

# EmotionsCare - Backlog Technique

## Détaillé

---

### Sprint 1: MVP Consolidation (2 semaines)

Vélocité estimée: 40 points Équipe: 2 dev full-stack

---

#### EPIC 1: Onboarding & Auth

##### S1-01: Refonte flow onboarding

**Story:** En tant qu'utilisateur, je peux compléter un onboarding fluide avec tous les consentements requis

Points: 5 | Priorité: P0

Tâches techniques:

- [ ] 1. Créer composant OnboardingWizard avec 6 steps
  - Fichier: src/components/onboarding/OnboardingWizard.tsx
  - État: step actuel, données collectées
  - Navigation: next/back/skip (si autorisé)
  
- [ ] 2. Step 1: SplashScreen
  - Animation logo Lottie (2s)
  - CTA "Commencer" / "J'ai un compte"
  
- [ ] 3. Step 2: SignupForm
  - Email + password (zod validation)
  - Boutons OAuth (Apple, Google)
  - Checkbox CGU obligatoire
  - Call: supabase.auth.signUp()
  
- [ ] 4. Step 3-4: ConsentScreens
  - Consent IA processing (obligatoire)
  - Consent voice recording (optionnel)
  - Persist: INSERT consent\_records

- [ ] 5. Step 5: HealthDisclaimer
  - Texte légal complet
  - Checkbox "J'ai compris" (required)
  - Numéro 3114 affiché
  
- [ ] 6. Step 6: PersonalizationQuiz
  - Question émotion actuelle (roue)
  - Objectifs (multi-select)
  - Préférence durée sessions
  - UPDATE profiles.preferences
  
- [ ] 7. Tests E2E onboarding complet
  - Happy path
  - Abandon mi-parcours
  - Validation erreurs

**Critères d'acceptation:** - [ ] 6 étapes complétées en < 5 min - [ ] Consentements IA obligatoires bloquants - [ ] Disclaimer santé non-skippable - [ ] Données persistées en BDD - [ ] Redirection dashboard à la fin

**Fichiers impactés:** - `src/components/onboarding/*` - `src/pages/Onboarding.tsx` - `src/stores/onboardingStore.ts` - `supabase/migrations/xxx_consent_records.sql`

---

## S1-02: Système de consentements RGPD

**Story:** En tant que système, je dois tracker tous les consentements utilisateur

**Points:** 3 | **Priorité:** P0

### Tâches techniques:

- [ ] 1. Migration SQL `consent_records`

```
CREATE TABLE consent_records (
    id UUID PRIMARY KEY,
    user_id UUID REFERENCES users(id),
    consent_type TEXT NOT NULL,
    granted BOOLEAN NOT NULL,
    version TEXT NOT NULL,
    ip_address INET,
    granted_at TIMESTAMPTZ DEFAULT now(),
    revoked_at TIMESTAMPTZ
);
```

- [ ] 2. Service ConsentService
  - recordConsent(userId, type, granted, version)
  - revokeConsent(userId, type)
  - getUserConsents(userId)
  - hasConsent(userId, type)
  
- [ ] 3. Hook useConsent
  - Expose: consents, grantConsent, revokeConsent
  - Cache React Query
  
- [ ] 4. RLS policies
  - Users can only read/write own consents

**Critères d'acceptation:** - [ ] 7 types de consentement supportés - [ ] Versioning des documents légaux - [ ] Historique complet conservé - [ ] IP address logged

---

## EPIC 2: Session Émotionnelle

### S1-03: Input émotion texte/choix

**Story:** En tant qu'utilisateur, je peux exprimer mon émotion via texte libre ou sélection guidée

**Points:** 5 | **Priorité:** P0

**Tâches techniques:**

- [ ] 1. Composant EmotionInputSelector
  - Tabs: "Écrire" / "Choisir"
  - Transition fluide entre modes
  
- [ ] 2. Mode Texte: EmotionTextInput
  - Textarea max 500 chars
  - Placeholder: "Comment vous sentez-vous?"
  - Compteur caractères
  - Suggestions autocomplete émotions
  
