même coiffeur
- Alerte si chevauchement détecté
- Un coiffeur ne peut pas avoir plus de 2 congés en attente simultanément

### 10.3.3 Statuts et Workflow

- Statut initial : "en\_attente"
- Seul l'admin peut approuver/refuser
- Notification automatique au coiffeur lors du changement de statut
- Les congés approuvés ne peuvent pas être modifiés (seulement annulés)

## 10.4 Règles de Gestion des Stocks

### 10.4.1 Seuils d'Alerte

- "Épuisé" : quantité = 0

- "Presque épuisé" : quantité  $\leq$  seuil d'alerte du produit (défaut : 5)
- Alerte automatique si quantité atteint le seuil (fonctionnalité future)

#### 10.4.2 Validation des Avertissements

- Un produit peut être créé lors du signalement s'il n'existe pas
- Un coiffeur ne peut pas créer plus de 10 avertissements par jour
- Les avertissements résolus ne peuvent pas être modifiés

### 10.5 Règles de Statistiques et Rapports

#### 10.5.1 Calcul des Statistiques

- Revenus totaux : somme de tous les totaux (tarif + pourboire) sur la période
- Revenu moyen : revenus totaux / nombre d'opérations
- Pourcentage par type de travail : (nombre opérations type X / total opérations)  $\times$  100

#### 10.5.2 Périodes de Calcul

- Jour : du début de la journée (00 :00) à la fin (23 :59)
- Semaine : du lundi au dimanche
- Mois : du 1er au dernier jour du mois
- Année : du 1er janvier au 31 décembre
- Personnalisé : dates sélectionnées par l'utilisateur

#### 10.5.3 Export de Rapports

- PDF : génération avec bibliothèque (ex : jsPDF, PDFKit)
- Excel : export avec formatage (colonnes, totaux, graphiques)
- Limite : maximum 10 000 lignes par export

### 10.6 Règles de Privilèges

#### 10.6.1 Attribution des Privilèges

- Seul l'admin peut attribuer/modifier les privilèges
- Les privilèges sont appliqués immédiatement après sauvegarde
- Un coiffeur avec le privilège "gestionRapports" peut voir les statistiques de tous les coiffeurs

#### 10.6.2 Hiérarchie des Privilèges

- Admin : tous les privilèges par défaut
- Coiffeur : aucun privilège par défaut
- Les privilèges peuvent être combinés

## 11 Architecture API REST

### 11.1 Principe Général

L'application expose une API REST pour la communication entre le frontend et le backend. L'architecture suit les principes RESTful standards avec une séparation claire des responsabilités.

### 11.2 Documentation API

#### Documentation API Séparée

La documentation complète et détaillée de l'API REST sera créée et maintenue dans un document séparé utilisant des outils professionnels dédiés :

- **Swagger/OpenAPI** : Documentation interactive avec spécification OpenAPI 3.0
- **Postman Collection** : Collection Postman pour les tests et la documentation
- **Documentation en ligne** : Accès via une interface dédiée (ex : Swagger UI)

Cette séparation permet :

- Une documentation API toujours à jour et synchronisée avec le code
- Des outils spécialisés pour tester et valider les endpoints
- Une meilleure maintenabilité et évolution de la documentation
- Une utilisation facilitée par les développeurs frontend et les intégrateurs

### 11.3 Caractéristiques Générales de l'API

#### 11.3.1 Architecture

- **Style** : RESTful API
- **Format des données** : JSON
- **Versioning** : /api/v1/ (préparation pour futures versions)
- **Authentification** : JWT (JSON Web Tokens)

#### 11.3.2 Standards et Conventions

- Utilisation des méthodes HTTP standards (GET, POST, PUT, DELETE)
- Codes de statut HTTP appropriés (200, 201, 400, 401, 403, 404, 500, etc.)
- Pagination pour les listes importantes
- Filtrage et recherche via query parameters
- Gestion d'erreurs standardisée avec messages clairs

#### 11.3.3 Sécurité

- Authentification JWT obligatoire pour toutes les routes protégées
- Rate limiting pour prévenir les abus

