

KAEB - Documentation Technique

Suivi 2 - 12 décembre 2025

Table des matières

1. Présentation de l'application
 2. Identifiants de test
 3. Procédurier des fonctionnalités
 4. Standards de développement
 5. Décisions UI/UX
 6. Guide de déploiement
-

1. Présentation de l'application

KAEB est une application mobile de productivité sociale développée avec React Native et Expo. Elle permet aux utilisateurs de : - Suivre leurs habitudes quotidiennes - Partager leurs accomplissements avec la communauté - Se concentrer avec le mode “Lock In” - Interagir avec d’autres utilisateurs via des publications

Technologies utilisées

- **Frontend** : React Native ~0.76.8, Expo ~54.0.22
 - **Backend** : Martha API (<http://martha.jh.shawinigan.info/queries/>)
 - **Base de données** : MySQL via Martha
 - **Navigation** : @react-navigation/native
 - **Stockage local** : AsyncStorage
-

2. Identifiants de test

Comptes utilisateurs avec données représentatives

Email	Mot de passe	Description
s@gmail.com	123123	Utilisateur avec publications et habitudes
d@gmail.com	123123	Utilisateur avec publications et habitudes

Données de test présentes

- **Publications** : Plusieurs posts avec images, descriptions et liens vers des habitudes
 - **Habitudes** : Habitudes de différentes fréquences (quotidienne, hebdomadaire, mensuelle)
 - **Communautés** : Plusieurs communautés disponibles pour les publications
-

3. Procédurier des fonctionnalités

3.1 Authentification

Connexion

1. Lancer l'application
2. Sur l'écran de connexion, entrer l'email et le mot de passe
3. Appuyer sur "Login"
4. L'utilisateur est redirigé vers la page d'accueil (Timeline)

Inscription

1. Depuis l'écran de connexion, appuyer sur "Register"
2. Remplir les champs requis :
 - Prénom
 - Nom
 - Email
 - Mot de passe
 - Carrière (optionnel)
3. Appuyer sur "Register"
4. L'utilisateur est connecté automatiquement

Déconnexion

1. Naviguer vers le profil (icône en haut à droite ou barre de navigation)
 2. Appuyer sur le bouton "Logout"
 3. L'utilisateur est redirigé vers l'écran de connexion
-

3.2 Timeline (Page d'accueil)

Visualiser les publications

1. Après connexion, l'écran d'accueil affiche toutes les publications
2. Faire défiler vers le bas pour voir plus de publications
3. Chaque carte de publication affiche :
 - Avatar et nom de l'auteur

- Communauté
- Description
- Image (si présente)
- Habitude liée (si présente)
- Boutons de réaction et commentaire

Accéder à une publication détaillée

1. Appuyer sur une carte de publication
 2. L'écran de détail s'ouvre avec les commentaires
-

3.3 Crédit de publication

Créer une nouvelle publication

1. Depuis la Timeline, appuyer sur l'icône “+” (cercle) en haut à gauche
2. Remplir les informations :
 - **Communauté** (obligatoire) : Sélectionner dans la liste déroulante
 - **Habitude** (optionnel) : Lier la publication à une habitude personnelle
 - **Description** (obligatoire) : Texte de la publication
 - **URL Image** (optionnel) : Coller une URL d'image
3. Appuyer sur “Post”
4. La publication est créée et l'utilisateur retourne à la Timeline

Note technique : Les images sont automatiquement converties en Base64 pour un stockage fiable dans la base de données.