- [ ] 3. Mode Choix: EmotionWheel
  - Roue Plutchik 8 émotions primaires
  - Touch/click sur segment
  - Animation highlight sélection
  - Sous-émotions au 2ème tap
  
- [ ] 4. IntensitySlider

- Range 1-10
- Labels: "Légère" → "Intense"
- Haptic feedback (mobile)

[ ] 5. ContextTags (optionnel)

- Chips sélectionnables
- Tags: travail, famille, santé, relations, argent, autre
- Multi-select

[ ] 6. Bouton "Analyser mon émotion"

- Disabled si input vide
- Loading state pendant analyse

**Critères d'acceptation:** - [ ] Switch fluide texte ↔ choix - [ ] Roue émotions tactile responsive - [ ] Intensité obligatoire - [ ] Contexte optionnel - [ ] Validation avant soumission

**Fichiers impactés:** - `src/components/emotion/EmotionInputSelector.tsx` - `src/components/emotion/EmotionWheel.tsx` - `src/components/emotion/IntensitySlider.tsx` - `src/components/emotion/ContextTags.tsx`

---

## S1-04: Analyse IA émotion

**Story:** En tant qu'utilisateur, je reçois une analyse IA de mon état émotionnel

**Points:** 5 | **Priorité:** P0

**Tâches techniques:**

[ ] 1. Edge Function: `analyze-emotion`

Fichier: `supabase/functions/analyze-emotion/index.ts`

Input:

```
{
  input_type: "text" | "choice",
  raw_input: string,
  selected_emotion?: string,
  intensity?: number,
  context_tags?: string[]
}
```

Process:

- Call OpenAI GPT-4 avec prompt structuré
- Extraction: émotions détectées, valence, arousal

- Fallback si API down

Output:

```
{  
    detected_emotions: [  
        { label: "anxiété", intensity: 0.8, confidence: 0.92 }  
    ],  
    primary_emotion: "anxiété",  
    valence: -0.6,  
    arousal: 0.7,  
    summary: "Je détecte principalement de l'anxiété..."  
}
```

#### [ ] 2. Prompt engineering

- System prompt: rôle analyste émotionnel
- Contraintes: pas de diagnostic médical
- Format output JSON strict
- Fallback si input ambigu

#### [ ] 3. Composant AnalysisResult

- Affichage émotions détectées
- Barres de progression %
- Texte résumé IA
- Boutons: "Corriger" / "Confirmer"

#### [ ] 4. Feedback loop correction

- Si "Corriger": modal sélection manuelle
- Update session avec correction
- Log pour amélioration modèle

#### [ ] 5. Persistance session

```
INSERT emotion_sessions (  
    user_id, input_type, raw_input,  
    detected_emotions, primary_emotion,  
    intensity, valence, arousal,  
    context_tags, ai_model_version  
)
```

**Critères d'acceptation:** - [ ] Réponse IA < 3 secondes - [ ] Affichage clair des émotions détectées - [ ] Option correction si erreur - [ ] Langage non-médical ("je détecte" vs "vous avez") - [ ] Données persistées avec version modèle

---

## S1-05: Génération plan personnalisé

**Story:** En tant qu'utilisateur, je reçois un plan d'actions adapté à mon état

**Points:** 5 | **Priorité:** P0

**Tâches techniques:**

[ ] 1. Edge Function: generate-plan  
Fichier: supabase/functions/generate-plan/index.ts

Input:

```
{  
    session_id: UUID,  
    user_preferences: { duration_pref, goals },  
    emotion_analysis: { ... },  
    history_summary?: { recent_exercises, effective_ones }  
}
```

Process:

- Règles métier: mapping émotion → exercices
- Personnalisation selon préférences
- Priorisation intelligente
- Call OpenAI pour personnalisation texte

Output:

```
{  
    recommendations: [  
        {  
            type: "breathing",  
            exercise_id: "box-breathing",  
            title: "Respiration carrée",  
            description: "Calmez votre système nerveux...",  
            duration_min: 5,  
            priority: 1  
        },  
        {  
            type: "music",  
            style: "ambient",  
            target_energy: "calming",  
            priority: 2  
        },  
        ...  
    ],  
    estimated_duration_min: 15
```

```
}
```

