



Rapport

Café sans-fil: Évaluation et développement du segment social

Eed Flory Jean-Baptiste (20168335)
IFT 3150 – Projet Informatique

Hiver 2024

Table des matière

Introduction.....	3
Contexte.....	3
Problématique.....	3
Proposition.....	3
Analyse.....	4
Contexte.....	4
Méthodologie.....	4
Résultats et observations.....	4
Besoins fonctionnels.....	4
Besoins non fonctionnels.....	5
Risques.....	5
Choix technologique.....	6
Conception.....	7
Prototypage.....	7
Architecture des composants.....	9
Implémentation.....	11
Développement Backend.....	11
Difficultés rencontrés.....	12
Tests.....	13
Tests fonctionnels.....	13
Tests utilisateur.....	13
Documentation.....	14
Conclusion.....	15
Remerciements.....	15
Bibliographie.....	16
Annexes.....	17
Annexe 1: Application version 0.2.0.....	17
Annexe 2: Prototype de la version 0.2.1.....	19
Annexe 3: Capture d'écran de l'application (version 0.2.1).....	21
Annexe 4: Diagramme de dépendances des composants de l'application.....	22

Introduction

Contexte

Afin d'améliorer le service des cafés étudiants de l'Université de Montréal (UdeM), le projet Café sans-fil a été initié au trimestre d'automne dernier. Ce projet vise à établir une plateforme unifiée pour les cafés étudiants, ainsi qu'une source d'informations fiable pour la communauté universitaire. À l'automne l'infrastructure de la plateforme, comprenant une API et une base de données MongoDB, et un premier prototype de l'application (version 0.2.0) ont été conçus.

Problématique

Bien que la solution développée à l'automne semble répondre aux besoins initiaux identifiés, plusieurs points restent à éclaircir. Premièrement, le manque d'évaluation systématique du prototype développé à l'automne a limité la compréhension des besoins réels des utilisateurs et de l'efficacité de la solution proposée. Deuxièmement, les fonctionnalités développées se concentrent principalement sur la vente des produits proposés par les cafés, ignorant les aspects sociaux qui pourraient enrichir l'expérience des utilisateurs en facilitant les interactions et le partage d'informations au sein de la communauté universitaire. Ces lacunes dans l'évaluation et l'engagement social compromettent la pertinence et l'utilité de l'application pour les étudiants et les bénévoles des cafés.

Proposition

Pour adresser ces enjeux, une phase d'évaluation a été proposée pour approfondir le prototype Café sans-fil, en mettant un accent particulier sur la collecte de retours d'utilisateurs pour affiner les fonctionnalités existantes et identifier les nouvelles nécessités. Cette évaluation comprend des tests utilisateurs, des sondages et des groupes de discussion pour recueillir des données précises sur l'expérience utilisateur.

Parallèlement, il est convenu de développer le segment social de l'application. Ce développement vise à intégrer des fonctionnalités permettant aux utilisateurs de partager des événements et d'interagir de manière plus dynamique.

Ces améliorations sont destinées à transformer Café sans-fil en un véritable carrefour social et informatif pour la communauté universitaire, en rendant la navigation plus intuitive et en augmentant l'engagement des utilisateurs. Cette approche aidera non seulement à améliorer l'expérience utilisateur, mais aussi à renforcer la gestion quotidienne des cafés, rendant le service plus adaptatif et réactif aux besoins changeants de la communauté universitaire. Au final, cela a mené à créer la nouvelle version 0.2.1 de café sans-fil, disponible sur GitHub¹.

¹ Branche de développement: https://github.com/ceduni/cafe-sans-fil/tree/Flory_v2

Analyse

Contexte

Une analyse approfondie des versions mobile et desktop de l'application Café sans-fil a été réalisée, avec l'objectif de recueillir des données essentielles sur l'utilisation actuelle de l'application par les testeurs. Cette démarche visait à évaluer la réception des fonctionnalités existantes et à identifier les domaines où des améliorations pourraient rendre l'expérience utilisateur plus agréable et efficace.