- Validation des données côté serveur
- Protection CSRF pour les requêtes modifiantes

#### 11.3.4 Modules API

L'API sera organisée en modules correspondant aux fonctionnalités métier :

- **Authentification** : /api/v1/auth/\*
- **Utilisateurs** : /api/v1/users/\*
- **Opérations** : /api/v1/operations/\*
- **Statistiques** : /api/v1/statistics/\*
- **Congés** : /api/v1/conges/\*
- **Stocks** : /api/v1/stock-alerts/\*

#### 11.4 Livrables API

- **Spécification OpenAPI** : Fichier YAML/JSON conforme à OpenAPI 3.0
- **Collection Postman** : Collection complète avec exemples de requêtes
- **Documentation Swagger** : Interface interactive accessible en ligne
- **Guide d'intégration** : Documentation pour les développeurs frontend

## 12 Exigences de Performance

---

### 12.1 Temps de Réponse

#### API REST :

- Temps de réponse moyen : < 200ms (95e percentile)
- Temps de réponse maximum : < 500ms (99e percentile)
- Requêtes critiques (login, création opération) : < 300ms

#### Interface Utilisateur :

- Temps de chargement initial : < 2 secondes
- Temps de chargement des pages suivantes : < 1 seconde
- Temps de rendu des graphiques : < 500ms

### 12.2 Capacité et Scalabilité

- **Utilisateurs simultanés** : Minimum 100 utilisateurs simultanés
- **Throughput** : 1000 requêtes/seconde
- **Opérations par seconde** : 500 opérations/seconde
- **Stockage** : Support jusqu'à 1 million d'opérations par salon

### 12.3 Disponibilité

- **SLA** : 99.9% de disponibilité (maximum 8.76 heures d'indisponibilité par an)
- **Maintenance planifiée** : Maximum 4 heures par mois, préavis de 48h
- **Temps de récupération** : < 15 minutes en cas d'incident

### 12.4 Optimisation

#### 12.4.1 Cache

- Cache des statistiques : 5 minutes
- Cache des listes d'utilisateurs : 2 minutes
- Cache des produits : 10 minutes

#### 12.4.2 Base de données

- Index sur tous les champs fréquemment interrogés
- Pagination obligatoire pour les listes (> 20 éléments)
- Limite de 1000 résultats par requête

#### 12.4.3 Frontend

- Code splitting par route
- Lazy loading des composants lourds
- Compression GZIP pour tous les assets
- CDN pour les assets statiques

### 12.5 Monitoring

#### 12.5.1 Métriques à surveiller

- Temps de réponse API (p50, p95, p99)
- Taux d'erreur (< 0.1%)
- Utilisation CPU/RAM
- Taille de la base de données
- Nombre d'utilisateurs actifs

#### 12.5.2 Alertes

- Temps de réponse > 500ms
- Taux d'erreur > 1%
- Disponibilité < 99%
- Utilisation CPU > 80%

## 13 Politique de Sécurité

---

### 13.1 Authentification et Autorisation

#### 13.1.1 JWT (JSON Web Tokens)

- Algorithme : HS256 ou RS256
- Durée de vie du token : 24 heures
- Durée de vie du refresh token : 7 jours
- Secret stocké dans variable d'environnement (jamais en code)
- Tokens invalidés lors de la déconnexion

#### 13.1.2 Hashage des Mots de Passe

- Algorithme : bcrypt avec salt rounds = 10
- Mots de passe jamais stockés en clair
- Validation côté serveur obligatoire

#### 13.1.3 Protection CSRF

- Tokens CSRF pour les requêtes POST/PUT/DELETE
- Validation côté serveur

#### 13.1.4 Protection XSS

- Sanitization de toutes les entrées utilisateur
- Échappement HTML côté serveur
- Content Security Policy (CSP) headers

### 13.2 Chiffrement des Données

#### 13.2.1 Données en Transit

- HTTPS obligatoire en production (TLS 1.2 minimum)
- Certificat SSL valide
- Redirection HTTP vers HTTPS