- [ ] 2. Règles mapping émotion → exercices
  - anxiété → [respiration 4-7-8, cohérence cardiaque, musique calme]
  - colère → [respiration carrée, scan corporel, musique neutre]
  - tristesse → [respiration douce, lumière chaude, musique douce]
  - fatigue → [respiration énergisante, lumière bleue, musique rythmée]

- [ ] 3. Composant PlanDisplay

- Liste verticale recommandations
- Icônes par type (肺💡🎵📝)
- Boutons action: "Commencer" / "Plus tard"
- Durée totale estimée

- [ ] 4. Persistance plan

```
INSERT emotion_plans (  
    session_id, user_id, plan_type,  
    recommendations, status, expires_at  
)
```

- [ ] 5. Navigation vers modules

- Click "Commencer" → route vers module
- Passage session\_id + exercise\_id

**Critères d'acceptation:** - [ ] 3-5 recommandations générées - [ ] Priorisation cohérente avec émotion - [ ] Actions directes vers modules - [ ] Plan persisté en BDD - [ ] Durée estimée affichée

## EPIC 3: Module Respiration

### S1-06: Exercices respiration guidés

**Story:** En tant qu'utilisateur, je peux suivre un exercice de respiration avec animation

**Points:** 5 | **Priorité:** P0

**Tâches techniques:**

- [ ] 1. Composant BreathingExercise
  - Props: exerciseId, onComplete
  - États: idle, inhale, hold, exhale, pause
  - Timer global + timer phase

- [ ] 2. Animation BreathingCircle

- Cercle SVG animé
- Scale: 1 → 1.5 (inhale), 1.5 → 1 (exhale)
- Couleur adaptative selon phase
- Framer Motion pour fluidité

#### [ ] 3. Patterns respiration

```
const PATTERNS = {
  'box-breathing': {
    name: 'Respiration carrée',
    phases: [
      { type: 'inhale', duration: 4 },
      { type: 'hold', duration: 4 },
      { type: 'exhale', duration: 4 },
      { type: 'pause', duration: 4 }
    ],
    cycles: 4
  },
  '4-7-8': {
    phases: [
      { type: 'inhale', duration: 4 },
      { type: 'hold', duration: 7 },
      { type: 'exhale', duration: 8 }
    ],
    cycles: 3
  },
  'coherence': {
    phases: [
      { type: 'inhale', duration: 5 },
      { type: 'exhale', duration: 5 }
    ],
    cycles: 6
  }
}
```

#### [ ] 4. UI exercice

- Instruction texte: "Inspirez", "Retenez", "Expirez"
- Compteur secondes phase
- Progression cycles (2/4)
- Bouton pause/stop

#### [ ] 5. Haptic feedback (mobile)

- Vibration début chaque phase
- Pattern différent par phase

#### [ ] 6. Écran fin exercice

- "Bien joué! Comment vous sentez-vous?"
- Feedback mood: 😊 😃 😐 😞
- Persistance: INSERT exercise\_completions

#### [ ] 7. Calcul métriques

- Durée réelle
- Cycles complétés
- Mood before/after delta

**Critères d'acceptation:** - [ ] 3 exercices disponibles (box, 4-7-8, cohérence) - [ ] Animation cercle fluide 60fps - [ ] Instructions claires par phase - [ ] Timer précis - [ ] Feedback fin obligatoire - [ ] Données persistées

**Fichiers impactés:** - `src/components/breathing/BreathingExercise.tsx` - `src/components/breathing/BreathingCircle.tsx` - `src/components/breathing/BreathingPatterns.ts` - `src/components/breathing/ExerciseComplete.tsx`

---

## EPIC 4: Sécurité & Compliance

### S1-07: Disclaimers légaux permanents

**Story:** En tant qu'utilisateur, je suis informé que l'app ne remplace pas un suivi médical

**Points:** 2 | **Priorité:** PO

#### Tâches techniques:

- [ ] 1. Composant HealthDisclaimer (footer)
  - Texte: "EmotionsCare est un outil de bien-être..."
  - Style discret mais lisible
  - Lien vers page info complète
- [ ] 2. Composant SafetyBanner (pré-session)
  - Apparaît avant chaque nouvelle session
  - "Ceci est un exercice de bien-être"
  - Dismissible (session storage)
- [ ] 3. Page /safety-info
  - Contenu légal complet
  - Ressources professionnelles
  - Numéros urgence (3114, etc.)
- [ ] 4. Intégration layouts