3.4 Modification de publication

Modifier une publication existante

1. Ouvrir une publication personnelle depuis le profil
2. Appuyer sur l'icône de modification (crayon)
3. Modifier les champs souhaités :
 - Communauté
 - Habitude liée
 - Description
 - URL Image
4. Appuyer sur “Update”
5. Les modifications sont enregistrées

Supprimer une publication

1. Ouvrir une publication personnelle

2. Appuyer sur l'icône de suppression (poubelle)
 3. Confirmer la suppression dans la boîte de dialogue
-

3.5 Gestion des habitudes

Visualiser ses habitudes

1. Appuyer sur “Habits” dans la barre de navigation
2. La liste des habitudes personnelles s'affiche avec :
 - Nom de l'habitude
 - Fréquence (badge coloré)
 - Description
 - Icône de fréquence

Créer une habitude

1. Depuis l'écran des habitudes, appuyer sur “+”
2. Remplir les informations :
 - **Nom** (obligatoire)
 - **Fréquence** : Daily / Weekly / Monthly
 - **Description** (optionnel)
3. Appuyer sur “Create Habit”

Modifier une habitude

1. Appuyer sur l'icône “...” sur une carte d'habitude
2. Modifier les champs souhaités
3. Appuyer sur “Save Changes”

Supprimer une habitude

1. Depuis l'écran de modification d'habitude
 2. Appuyer sur “Delete Habit”
 3. Confirmer la suppression
-

3.6 Mode Lock In (Focus)

Le mode “Lock In” est une fonctionnalité de concentration qui détecte si l'utilisateur quitte l'application.

Démarrer une session Lock In

1. Appuyer sur “Lock In” dans la barre de navigation
2. Sélectionner une durée parmi les options :
 - 15 minutes

- 30 minutes
 - 45 minutes
 - 1 heure
 - 90 minutes
3. Appuyer sur “Start”
 4. Le minuteur démarre

Comportement pendant la session

- **Rester dans l'app** : Le minuteur continue normalement
- **Quitter l'app ou changer d'onglet** : La session échoue immédiatement
- **Écran d'échec** : Animation de secousse avec message “You failed!”
- **Écran de succès** : Animation de confettis avec message “Congratulations!”

Réinitialiser après une session

1. Appuyer sur “Try Again” (après échec) ou “Do it Again” (après succès)
 2. Sélectionner une nouvelle durée
 3. Recommencer
-

3.7 Profil utilisateur

Visualiser son profil

1. Appuyer sur l'avatar en haut à droite ou sur “Profile” dans la barre de navigation
2. Le profil affiche :
 - Photo de profil
 - Nom complet
 - Carrière
 - Liste des publications personnelles

Modifier son profil

1. Depuis le profil, appuyer sur l'icône engrenage (paramètres) en haut à droite
2. Modifier les champs :
 - **Prénom**
 - **Nom**
 - **Carrière**
 - **URL Avatar** : Coller une URL d'image pour changer la photo de profil
3. Appuyer sur “Save Changes”

Note technique : L'avatar est automatiquement converti en Base64 pour un stockage fiable.

Visualiser le profil d'un autre utilisateur

1. Appuyer sur l'avatar d'un auteur sur une publication
 2. Le profil public s'affiche avec les publications de cet utilisateur
-

3.8 Commentaires et Réactions

Ajouter un commentaire

1. Ouvrir une publication (appuyer sur la carte)
2. Écrire un commentaire dans le champ texte
3. Appuyer sur "Post"

Réagir à une publication

1. Appuyer sur l'icône de cœur sur une carte de publication
 2. La réaction est enregistrée
-

4. Standards de développement

4.1 Structure des fichiers

```
code
  App.js          # Point d'entrée
  index.js        # Configuration Expo
  package.json    # Dépendances
  components/
    BottomBar.js # Barre de navigation inférieure
    CustomButton.js # Bouton personnalisé
    Dropdown.js   # Liste déroulante (iOS compatible)
    PostCard.js   # Carte de publication
  config/
    api.js        # Fonctions API Martha
    theme.js       # Styles et couleurs
  context/
    AuthContext.js # Contexte d'authentification
  navigation/
    AppNavigator.js # Configuration des routes
  screens/
    Focus/
      LockInScreen.js
    Habits/
      AddHabitScreen.js
      EditHabitScreen.js
      HabitTrackerScreen.js
```

```

Home/
  HomeScreen.js
Login/
  LoginScreen.js
  RegisterScreen.js
Post/
  CreatePostScreen.js
  EditPostScreen.js
  PostScreen.js
Profile/
  EditProfileScreen.js
  ProfileScreen.js
  ViewProfileScreen.js