#### 13.2.2 Données au Repos

- Chiffrement des données sensibles (mots de passe, tokens)
- Chiffrement optionnel de la base de données MongoDB

### 13.3 Conformité RGPD

#### 13.3.1 Droit à l'Information

- Politique de confidentialité claire

- Consentement explicite pour le traitement des données

#### 13.3.2 Droit d'Accès

- Export des données personnelles (format JSON/CSV)
- Accès aux données via interface utilisateur

#### 13.3.3 Droit à l'Oubli

- Suppression complète des données utilisateur sur demande
- Anonymisation des données dans les statistiques

#### 13.3.4 Portabilité des Données

- Export des données au format standardisé
- API d'export pour faciliter la migration

#### 13.3.5 Sécurité des Données

- Chiffrement des données personnelles
- Accès restreint aux données sensibles
- Audit trail des accès aux données

### 13.4 Gestion des Sessions

#### 13.4.1 Sécurité des Sessions

- Invalidation après déconnexion
- Expiration automatique après inactivité
- Rotation des tokens lors du refresh
- Détection de sessions multiples (optionnel)

#### 13.4.2 Headers de Sécurité

- X-Content-Type-Options: nosniff
- X-Frame-Options: DENY
- X-XSS-Protection: 1; mode=block
- Strict-Transport-Security: max-age=31536000

### 13.5 Audit et Logging

#### 13.5.1 Événements à Logger

- Tentatives de connexion (succès/échec)
- Modifications de données sensibles
- Accès aux fonctionnalités admin
- Erreurs système

### 13.5.2 Informations de Log

- Timestamp
- ID utilisateur
- Action effectuée
- IP address
- User-Agent

### 13.5.3 Rétention des Logs

- Durée de conservation : 90 jours
- Archivage des logs anciens
- Rotation quotidienne des fichiers de log

## 14 Stratégie de Tests

---

### 14.1 Types de Tests

#### 14.1.1 Tests Unitaires (Backend)

- **Couverture cible :** > 80%
- **Outils :** Jest, Supertest
- **Modules à tester :**
  - Modèles Mongoose (validation, méthodes)
  - Controllers (logique métier)
  - Middleware (authentification, validation)
  - Utilitaires (calculs, formattage)

#### 14.1.2 Tests Unitaires (Frontend)

- **Couverture cible :** > 80%
- **Outils :** Jest, React Testing Library
- **Composants à tester :**
  - Composants React (rendu, interactions)
  - Hooks personnalisés
  - Redux reducers et actions
  - Utilitaires frontend

#### 14.1.3 Tests d'Intégration

- **Outils :** Jest, Supertest
- **Scénarios :**

- Flux d'authentification complet
- Création et modification d'opérations
- Génération de statistiques
- Workflow de gestion des congés

#### 14.1.4 Tests End-to-End (E2E)

- **Couverture cible :** Scénarios critiques
- **Outils :** Playwright ou Selenium
- **Scénarios critiques :**
  1. Connexion → Dashboard → Saisie opération → Vérification historique
  2. Admin : Création coiffeur → Attribution privilèges → Vérification accès
  3. Demande congé → Approbation admin → Vérification statut
  4. Génération rapport → Export PDF → Vérification contenu

#### 14.1.5 Tests de Performance

- **Outils :** Artillery, k6, ou JMeter
- **Métriques :**
  - Temps de réponse API sous charge
  - Throughput maximum
  - Utilisation mémoire/CPU
  - Temps de chargement des pages

#### 14.1.6 Tests de Sécurité

- **Outils :** OWASP ZAP, npm audit
- **Vérifications :**
  - Injection SQL/NoSQL
  - XSS (Cross-Site Scripting)
  - CSRF (Cross-Site Request Forgery)
  - Authentification et autorisation
  - Vulnérabilités des dépendances

