

# Améliorations de l'UI - Affichage des créneaux réservés

## Date de mise en œuvre

31 décembre 2025

## Contexte

Le projet Call a Star nécessitait une amélioration de l'affichage des créneaux déjà réservés pour éviter toute confusion utilisateur. Bien que le backend ait déjà été sécurisé pour empêcher les doubles réservations, l'interface utilisateur ne montrait pas clairement quand un créneau était déjà réservé.

## Objectifs

- Afficher clairement les créneaux déjà réservés avec un badge visible
- Désactiver complètement l'action de réservation/paiement pour ces créneaux
- Améliorer l'UX avec des messages explicatifs clairs
- Différencier visuellement un créneau disponible d'un créneau réservé

## Changements implémentés

### 1. Modifications de la page de booking

Fichier : app/[locale]/book/[offerId]/page.tsx

#### a. Imports ajoutés

```
import { Badge } from '@/components/ui/badge';
import { AlertCircle } from 'lucide-react';
```

#### b. Logique de détection des créneaux réservés

La page vérifie maintenant deux conditions pour déterminer si un créneau est réservé :

- **Statut du CallOffer** : Vérification si `status === 'BOOKED'`
- **Booking existant** : Vérification de l'existence d'un booking associé

```
// Check if offer is already booked (by status or by existing booking)
const isOfferBooked = offerData?.callOffer?.status === 'BOOKED' ||
                     offerData?.callOffer?.booking;

if (isOfferBooked) {
  setExistingBooking(offerData.callOffer.booking);
  setLoading(false);
  return;
}
```

#### c. Badge proéminent “Already booked”

Un badge visuel est affiché en haut du formulaire pour indiquer clairement l'état de la réservation :

```
<div className="flex justify-center mb-4">
  <Badge variant="destructive" className="text-lg px-6 py-2 bg-red-600 hover:bg-red-700">
    {isUserBooking ? t('yourBookingBadge') : t('alreadyBooked')}
  </Badge>
</div>
```

#### Caractéristiques du badge :

- Taille large (text-lg, padding étendu)
- Couleur rouge vif (bg-red-600) pour attirer l'attention
- Centré horizontalement
- Texte différencié selon que c'est la réservation de l'utilisateur ou d'un autre utilisateur

#### d. Message explicatif amélioré

Pour les créneaux réservés par d'autres utilisateurs, un message clair est affiché :

```
<div className="bg-red-50 border-2 border-red-300 rounded-lg p-4 flex items-start gap-3">
  <AlertCircle className="w-5 h-5 text-red-600 flex-shrink-0 mt-0.5" />
  <div>
    <p className="text-sm font-semibold text-red-900 mb-1">
      {t('offerNotAvailable')}
    </p>
    <p className="text-xs text-red-700">
      {t('offerBookedExplanation')}
    </p>
  </div>
</div>
```

#### Caractéristiques du message :

- Fond rouge clair (bg-red-50) avec bordure épaisse
- Icône AlertCircle pour attirer l'attention
- Texte en gras pour le titre
- Message explicatif secondaire plus détaillé

## 2. Modifications du composant CheckoutForm

#### a. Propriété `disabled` ajoutée

Le composant accepte maintenant une prop `disabled` pour désactiver complètement le formulaire :

```
function CheckoutForm({
  bookingId,
  onSuccess,
  disabled = false
}: {
  bookingId: string;
  onSuccess: () => void;
  disabled?: boolean
})
```

#### b. Désactivation visuelle et fonctionnelle

Quand `disabled` est `true` :

- Opacité réduite à 50% ( `opacity-50` )
- Aucune interaction possible ( `pointer-events-none` )

- Curseur “not-allowed” ( cursor-not-allowed )
- Bouton de paiement désactivé

```
<form onSubmit={handleSubmit} className={`space-y-6 ${disabled ? 'opacity-50 pointer-events-none cursor-not-allowed' : ''}`}>
  <PaymentElement />
  <Button
    type="submit"
    disabled={!stripe || loading || disabled}>
    {/* ... */}
  </Button>
</form>
```

### 3. Traductions ajoutées (fr.json)

Fichier : messages/fr.json

Nouvelles clés ajoutées dans la section booking :

```
{
  "alreadyBooked": "❌ Déjà réservé",
  "yourBookingBadge": "✅ Votre réservation",
  "offerBookedExplanation": "Ce créneau horaire a déjà été réservé par un autre utilisateur. Veuillez choisir un autre moment."
}
```

**Note** : Seul le fichier fr.json a été modifié conformément aux instructions. Le fichier en.json sera mis à jour lors d'une phase de traduction ultérieure.