- Footer sur toutes pages authentifiées
- Banner sur page nouvelle session

**Critères d'acceptation:** - [ ] Footer visible sur toutes pages - [ ] Numéro 3114 accessible en 2 taps - [ ] Textes validés juridiquement

---

### S1-08: Détection et escalade crise

**Story:** En tant que système, je détecte les situations à risque et propose des ressources

**Points:** 5 | **Priorité:** P0

**Tâches techniques:**

- [ ] 1. Service CrisisDetectionService
 

```
const CRISIS_KEYWORDS = [
  'suicide', 'suicider', 'mourir', 'mort',
  'me tuer', 'en finir', 'plus envie de vivre',
  'me faire du mal', 'automutilation'
];

function detectCrisis(text: string): {
  detected: boolean;
  severity: 'low' | 'medium' | 'high';
  matchedKeywords: string[];
}
```
- [ ] 2. Analyse patterns répétitifs
  - 3+ sessions consécutives avec valence < -0.7
  - Intensité négative croissante
  - Trigger: afficher ressources
- [ ] 3. Composant CrisisModal
  - Non-dismissible pendant 5s
  - Titre: "Nous sommes là pour vous"
  - Numéro 3114 (bouton appel direct mobile)
  - Fil Santé Jeunes
  - Bouton "Continuer dans l'app" (après 5s)
- [ ] 4. Audit logging
 

```
INSERT audit_logs (
  user_id,
  action: 'crisis_escalation',
```

```
    details: { keywords, severity, user_action }  
)
```

- [ ] 5. Edge case: faux positifs
  - Contexte: "je ne veux pas mourir de fatigue"
  - Analyse contextuelle basique
  - En cas de doute: afficher quand même

**Critères d'acceptation:** - [ ] Détection mots-clés < 100ms - [ ] Modal non-contournable 5s - [ ] Numéro urgence clickable - [ ] Log audit systématique - [ ] Taux faux positifs < 5%

---

## S1-09: Suppression compte RGPD

**Story:** En tant qu'utilisateur, je peux supprimer définitivement mon compte et toutes mes données

**Points:** 3 | **Priorité:** P0

**Tâches techniques:**

- [ ] 1. UI Settings > Supprimer mon compte
  - Bouton rouge "Supprimer mon compte"
  - Warning: "Cette action est irréversible"
- [ ] 2. Modal confirmation double
  - Étape 1: "Êtes-vous sûr?"
  - Étape 2: Saisir "SUPPRIMER" pour confirmer
  - Explication: données supprimées
- [ ] 3. Edge Function: delete-user-data
  - Soft delete immédiat (is\_active = false)
  - Hard delete après 72h (cron job)
  - Ordre suppression:
    1. audio\_files (+ fichiers Storage)
    2. exercise\_completions
    3. light\_sessions
    4. music\_generations
    5. emotion\_plans
    6. emotion\_sessions
    7. consent\_records (garder 5 ans légal)
    8. profiles
    9. users

```

[ ] 4. Email confirmation
    - "Votre compte a été supprimé"
    - Délai 72h mentionné
    - Lien annulation (si < 72h)

[ ] 5. Audit log
    INSERT audit_logs (
        user_id, action: 'account_deletion_requested'
    )

```

**Critères d'acceptation:** - [ ] Double confirmation obligatoire - [ ] Suppression effective < 72h - [ ] Email confirmation envoyé - [ ] Fichiers Storage supprimés - [ ] Consentements archivés (légal)