### 14.2 Scénarios de Tests Critiques

#### 14.2.1 US-001 : Authentification

```

1 // Tests effectuer
2 - Connexion avec identifiants valides
3 - Connexion avec email invalide
4 - Connexion avec mot de passe incorrect
5 - Tentatives multiples (blocage après 5)
6 - Refresh token valide
7 - Refresh token expir

```

- 8 - D connexion et invalidation token

### 14.2.2 US-005 : Saisie d'Opérations

```

1 // Tests effectuer
2 - Cr ation op ration avec donn es valides
3 - Validation tarif > 0
4 - Validation date non future
5 - Calcul automatique du total
6 - Cr ation sans authentification (doit chouer )
7 - Modification par autre coiffeur (doit chouer )

```

### 14.2.3 US-009 : Consultation Opérations Admin

```

1 // Tests effectuer
2 - Liste toutes les op rations (admin)
3 - Filtrage par coiffeur
4 - Filtrage par p riode
5 - Pagination fonctionnelle
6 - Acc s refus pour coiffeur non-admin

```

## 14.3 Données de Test

### 14.3.1 Jeu de données de base

- 10 coiffeurs (dont 1 admin)
- 500 opérations réparties sur 6 mois
- 20 demandes de congés
- 15 avertissements stock

### 14.3.2 Données de charge

- 1000 coiffeurs
- 100 000 opérations
- Tests de performance avec ces volumes

## 14.4 Environnements de Test

### 14.4.1 Développement

- Base de données locale MongoDB
- Données de test minimales
- Logs détaillés activés

### 14.4.2 Staging

- Environnement identique à la production
- Données anonymisées de production
- Tests E2E complets

#### 14.4.3 Production

- Tests de smoke (vérification basique)
- Monitoring continu
- Tests de régression avant déploiement

## 15 Spécifications UI/UX

---

### 15.1 Charte Graphique

#### 15.1.1 Couleurs Principales

- **Primaire :**  #6366f1 (Indigo) - Boutons principaux, liens
- **Secondaire :**  #8b5cf6 (Violet) - Accents, highlights
- **Succès :**  #10b981 (Vert) - Messages de succès
- **Erreur :**  #ef4444 (Rouge) - Messages d'erreur
- **Avertissement :**  #f59e0b (Orange) - Avertissements
- **Info :**  #3b82f6 (Bleu) - Informations

#### 15.1.2 Typographie

- **Police principale :** Inter ou System UI
- **Titres H1 :** 32px, poids 700
- **Titres H2 :** 24px, poids 600
- **Titres H3 :** 20px, poids 600
- **Corps de texte :** 16px, poids 400
- **Texte secondaire :** 14px, poids 400
- **Labels :** 14px, poids 500

### 15.2 Responsive Design

#### 15.2.1 Breakpoints

- **Mobile :** < 640px
- **Tablet :** 640px - 1024px
- **Desktop :** > 1024px

#### 15.2.2 Adaptations Mobile

- Menu latéral → Menu hamburger
- Tableaux → Cartes empilées

- Colonnes multiples → Colonne unique
- Boutons pleine largeur sur petits écrans

### 15.3 Guidelines UX

#### 15.3.1 Navigation

- **Menu latéral** : Toujours visible sur desktop, hamburger menu sur mobile
- **Breadcrumbs** : Affichés sur les pages de niveau 2+
- **Recherche globale** : Disponible dans le header (admin uniquement)

#### 15.3.2 Feedback Utilisateur

- **Messages de succès** : Affichage toast vert, disparition après 3 secondes
- **Messages d'erreur** : Affichage toast rouge, persistant jusqu'à action utilisateur
- **Chargement** : Spinner ou skeleton loader pendant les chargements
- **Confirmations** : Modal de confirmation pour actions destructives

#### 15.3.3 Formulaires

- **Validation en temps réel** : Affichage des erreurs au blur
- **Champs obligatoires** : Marqués avec astérisque (\*)
- **Messages d'erreur** : Sous chaque champ concerné
- **Bouton de soumission** : Désactivé pendant la soumission