```

4.2 Conventions de nommage

Élément	Convention	Exemple
Fichiers écrans	PascalCase + Screen	HomeScreen.js
Fichiers composants	PascalCase	PostCard.js
Fonctions	camelCase	fetchPosts()
Variables	camelCase	userId, selectedCommunity
Constantes	UPPER_SNAKE_CASE	AUTH_TOKEN, TIME_OPTIONS
Styles	camelCase	cardHeader, frequencyBadge

4.3 Gestion d'état

- **État local** : useState pour les données de composant
- **Persistance** : AsyncStorage pour les données utilisateur
- **Contexte** : AuthContext pour l'authentification globale

4.4 API Martha

Toutes les requêtes API passent par la fonction marthaFetch dans config/api.js :

```

const marthaFetch = async (queryName, params = {}) => {
  const response = await fetch(
    "http://martha.jh.shawinigan.info/queries/" + queryName,
    {
      method: "POST",
      headers: {
        "Authorization": `Bearer ${AUTH_TOKEN}`,
        "Content-Type": "application/x-www-form-urlencoded",
      },
      body: new URLSearchParams(params).toString(),
    }
  )
  return response.json();
}

```

```

        }
    );
    return await response.json();
};


```

4.5 Fonctions utilitaires

Fonction	Description
<code>sanitizeForSQL(str)</code>	Échappe les caractères spéciaux SQL
<code>urlToBase64(url)</code>	Convertit une URL d'image en Base64
<code>getPostId(post)</code>	Récupère l'ID d'un post (p_id ou id)
<code>getUserIds(user)</code>	Récupère l'ID d'un utilisateur
<code>getHabitId(habit)</code>	Récupère l'ID d'une habitude
<code>getCommunityId(community)</code>	Récupère l'ID d'une communauté

5. Décisions UI/UX

5.1 Palette de couleurs

Couleur	Code hex	Utilisation
Primary	#2A9D8F	Boutons principaux, accents
Background	#121212	Fond des écrans (mode sombre)
Card	#1E1E1E	Fond des cartes
Text	#FFFFFF	Texte principal
Gray	#888888	Texte secondaire
Error	#FF6B6B	Messages d'erreur

5.2 Thème sombre

L'application utilise exclusivement un thème sombre pour : - Réduire la fatigue oculaire - Économiser la batterie sur écrans OLED - Offrir une expérience moderne et cohérente

5.3 KeyboardAvoidingView

Tous les écrans avec des champs de saisie utilisent `KeyboardAvoidingView` pour éviter que le clavier ne cache les champs : - `CreatePostScreen` - `EditPostScreen` - `AddHabitScreen` - `EditHabitScreen` - `EditProfileScreen`

5.4 Composant Dropdown personnalisé

Un composant `Dropdown` personnalisé a été créé pour remplacer le `Picker` natif qui ne fonctionne pas correctement sur iOS. Ce composant : - Affiche un modal

avec une liste de choix - Fonctionne de manière identique sur iOS et Android - Applique les styles du thème sombre

5.5 Conversion Base64 des images

Les images sont converties en Base64 avant stockage pour : - Éviter les problèmes de liens cassés - Garantir la persistance des images - Simplifier la gestion du stockage

5.6 Animations

- **HabitCard** : Animation de fondu au chargement (fade in + slide up)
 - **LockIn** : Animation de verrouillage, secousse (échec), confettis (succès)
-

6. Guide de déploiement

6.1 Prérequis

- Node.js (version 18+)
- npm ou yarn
- Expo CLI (`npm install -g expo-cli`)
- Un appareil mobile avec Expo Go ou un émulateur

6.2 Installation du projet

```
# 1. Cloner ou télécharger le projet
cd /chemin/vers/Final-ProjetSession/code
```

```
# 2. Installer les dépendances
npm install
```

```
# 3. Lancer l'application en mode tunnel
npx expo start --tunnel
```

6.3 Configuration de la base de données Martha

Note : Le fichier SQL (dump) de la base de données est inclus dans le même dossier que ce document PDF.