## Comment le système détecte les créneaux réservés

### Flux de détection

#### 1. Récupération de l'offre

- Son statut ( AVAILABLE , BOOKED , COMPLETED , CANCELLED )
- Le booking associé (s'il existe)
- Les détails du créateur

#### 2. Vérification double :

typescript

```
const isOfferBooked = offerData?.callOffer?.status === 'BOOKED' ||
                     offerData?.callOffer?.booking;
```

- Première condition : Vérifie le statut du CallOffer

- Deuxième condition : Vérifie l'existence d'un booking

#### 3. Identification du propriétaire :

typescript

```
const isUserBooking = user && existingBooking.userId === user.id;
```

- Compare l'ID de l'utilisateur connecté avec l'ID du booking

## Résultats possibles

Cas	Badge affiché	Interface
Créneau disponible	Aucun	Formulaire de paiement actif
Réservation de l'utilisateur	" Votre réservation"	Détails de la réservation + actions (voir, rejoindre)
Réservation d'un autre utilisateur	" Déjà réservé"	Message explicatif + liens vers profil créateur/autres créateurs

## Composants modifiés

### 1. Page de booking principale

- **Fichier :** app/[locale]/book/[offerId]/page.tsx
- **Lignes modifiées :** ~15 lignes ajoutées, ~5 lignes modifiées
- **Fonction :** BookOfferPage (composant principal)

### 2. Composant CheckoutForm

- **Fichier :** app/[locale]/book/[offerId]/page.tsx (même fichier)
- **Lignes modifiées :** ~3 lignes ajoutées, ~2 lignes modifiées
- **Fonction :** CheckoutForm (composant interne)

### 3. Fichier de traductions

- **Fichier :** messages/fr.json
- **Modifications :** 3 nouvelles clés ajoutées dans la section booking

## Composants UI utilisés

### Badge ( @/components/ui/badge )

- **Variante utilisée :** destructive (fond rouge pour les erreurs/alertes)
- **Personnalisation :** Classes Tailwind pour augmenter la taille et améliorer la visibilité

### AlertCircle ( lucide-react )

- **Usage :** Icône d'alerte dans le message explicatif
- **Taille :** 5x5 (w-5 h-5)
- **Couleur :** Rouge (text-red-600)

## Style et cohérence visuelle

### Palette de couleurs pour les créneaux réservés

- **Badge "Déjà réservé" :**
- Fond : bg-red-600
- Hover : hover:bg-red-700

- **Message d'alerte :**

- Fond : `bg-red-50`
- Bordure : `border-red-300` (épaisseur 2px)
- Texte : `text-red-900` (titre), `text-red-700` (description)
- Icône : `text-red-600`

## Palette de couleurs pour les réservations de l'utilisateur

- **Badge “Votre réservation”** : Même style mais avec le badge destructive
- **Zone de statut** : Fond vert (`bg-green-50`) avec bordure verte

Cette cohérence permet une distinction claire entre :

- Réservations de l'utilisateur (vert)
- Créneaux indisponibles (rouge)
- Actions disponibles (violet/rose - gradient de l'application)

## Tests recommandés

---

### Scénarios à vérifier

#### 1. Créneau disponible

- Aucun badge affiché
- Formulaire de paiement actif
- Bouton “Payer et réserver” cliquable

#### 2. Créneau réservé par l'utilisateur connecté

- Badge “ Votre réservation” affiché
- Détails de la réservation visibles
- Boutons d'action appropriés (Voir mes réservations, Rejoindre l'appel)
- Statut de la réservation affiché (Confirmé, En attente)

#### 3. Créneau réservé par un autre utilisateur

- Badge “ Déjà réservé” affiché
- Message explicatif visible avec icône d'alerte
- Aucun formulaire de paiement affiché
- Liens vers le profil du créateur et la liste des créateurs

#### 4. Styles et accessibilité

- Les couleurs sont suffisamment contrastées
- Les textes sont lisibles
- Les badges sont visibles même sur mobile
- Les icônes ont une taille appropriée

## Sécurité et double protection

---

### Protection en couches

#### 1. Couche UI (ce changement)

- Empêche les utilisateurs de cliquer sur “Payer et réserver” pour un créneau déjà pris
- Fournit un feedback visuel immédiat
- Réduit les appels API inutiles

## 2. Couche Backend (subtask 3 - déjà implémentée)

- Vérification côté serveur avant la création d'un booking
- Empêche les doubles réservations même si quelqu'un contourne l'UI
- Retourne une erreur appropriée si le créneau est déjà réservé

Cette approche en double couche garantit une sécurité maximale contre les doubles réservations.

## Améliorations futures possibles

1. **Rafraîchissement automatique** : Implémenter un polling ou WebSocket pour détecter automatiquement quand un créneau devient réservé pendant que l'utilisateur est sur la page
2. **Animation du badge** : Ajouter une animation subtile au badge pour attirer davantage l'attention
3. **Traductions anglaises** : Mettre à jour `messages/en.json` avec les traductions appropriées
4. **Notification toast** : Afficher une notification toast si l'utilisateur tente de cliquer sur un élément désactivé
5. **Analytics** : Tracker combien d'utilisateurs arrivent sur des créneaux déjà réservés pour optimiser l'UX

## Impact utilisateur

### Avant ces changements

- ✗ Confusion possible : les utilisateurs pouvaient voir un formulaire de paiement pour un créneau déjà réservé
- ✗ Mauvaise expérience : découverte de l'indisponibilité uniquement après tentative de paiement
- ✗ Frustration potentielle

### Après ces changements

- ✅ Clarté immédiate : badge visible dès le chargement de la page
- ✅ Feedback instantané : pas de tentative de paiement possible
- ✅ Guidance claire : messages explicatifs et actions alternatives proposées
- ✅ Expérience utilisateur améliorée

## Conclusion

Ces améliorations UI complètent la sécurisation backend déjà en place et offrent une expérience utilisateur claire et sans ambiguïté. Les utilisateurs savent immédiatement si un créneau est disponible ou non, ce qui réduit la frustration et améliore la confiance dans la plateforme.