#### 15.3.4 Tableaux et Listes

- **Pagination** : 20 éléments par page par défaut
- **Tri** : Clic sur en-tête de colonne
- **Recherche** : Barre de recherche au-dessus du tableau
- **Actions en masse** : Checkbox pour sélection multiple (admin)

### 15.4 Wireframes Principaux

#### 15.4.1 Page de Connexion

Logo Salon

Email  
[-----]

Mot de passe  
[-----]

[ ] Se souvenir

[ Connexion ]

Mot de passe oublié

#### 15.4.2 Dashboard Coiffeur

[Menu] Dashboard [Profil]

Revenu Opérat Moyen  
Total ions  
1250Dh 45 27Dh

[Graphique évolution revenus]

[Dernières opérations]

Date	Type	Tarif	Total
13/11	Coupe	25Dh	30Dh

#### 15.4.3 Formulaire Saisie Opération

[Menu] Nouvelle Opération

Type de travail \*  
[Sélectionner ]

Tarif \*  
[\_\_\_\_\_] Dh

Pourboire  
[\_\_\_\_\_] Dh

Total : 0.00 Dh

Date et heure \*  
[\_\_\_\_\_] [\_\_\_\_\_]

Notes  
[\_\_\_\_\_  
[\_\_\_\_\_]

[Annuler] [Enregistrer]

### 15.5 Accessibilité (WCAG 2.1)

### 15.5.1 Niveau de Conformité

- **Cible** : Niveau AA (recommandé)

### 15.5.2 Exigences

- **Contraste** : Ratio minimum 4.5 :1 pour texte normal, 3 :1 pour texte large
- **Navigation clavier** : Toutes les fonctionnalités accessibles au clavier
- **Focus visible** : Indicateur de focus clair sur tous les éléments interactifs
- **Labels** : Tous les champs de formulaire ont des labels associés
- **Alt text** : Toutes les images ont un texte alternatif
- **ARIA** : Utilisation d'attributs ARIA pour améliorer l'accessibilité

## 16 Technologies et outils

---

### 16.1 Frontend

- **ReactJS** : bibliothèque principale pour l'interface utilisateur
- **Redux** : gestion d'état globale
- **React Router** : gestion de la navigation
- **Axios** : requêtes HTTP
- **Tailwind CSS** : framework CSS
- **Chart.js / Recharts** : bibliothèques pour les graphiques

### 16.2 Backend

- **Node.js** : environnement d'exécution
- **Express.js** : framework web
- **MongoDB** : base de données NoSQL
- **Mongoose** : ODM pour MongoDB
- **JWT** : authentification par tokens
- **bcrypt** : hashage des mots de passe
- **Nodemailer** : envoi d'emails

### 16.3 DevOps

- **Docker** : conteneurisation
- **Docker Compose** : orchestration
- **GitHub Actions** : CI/CD
- **GitHub** : contrôle de version

## 16.4 Qualité

- **ESLint** : linting
- **Prettier** : formatage
- **Husky** : hooks Git

## 16.5 Tests

- **Jest** : tests unitaires
- **Playwright / Selenium** : tests E2E
- **Supertest** : tests d'API

## 16.6 Gestion de projet

- **JIRA** : planification et suivi
- **GitHub** : contrôle de version

## 16.7 Internationalisation

- **i18next** : gestion de l'internationalisation
- **react-i18next** : intégration React

# 17 Ordre de développement recommandé

---

## 1. Phase 1 - FONDATIONS

- Module 1 : Authentification et Sécurité
- Module 2 : Gestion Clientèle (base)
- Module 3 : Gestion des Rendez-vous (base)

## 2. Phase 2 - OPÉRATIONS

- Module 5 : Services et Prestations
- Module 4 : Gestion du Personnel
- Module 7 : Facturation et Paiements (base)

## 3. Phase 3 - OPTIMISATION

- Module 6 : Gestion de Stock
- Module 8 : Reporting et Analytics
- Module 9 : Administration et Audit
- Module 10 : Gestion des Congés