---

## S1-10: Tests E2E critiques

**Story:** En tant que développeur, je m'assure que les flows critiques fonctionnent

**Points:** 5 | **Priorité:** P0

### Tâches techniques:

```

[ ] 1. Test onboarding complet
    test('user can complete onboarding', async () => {
        // Signup
        // Accept consents
        // Accept disclaimer
        // Complete personalization
        // Verify redirect to dashboard
    })

[ ] 2. Test session émotionnelle
    test('user can complete emotion session', async () => {
        // Input emotion (text)
        // View analysis result
        // View generated plan
        // Start breathing exercise
        // Complete exercise
        // Submit feedback
    })

[ ] 3. Test escalade crise
    test('crisis keywords trigger safety modal', async () => {
        // Input text with crisis keyword
    })

```

```

    // Verify modal appears
    // Verify cannot dismiss for 5s
    // Verify 3114 link works
  })

[ ] 4. Test suppression compte
test('user can delete account', async () => {
  // Navigate to settings
  // Click delete
  // Double confirm
  // Verify logout
  // Verify cannot login again
})

[ ] 5. Tests accessibilité
- Keyboard navigation
- Screen reader labels
- Color contrast

```

**Critères d'acceptation:** - [ ] 4 flows critiques couverts - [ ] Tests passent en CI - [ ] Temps exécution < 2 min - [ ] Screenshots on failure

---

## Sprint 2: V1 Features (2 semaines)

**Vélocité estimée:** 40 points **Équipe:** 2 dev full-stack

---

### EPIC 5: Input Vocal

#### S2-01: Enregistrement et analyse vocale

**Story:** En tant qu'utilisateur, je peux exprimer mes émotions par la voix

**Points:** 8 | **Priorité:** P0

**Tâches techniques:**

- ```

[ ] 1. Composant VoiceRecorder
- Bouton push-to-talk
- Visualisation waveform temps réel
- Timer (max 60s)
- États: idle, recording, processing

```

[ ] 2. Consentement pré-enregistrement  
- Modal si premier usage  
- "Autorisez-vous l'enregistrement vocal?"  
- Persist consent\_records

[ ] 3. API MediaRecorder  
const mediaRecorder = new MediaRecorder(stream, {  
 mimeType: 'audio/webm;codecs=opus'  
});  
- Gestion permissions microphone  
- Fallback si refusé

[ ] 4. Upload Supabase Storage  
- Bucket: 'voice-recordings'  
- Path: {userId}/{timestamp}.webm  
- Chiffrement at rest

[ ] 5. Transcription (Whisper via OpenAI)  
- Edge Function: transcribe-audio  
- Input: audio file URL  
- Output: texte transcript

[ ] 6. Analyse prosodique Hume AI  
- Edge Function: analyze-voice-emotion  
- Input: audio file  
- Output: {  
 emotions: [...],  
 prosody: { pitch, tempo, energy }  
}

[ ] 7. Affichage résultat unifié  
- Transcription texte  
- Émotions détectées (voix + texte)  
- Fusion des analyses

[ ] 8. Cleanup automatique  
- Supprimer audio 24h après transcription  
- pg\_cron job quotidien

**Critères d'acceptation:** - [ ] Enregistrement max 60s - [ ] Consentement explicite requis - [] Transcription < 10s - [ ] Analyse Hume AI intégrée - [ ] Audio supprimé après 24h

## EPIC 6: Musicothérapie IA

### S2-02: Génération musique Suno

**Story:** En tant qu'utilisateur, je peux générer une musique personnalisée selon mon émotion

**Points:** 8 | **Priorité:** P0

**Tâches techniques:**

```
[ ] 1. Edge Function: generate-music
  Input:
  {
    emotion: "anxiété",
    target_energy: "calming",
    duration_seconds: 60,
    style_preferences: ["ambient", "piano"]
  }
```

```
  Process:
  - Générer prompt Suno depuis émotion
  - Call Suno API (async)
  - Retourner request_id
```

```
[ ] 2. Prompt engineering musique
  const MUSIC_PROMPTS = {
    anxiété_calming: "Soft ambient piano, 60bpm,
      gentle pads, nature sounds, peaceful",
    tristesse_comforting: "Warm acoustic guitar,
      gentle strings, 70bpm, hopeful melody",
    ...
  }
```

```
[ ] 3. Queue system musique
  CREATE TABLE music_queue (
    id UUID PRIMARY KEY,
    user_id UUID,
    suno_request_id TEXT,
    status TEXT, -- pending, processing, completed, failed
    created_at TIMESTAMPTZ,
    completed_at TIMESTAMPTZ
  );
```

```
[ ] 4. Polling status Suno
```