Tables requises Users

```
CREATE TABLE Users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(100),
    lastname VARCHAR(100),
    email VARCHAR(255) UNIQUE,
```

```

    password VARCHAR(255),
    career VARCHAR(100),
    avatar_url LONGTEXT
);



### Posts



```

CREATE TABLE Posts (
 id INT AUTO_INCREMENT PRIMARY KEY,
 user_id INT,
 community_id INT,
 habit_id INT,
 description TEXT,
 image_url LONGTEXT,
 created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 FOREIGN KEY (user_id) REFERENCES Users(id),
 FOREIGN KEY (community_id) REFERENCES Communities(id),
 FOREIGN KEY (habit_id) REFERENCES Habits(id)
);

```



### Habits



```

CREATE TABLE Habits (
 id INT AUTO_INCREMENT PRIMARY KEY,
 user_id INT,
 name VARCHAR(255),
 description TEXT,
 frequency VARCHAR(50),
 FOREIGN KEY (user_id) REFERENCES Users(id)
);

```



### Communities



```

CREATE TABLE Communities (
 id INT AUTO_INCREMENT PRIMARY KEY,
 name VARCHAR(255),
 description TEXT
);

```



### Comments



```

CREATE TABLE Comments (
 id INT AUTO_INCREMENT PRIMARY KEY,
 post_id INT,
 user_id INT,
 content TEXT,
 created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
 FOREIGN KEY (post_id) REFERENCES Posts(id),
 FOREIGN KEY (user_id) REFERENCES Users(id)
);

```


```

Reactions

```
CREATE TABLE Reactions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    post_id INT,
    user_id INT,
    FOREIGN KEY (post_id) REFERENCES Posts(id),
    FOREIGN KEY (user_id) REFERENCES Users(id)
);
```

6.4 Requêtes Martha nécessaires

Nom de la requête	Paramètres	Description
login	email, password	Authentification
register	firstname, lastname, email, password, career	Inscription
select-posts	-	Liste toutes les publications
select-user-posts	user_id	Publications d'un utilisateur
create-post	user_id, community_id, description, image_url, habit_id	Créer une publication
update-post	post_id, community_id, description, image_url, habit_id	Modifier une publication
delete-post	post_id	Supprimer une publication
select-communities	-	Liste des communautés
select-user-habits	user_id	Habitudes d'un utilisateur
create-habit	user_id, name, description, frequency	Créer une habitude
update-habit	habit_id, name, description, frequency	Modifier une habitude
delete-habit	habit_id	Supprimer une habitude

Nom de la requête	Paramètres	Description
<code>update-profile-details</code>	<code>user_id, firstname, lastname, career, avatar_url</code>	Modifier le profil
<code>select-user-by-id</code>	<code>user_id</code>	Récupérer un utilisateur
<code>select-comments</code>	<code>post_id</code>	Commentaires d'un post
<code>create-comment</code>	<code>post_id, user_id, content</code>	Ajouter un commentaire
<code>create-reaction</code>	<code>post_id, user_id</code>	Ajouter une réaction
<code>delete-reaction</code>	<code>post_id, user_id</code>	Supprimer une réaction

6.5 Configuration API

Dans `config/api.js`, assurez-vous que le token d'authentification Martha est configuré :

```
const AUTH_TOKEN = "votre_token_martha";
```

6.6 Test de l'application

1. Lancer `npx expo start --tunnel`
 2. Scanner le QR code avec l'application Expo Go
 3. Se connecter avec un compte de test
 4. Tester toutes les fonctionnalités du procédurier
-

Annexes

Dépendances principales (`package.json`)

```
{
  "dependencies": {
    "expo": "~54.0.22",
    "react": "18.3.1",
    "react-native": "0.76.8",
    "@react-navigation/native": "^7.0.19",
    "@react-navigation/stack": "^7.2.0",
    "@react-native-async-storage/async-storage": "^2.1.2",
    "react-native-safe-area-context": "5.0.0",
    "@expo/vector-icons": "14.0.2"
  }
}
```

Contact

Équipe KAEB - Projet de session Cegep Shawinigan - Décembre 2025