Méthodologie

Pour cette analyse, des entrevues dédiées ont été menées pour chaque version (mobile et *desktop*) afin de capturer des informations critiques concernant les fonctionnalités actuelles et les préférences des utilisateurs. Une méthode particulière utilisée durant ces entrevues a été la création d'un formulaire², conçu pour structurer la collecte des retours et faciliter l'analyse des données.

L'approche consistait à écouter activement les utilisateurs pour comprendre leurs expériences, leurs défis et leurs suggestions d'amélioration. Cela a permis de déterminer les améliorations nécessaires et les nouvelles fonctionnalités à intégrer pour répondre aux besoins spécifiques des utilisateurs. Cette stratégie a renforcé l'efficacité de la collecte de données, contribuant à une analyse plus approfondie et à une meilleure compréhension des attentes des utilisateurs.

Résultats et observations

Les résultats de l'enquête ont montré une réception globalement positive des fonctionnalités de la première version de l'application. Cependant, les utilisateurs ont également signalé plusieurs points d'amélioration pour rendre l'expérience plus fluide, notamment des interfaces plus intuitives et des fonctionnalités plus réactives.

De plus, de nouveaux besoins ont été identifiés pour enrichir l'application, tels que des fonctionnalités supplémentaires qui pourraient améliorer la connectivité et l'engagement entre les utilisateurs.

Besoins fonctionnels

Ci-dessous, voici une liste mise à jour des fonctionnalités recherchées dans la nouvelle version.

Grand Public

- ★ Voir la liste des cafés
- ★ Voir les informations d'un café
- ★ Voir le menu d'un café
- ★ Voir les annonces d'un café
- ★ Voir les événements attachés à un café
- ★ Chercher un café par nom, par item ou par tag accompagné de l'autocomplétion
- ★ Chercher un café à l'aide de filtre
- ★ Retrouver les cafés en utilisant la géolocalisation

² Formulaire utilisé pour les entretiens: <https://cgwrmfp5qw7.typeform.com/to/Drh5e3Ff>

Membre

- ★ Se créer un compte : ceci nécessite un matricule et une adresse courriel de l'UdeM
- ★ Se connecter avec son matricule ou son adresse courriel
- ★ Voir et modifier son profil
- ★ Passer ou annuler une commande (pour ramassage)
- ★ Voir l'historique de ces commandes
- ★ Assigner une note aux items du menu
- ★ Réagir aux annonces et évènements à l'aide de réactions prédéfinies
- ★ Soumettre un évènement

Gérant

- ★ Modifier les informations de base d'un café telles que nom, description, horaire, réseaux sociaux
- ★ Ajouter, modifier ou supprimer un bénévole (staff)
- ★ Générer des rapports de vente : journalières, hebdomadaires, mensuelles
- ★ Ajouter, modifier ou supprimer un item du menu
- ★ Ajouter, modifier ou supprimer un événement
- ★ Ajouter, modifier ou supprimer une annonce

Besoins non fonctionnels

Afin d'offrir une très bonne expérience utilisateur (UX) et faciliter le développement de l'application, il y a eu considération des besoins suivants :

- **Performance** : Le maintien de l'environnement de développement existant avec l'utilisation de Vite a permis des mises à jour en temps réel grâce à la fonctionnalité de "hot-reloading", améliorant significativement la réactivité de l'application. Cette continuité dans l'environnement a également réduit les temps de chargement, contribuant à une expérience utilisateur plus fluide et rapide.
- **Sécurité** : Les données des utilisateurs doivent être stockées de manière sécurisée, et l'application doit prévenir tout accès non autorisé.
- **Utilisabilité** : L'interface utilisateur doit être intuitive, facile à utiliser, esthétiquement plaisante et compatible avec différents formats d'écrans. Elle doit être accessible aux personnes ayant des handicaps mineurs, et plus particulièrement celles utilisant un lecteur d'écran.
- **Maintenabilité** : La décision de conserver le même environnement de développement a facilité la gestion des dépendances via un environnement virtuel Python, assurant une base stable pour le développement backend. Cette approche a minimisé les risques de conflits ou de problèmes de compatibilité lors des mises à jour, permettant ainsi à l'équipe de se concentrer sur l'amélioration continue de l'application sans interruptions.