- Edge Function: check-music-status
- Appelé toutes les 5s par client
- Update status quand complété
- Récupérer URL audio

[ ] 5. Composant MusicPlayer

- Waveform visualisation
- Play/pause/seek
- Volume control
- Bouton: "Sauvegarder en favoris"
- Bouton: "Régénérer"

[ ] 6. Persistance résultat

```
UPDATE music_generations SET
    status = 'completed',
    audio_url = '...',
    completed_at = now()
```

[ ] 7. Analytics écoute

- Temps écoute total
- Rating 1-5 étoiles
- Play count

**Critères d'acceptation:** - [ ] Génération < 30s - [ ] Prompt adapté à l'émotion - [ ] Player audio fonctionnel - [ ] Sauvegarde favoris - [ ] Tracking analytics

---

## EPIC 7: Luminothérapie

### S2-03: Module luminothérapie

**Story:** En tant qu'utilisateur, je peux utiliser l'écran comme source de lumière thérapeutique

**Points:** 5 | **Priorité:** P1

#### Tâches techniques:

[ ] 1. Composant LightTherapyScreen

- Fullscreen mode
- Couleur de fond dynamique
- Transitions douces

[ ] 2. Presets lumière

```
const LIGHT_PRESETS = {
```

```
    calm: {
      color: '#4A90D9', // Bleu doux
      brightness: 0.7,
      rhythm: 'steady'
    },
    focus: {
      color: '#90D94A', // Vert
      brightness: 0.8,
      rhythm: 'pulse_slow'
    },
    energize: {
      color: '#FFFFFF', // Blanc
      brightness: 1.0,
      rhythm: 'pulse_fast'
    },
    sleep: {
      color: '#FF6B4A', // Orange/rouge
      brightness: 0.5,
      rhythm: 'fade_out'
    }
  }
}
```

#### [ ] 3. Animations rythme

- steady: couleur fixe
- pulse\_slow: 4s cycle
- pulse\_fast: 2s cycle
- fade\_out: diminution 10min
- sunrise: gradient 15min

#### [ ] 4. Timer session

- Durée configurable (5, 10, 15, 20 min)
- Affichage discret temps restant
- Notification fin

#### [ ] 5. Instructions utilisateur

- "Regardez doucement l'écran"
- "Clignez normalement des yeux"
- "Évitez en cas d'épilepsie"

#### [ ] 6. Persistance session

```
INSERT light_sessions (
  user_id, session_id, color_hex,
  brightness, rhythm_pattern,
  target_effect, duration_minutes
)
```

**Critères d'acceptation:** - [ ] 4 presets disponibles - [ ] Animations fluides - [ ] Mode fullscreen -  
[ ] Warning épilepsie - [ ] Timer configurable

---

## EPIC 8: Dashboard Progression

### S2-04: Visualisation progression

**Story:** En tant qu'utilisateur, je vois mon évolution émotionnelle sur 7/30 jours

**Points:** 5 | **Priorité:** P0

**Tâches techniques:**

[ ] 1. Edge Function: get-user-metrics

- Agrégation depuis user\_metrics\_daily
- Période: 7j / 30j / 90j
- Retour: KPIs + données graphiques

[ ] 2. Composant MetricsCards

- 3 cards: Bien-être, Stress, Sessions
- Valeur actuelle + delta vs période précédente
- Flèche up/down colorée

[ ] 3. Graphique évolution (Recharts)

- Line chart valence sur période
- Tooltips interactifs
- Zoom/pan sur mobile

[ ] 4. Composant EmotionBreakdown

- Pie chart émotions dominantes
- Légende interactive
- Drill-down possible

[ ] 5. Job calcul métriques quotidien

- pg\_cron: 00:05 chaque jour
- Agrège sessions jour précédent
- INSERT/UPDATE user\_metrics\_daily