## 4. Phase 4 - AVANCÉ

- Interface publique clients
- Fonctionnalités avancées

## 18 Écrans à développer

---

### 18.1 Authentification

- **Connexion** : formulaire de login pour coiffeur
- **Réinitialisation mot de passe** : récupération de compte

### 18.2 Gestion des Utilisateurs (Admin)

- **Liste des coiffeurs** : tableau avec recherche et filtres
- **Création de compte** : formulaire de création de coiffeur
- **Profil coiffeur** : vue et édition des informations personnelles
- **Gestion des priviléges** : interface d'attribution des permissions

### 18.3 Gestion des Opérations (Coiffeur)

- **Dashboard coiffeur** : vue d'ensemble avec statistiques rapides
- **Saisie d'opération** : formulaire de création d'opération
- **Historique des opérations** : liste avec filtres et recherche
- **Détail d'opération** : vue et édition d'une opération

### 18.4 Statistiques et Rapports

- **Dashboard statistiques coiffeur** : graphiques et indicateurs personnels
- **Vue globale admin** : liste de toutes les opérations avec filtres
- **Génération de rapports** : interface de création de rapports
- **Visualisation de rapports** : affichage des rapports générés

### 18.5 Gestion des Congés (Coiffeur)

- **Demande de congé** : formulaire de création de demande de congé
- **Historique des congés** : liste des demandes de congés avec statuts
- **Détail d'une demande** : vue détaillée d'une demande de congé

### 18.6 Gestion des Stocks (Coiffeur)

- **Signalement produit épuisé** : formulaire de signalement d'un produit/matière première
- **Historique des avertissements** : liste des avertissements envoyés
- **Recherche de produits** : interface de recherche de produits existants

### 18.7 Pages système

- **403** : Accès interdit

- **404** : Page non trouvée
- **500** : Erreur serveur
- **Maintenance** : page de maintenance

## 19 Modalités pédagogiques

---

- **Type de travail** : Individuel
- **Dates** : Du 11/11/2025 au 30/03/2026

## 20 Modalités d'évaluation

---

La présentation se déroulera sur une durée de **45 minutes**, répartie de la manière suivante :

1. **Introduction du travail (10 minutes)**
  - Présentation du projet devant le jury
  - Contexte et objectifs
2. **Démonstration (15 minutes)**
  - Démonstration de l'application et des fonctionnalités
  - Parcours utilisateur complet
3. **Mise en situation pratique (10 minutes)**
  - Scénarios pratiques et démonstrations
  - Réponses aux questions techniques
4. **Code Review et Questions (10 minutes)**
  - Revue de code
  - Questions sur la culture web et les bonnes pratiques

## 21 Livrables attendus

---

### 21.1 Code source

- **Lien GitHub** : dépôt public de l'application
- **Sources complètes** : code correctement commenté et organisé
- **Qualité du code** : modulaire et respectant les bonnes pratiques

### 21.2 Documentation technique

### 21.2.1 Architecture

- **Description de l'architecture** : structure de l'application, choix techniques, diagrammes
- **Diagrammes UML** : diagrammes de classes, séquences, déploiement

### 21.2.2 Installation et configuration

- **Guide d'installation** : étapes détaillées pour mettre en place l'environnement
- **Guide de configuration** : paramétrage de l'application
- **Variables d'environnement** : documentation complète des variables nécessaires

### 21.2.3 API

- **Documentation API** : endpoints avec exemples de requêtes/réponses
- **Schémas de données** : modèles de données et validations

### 21.2.4 Dépendances

- **Bibliothèques externes** : liste complète avec justifications
- **Documentation des dépendances** : versions, utilités, alternatives considérées

## 21.3 Dockerisation

- **Docker Compose** : configuration pour back-end, base de données et front-end
- **Dockerfile** : fichiers Dockerfile optimisés pour chaque service
- **Documentation Docker** : guide d'utilisation et déploiement

## 21.4 CI/CD

- **Pipeline CI/CD** : configuration GitHub Actions ou Jenkins
- **Tests automatisés** : intégration dans le pipeline
- **Déploiement automatique** : configuration du déploiement