[ ] 6. Cache React Query

- staleTime: 5 min
- Invalidation après nouvelle session

**Critères d'acceptation:** - [ ] KPIs 7j affichés par défaut - [ ] Graphique interactif - [ ] Comparaison période précédente - [ ] Chargement < 1s

## S2-05: Insights IA personnalisés

**Story:** En tant qu'utilisateur, je reçois des analyses et recommandations basées sur mes patterns

**Points:** 5 | **Priorité:** P1

**Tâches techniques:**

[ ] 1. Edge Function: generate-insights  
Input: user\_id, period (7d/30d)

Analyse:

- Patterns temporels (heures, jours)
- Corrélations contexte ↔ émotion
- Efficacité exercices
- Tendance générale

Output:

```
{  
  patterns: [  
    { type: 'temporal', description: 'Lundis plus difficiles' },  
    { type: 'correlation', description: 'Travail → stress' }  
  ],  
  recommendations: [  
    { text: 'Routine matinale recommandée', action: 'setReminder' }  
  ],  
  trend: 'improving' | 'stable' | 'declining'  
}
```

[ ] 2. Prompt GPT-4 pour insights

- Données anonymisées en input
- Génération texte naturel
- Suggestions actionnables

[ ] 3. Composant InsightsCard

- Titre: "Vos patterns de la semaine"
- Liste insights avec icônes
- CTA pour recommandations

[ ] 4. Job génération hebdo

- pg\_cron: dimanche 20:00
- Génère insights pour tous users actifs
- Notification push si activée

[ ] 5. Cache insights

- Stockage en BDD
- Refresh hebdo ou on-demand

**Critères d'acceptation:** - [ ] 2-4 patterns détectés - [ ] Recommandations actionnables - [ ] Génération hebdo automatique - [ ] Langage naturel et bienveillant

---

## EPIC 9: RGPD & Notifications

### S2-06: Export données utilisateur

**Story:** En tant qu'utilisateur, je peux télécharger toutes mes données personnelles

**Points:** 3 | **Priorité:** P0

**Tâches techniques:**

- [ ] 1. Edge Function: export-user-data
  - Collecte toutes tables user
  - Format JSON structuré
  - Exclusion données techniques
- [ ] 2. Structure export
 

```
{
        exported_at: "2025-01-15T10:00:00Z",
        user: { email, created_at },
        profile: { ... },
        sessions: [ ... ],
        exercises: [ ... ],
        music: [ ... ],
        consents: [ ... ]
      }
```
- [ ] 3. UI Export
  - Settings > Mes données > Exporter
  - Bouton "Télécharger JSON"
  - Génération async si > 1000 entrées

```
[ ] 4. Audit log  
INSERT audit_logs (action: 'data_export')
```

**Critères d'acceptation:** - [ ] Export complet < 30s - [ ] Format JSON lisible - [ ] Toutes données personnelles incluses - [ ] Log audit créé

---

## S2-07: Notifications push

**Story:** En tant qu'utilisateur, je reçois des rappels pour mes sessions

**Points:** 3 | **Priorité:** P1

**Tâches techniques:**

- [ ] 1. Service Web Push (VAPID)
  - Génération clés VAPID
  - Registration service worker
  - Stockage subscriptions
- [ ] 2. Permission request post-onboarding
  - Modal explicatif bénéfices
  - Bouton "Activer" / "Plus tard"
  - Persist préférence
- [ ] 3. Settings notifications
  - Toggle on/off
  - Heure rappel configurable
  - Mode silencieux (pas après 23h)
- [ ] 4. Edge Function: send-push
  - Input: user\_id, title, body
  - Récupère subscription
  - Envoie via web-push
- [ ] 5. Triggers notifications
  - Rappel quotidien (heure choisie)
  - Streak risque de casser
  - Insights hebdo prêts

**Critères d'acceptation:** - [ ] Permission demandée poliment - [ ] Heure configurable - [ ] Mode nuit respecté - [ ] Désinscription facile

---

## S2-08: Bibliothèque exercices

**Story:** En tant qu'utilisateur, je peux parcourir et choisir parmi une bibliothèque d'exercices

**Points:** 3 | **Priorité:** P1

**Tâches techniques:**

- [ ] 1. Seed exercices (10+)
  - 4 respirations (box, 4-7-8, cohérence, énergisante)
  - 3 relaxations (scan corporel, relaxation musculaire, visualisation)
  - 3 méditations (pleine conscience, gratitude, compassion)
  
- [ ] 2. Page /exercises/library
  - Grid/list view
  - Filtres: type, durée, émotion cible
  - Search
  
- [ ] 3. Card exercice
  - Nom + icône
  - Durée
  - Difficulté (dots)
  - Tags émotions ciblées
  
- [ ] 4. Preview exercice
  - Description détaillée
  - "Ce que vous allez faire"
  - Bouton "Commencer"
  
- [ ] 5. Favoris utilisateur
  - Bouton cœur sur card
  - Section "Mes favoris" en haut

**Critères d'acceptation:** - [ ] 10+ exercices disponibles - [ ] Filtrage fonctionnel - [ ] Preview avant lancement - [ ] Système favoris

---

## S2-09: Tests E2E Sprint 2

**Story:** En tant que développeur, je m'assure que les nouvelles features fonctionnent

**Points:** 3 | **Priorité:** P0

**Tâches techniques:**

- [ ] 1. Test input vocal
  - Mock MediaRecorder
  - Vérifier transcription affichée
  - Vérifier analyse émotions
  
- [ ] 2. Test génération musique
  - Mock Suno API
  - Vérifier queue créée
  - Vérifier player affiché
  
- [ ] 3. Test dashboard progression
  - Seed données test
  - Vérifier KPIs affichés
  - Vérifier graphique rendu
  
- [ ] 4. Test export données
  - Déclencher export
  - Vérifier fichier téléchargé
  - Vérifier contenu JSON

**Critères d'acceptation:** - [ ] 4 nouveaux flows couverts - [ ] Mocks API stables - [ ] CI green

---

## S2-10: Optimisation performance

**Story:** En tant qu'utilisateur, l'app se charge rapidement

**Points:** 2 | **Priorité:** P1

**Tâches techniques:**

- [ ] 1. Audit Lighthouse
  - Score Performance > 90
  - Score Accessibility > 90
  - Score Best Practices > 90
  
- [ ] 2. Lazy loading routes
  - Vérifier tous les React.lazy()
  - Suspense fallbacks appropriés
  
- [ ] 3. Bundle analysis
  - npm run build:analyze
  - Identifier chunks > 100KB
  - Split si nécessaire

- [ ] 4. Images optimisation
  - WebP/AVIF formats
  - Lazy loading images
  - Dimensions appropriées
  
- [ ] 5. API caching
  - React Query staleTime optimal
  - Prefetch routes probables

**Critères d'acceptation:** - [ ] LCP < 2.5s - [ ] TTI < 3.5s - [ ] Bundle initial < 500KB - [ ]  
Lighthouse > 90

---

## Récapitulatif Points

| Sprint | Epic                  | Points    |
|--------|-----------------------|-----------|
| S1     | Onboarding & Auth     | 8         |
| S1     | Session Émotionnelle  | 15        |
| S1     | Module Respiration    | 5         |
| S1     | Sécurité & Compliance | 15        |
|        | <b>Total Sprint 1</b> | <b>43</b> |
| S2     | Input Vocal           | 8         |
| S2     | Musicothérapie IA     | 8         |
| S2     | Luminothérapie        | 5         |
| S2     | Dashboard Progression | 10        |
| S2     | RGPD & Notifications  | 9         |
|        | <b>Total Sprint 2</b> | <b>40</b> |

---

# Dépendances critiques

S1-01 (Onboarding) —> S1-02 (Consents) —> Tous les autres

S1-03 (Input) —> S1-04 (Analyse IA) —> S1-05 (Plan)

S1-06 (Respiration) ← indépendant

S1-07/08 (Sécurité) ← indépendant, parallel

S2-01 (Vocal) dépend de S1-04 (Analyse IA)

S2-02 (Musique) dépend de S1-05 (Plan)

S2-04 (Dashboard) dépend de S1 complet

---

Document généré le 2025-12-20 - EmotionsCare Technical Backlog