## 21.5 Présentation

- **Présentation du projet** : support de présentation (PowerPoint, PDF, etc.)
- **Vidéo de démonstration** : optionnel mais recommandé

# 22 Critères de performance

---

## 22.1 Fonctionnalités

- **Taux de complétion** : pourcentage de fonctionnalités implémentées par rapport aux spécifications (minimum 80%)
- **Qualité fonctionnelle** : toutes les fonctionnalités Must Have doivent être opérationnelles

## 22.2 Authentification et Sécurité

- **Authentification JWT** : sécuriser les données avec JWT, middleware correctement implémenté
- **Gestion des exceptions** : traitement approprié des erreurs
- **Validation des données** : validation côté backend et frontend
- **Sécurité des mots de passe** : hashage avec bcrypt, critères de sécurité respectés

## 22.3 Architecture

- **Architecture n-tiers** : structure claire et modulaire (présentation, logique métier, données)
- **Microservices ou distribués** : architecture backend modulaire (optionnel mais recommandé)
- **Structuration du projet** : architecture claire et logique
- **Conventions de nommage** : respect des standards (camelCase, PascalCase, etc.)

## 22.4 React et Hooks

- **Hooks personnalisés** : maîtrise de la création et de l'utilisation
- **Functional Components** : utilisation exclusive des composants fonctionnels (pas de class components)
- **Gestion d'état** : utilisation de Redux pour la gestion des états globaux
- **Architecture Redux** : actions, reducers, store bien structurés

## 22.5 Tests

- **Tests des services** : tests avec Jest avec un taux de couverture qui dépasse 80%
- **Tests automatisés** : utilisation de Playwright ou Selenium pour les tests E2E
- **Tests unitaires** : Jest pour les composants React
- **Couverture de tests** : taux de couverture acceptable (minimum 80%)

## 22.6 Gestion de projet

- **Planification JIRA** :
  - Utilisation des Epics
  - Création de User Stories / Tasks et Sub-tasks
  - Intégration JIRA avec GitHub
  - Mise en place de règles d'automatisation

## 22.7 Dockerisation

- **Docker Compose** : configuration pour back-end, base de données et front-end
- **Dockerfile** : maîtrise de la création et optimisation
- **Networking** : configuration DNS et communication entre services

- **Volumes** : gestion de la persistance des données
- **Dépendances** : gestion des dépendances entre services

## 22.8 CI/CD

- **GitHub Actions ou Jenkins** :

- Workflow de tests automatisés
- Push automatique sur Docker Hub
- Gestion sécurisée des clés secrètes

## 22.9 Qualité du code

- **ESLint** : configuration et respect des règles
- **Prettier** : formatage automatique du code
- **Husky** : hooks Git (pre-commit) pour validation automatique

## 22.10 Internationalisation

- **I18n** : support de plusieurs langues (arabe, français et anglais)
- **Gestion des traductions** : système de traduction fonctionnel

## 22.11 Documentation

- **Documentation complète** : README, diagrammes, endpoints API
- **Commentaires de code** : code bien commenté et documenté
- **Guide utilisateur** : documentation pour les utilisateurs finaux (optionnel)

# 23 Glossaire Technique

---

## 23.1 Termes Généraux

**MERN Stack** Stack technologique composée de MongoDB, Express.js, React.js et Node.js

**Frontend** Partie visible et interactive de l'application côté client

**Backend** Partie serveur de l'application qui gère la logique métier et les données

**API** Interface de programmation applicative permettant la communication entre frontend et backend

**REST API** Architecture d'API basée sur les principes REST

## 23.2 Authentification et Sécurité

**JWT (JSON Web Token)** Token d'authentification sécurisé utilisé pour l'authentification

**bcrypt** Algorithme de hashage pour sécuriser les mots de passe

**Middleware** Fonction intermédiaire qui traite les requêtes avant qu'elles n'atteignent les routes

**Session** État de connexion d'un utilisateur maintenu côté serveur ou client

### 23.3 Technologies Frontend

**ReactJS** Bibliothèque JavaScript pour construire des interfaces utilisateur

**Redux** Bibliothèque de gestion d'état global pour applications JavaScript

**Hooks** Fonctions React permettant d'utiliser l'état et d'autres fonctionnalités sans classes

**Functional Components** Composants React écrits comme des fonctions JavaScript

**React Router** Bibliothèque de routage pour React

### 23.4 Technologies Backend

**Express.js** Framework web minimaliste pour Node.js

**MongoDB** Base de données NoSQL orientée documents

**Mongoose** ODM (Object Document Mapper) pour MongoDB

**REST** Style architectural pour les services web

### 23.5 Gestion de Données

**Opération** Prestation effectuée par un coiffeur (coupe, coloration, etc.)

**Pourboire** Gratification optionnelle ajoutée au tarif de base

**Privilège** Permission accordée à un coiffeur pour accéder à certaines fonctionnalités

**Congé régulier** Congé prévu et planifié à l'avance (vacances, congés payés)

**Congé exceptionnel** Congé imprévu ou pour raison particulière (maladie, urgence familiale)

**Matière première** Produit utilisé dans le salon (shampooing, coloration, produits de soin, etc.)

**Avertissement stock** Signalement d'un produit ou matière première épuisé(e) ou presque épuisé(e)

### 23.6 DevOps et Outils

**Docker** Plateforme de conteneurisation d'applications

**Docker Compose** Outil pour définir et exécuter des applications Docker multi-conteneurs

**CI/CD** Intégration Continue / Déploiement Continu

**GitHub Actions** Plateforme d'automatisation des workflows GitHub

**ESLint** Outil d'analyse statique du code JavaScript

**Prettier** Formateur de code automatique

**Husky** Outil pour gérer les hooks Git

### 23.7 Tests

**Tests E2E (End-to-End)** Tests qui vérifient le fonctionnement complet de l'application

**Tests unitaires** Tests qui vérifient une fonction ou un composant isolément

**Jest** Framework de tests JavaScript

**Playwright / Selenium** Outils d'automatisation de tests navigateur

**Supertest** Bibliothèque pour tester les APIs HTTP

### 23.8 Gestion de Projet

**JIRA** Outil de gestion de projet et suivi des tâches

**Epic** Grande fonctionnalité regroupant plusieurs user stories

**User Story** Description d'une fonctionnalité du point de vue de l'utilisateur

**Task / Sub-task** Tâches et sous-tâches de développement

### 23.9 Architecture

**Multi-tenant** Architecture permettant à plusieurs clients (salons) d'utiliser la même application de manière isolée

**N-tiers** Architecture en plusieurs couches (présentation, logique métier, données)

**Microservices** Architecture où l'application est divisée en petits services indépendants

### 23.10 Internationalisation

**I18n** Internationalisation (processus de préparation d'une application pour plusieurs langues)

**i18next** Framework d'internationalisation pour JavaScript

---

## 24 Notes importantes

---

### 24.1 Phase Multi-Tenant (Future)

- Architecture préparée pour l'isolation des données par salon
- Chaque salon aura son propre espace isolé
- Système d'abonnement à prévoir pour la monétisation
- Migration progressive des données existantes vers le modèle multi-tenant
- Ajout d'un champ **salon\_id** dans toutes les tables pour l'isolation

### 24.2 Bonnes Pratiques

- L'application doit être **performante**, **sécurisée** et **scalable**
- Respecter les **bonnes pratiques** de développement React
- Assurer la **sécurité** des données utilisateur
- Code **modulaire** et **maintenable**
- Documentation **complète** et **claire**
- Tests **complets** avec bonne couverture

### 24.3 Contraintes Techniques

- Utilisation exclusive de **functional components** (pas de class components)
- Gestion d'état avec **Redux** pour les états globaux
- Architecture **n-tiers** structurée

- Tests avec **Jest** (couverture > 80%)
- Dockerisation complète avec **Docker Compose**
- Pipeline **CI/CD** avec GitHub Actions ou Jenkins
- Support **I18n** (arabe, français, anglais)