Risques

J'ai identifié plusieurs risques associés à la poursuite du développement de l'application Café sans-fil, en mettant un accent particulier sur l'intégration des nouvelles fonctionnalités et l'amélioration des existantes.

Compréhension des Besoins des Utilisateurs

Une mauvaise compréhension des besoins des utilisateurs peut mener à une faible adoption de la

plateforme. Les retours obtenus à travers des tests utilisateurs approfondis pour les versions mobile et desktop, avec des formulaires structurés lors des entrevues, ont amélioré notre compréhension des attentes des utilisateurs et permis d'ajuster les fonctionnalités en conséquence.

Intégration et Adoption du Segment Social

Le développement et l'intégration du segment social pourraient ne pas répondre aux attentes des utilisateurs ou être sous-utilisés, réduisant ainsi l'impact souhaité sur l'engagement communautaire.

Pour minimiser ce risque, j'ai planifié des sessions de feedback ciblées et des tests d'usabilité spécifiquement pour les nouvelles fonctionnalités sociales. Cela permettra d'ajuster ces fonctionnalités basées sur les réactions directes des utilisateurs avant le déploiement complet.

Amélioration de la Recherche

Les améliorations apportées aux fonctions de recherche et de filtration pourraient ne pas fonctionner comme prévu, ce qui pourrait entraîner une frustration chez les utilisateurs.

Des tests fonctionnels et d'usabilité seront effectués pour s'assurer que ces fonctionnalités sont intuitives et répondent bien aux besoins des utilisateurs. Le feedback des utilisateurs durant la phase de test initial aidera à identifier et à résoudre les problèmes avant le lancement général.

Choix technologique

Pour le développement continu de l'application Café sans-fil cette session d'hiver, j'ai maintenu l'utilisation du FARM stack, qui comprend FastAPI, React, et MongoDB. Ce choix de continuité repose sur l'expérience positive de l'automne passé, où ces technologies ont prouvé leur efficacité, leur robustesse et leur flexibilité.

La continuation avec **FastAPI** est stratégique, surtout en raison de sa compatibilité avec Python, un langage largement enseigné à l'Université de Montréal. Cette compatibilité facilite l'engagement des étudiants et d'autres développeurs universitaires dans le projet. FastAPI continue de se démarquer pour sa haute performance et sa facilité de mise en œuvre, caractéristiques essentielles pour gérer les interactions en temps réel et assurer une expérience utilisateur fluide.

React a aussi été gardé comme choix pour le développement front-end en raison de son architecture basée sur les composants, ce qui facilite la maintenance et l'évolution du code. La popularité de React garantit un vaste écosystème de ressources et de support, permettant une intégration et une mise à jour aisées des fonctionnalités par les futurs développeurs.

La base de données **MongoDB** continue de répondre aux besoins du projet grâce à sa flexibilité structurelle et sa capacité à gérer efficacement une grande diversité de données. Cette flexibilité est cruciale pour s'adapter aux différents besoins des cafés étudiants dispersés à travers le campus, permettant des modifications et des ajouts de données sans contraintes structurelles majeures.

L'utilisation de **TailwindCSS** est reconduite pour sa capacité à créer des interfaces utilisateurs légères et hautement personnalisables. Sa structure basée sur des utilities rend le code facile à ajuster et à maintenir, ce qui est idéal pour un projet évolutif comme Café sans-fil.

Conception

Prototypage

Dans le cadre de l'amélioration continue de l'application Café sans-fil, j'ai adopté une approche itérative pour développer quatre prototypes distincts. Ces prototypes ont été conçus pour intégrer les retours des utilisateurs et répondre aux besoins spécifiques des cafés et de leur clientèle. Chaque prototype explorait diverses améliorations de fonctionnalité et d'esthétique, visant à perfectionner l'expérience utilisateur à chaque itération. Intégration des retours utilisateurs et des exigences. Les retours des utilisateurs et des gérants de café ont été essentiels pour identifier les fonctionnalités requises et les améliorations nécessaires. J'ai intégré ces éléments dans les prototypes de manière ciblée, afin de répondre au mieux aux attentes de nos utilisateurs.

Exigences générales retenues

1. **Recherche d'items dans le menu** : Incorporer une fonction de recherche rapide pour les items du menu.
2. **Taille des Images (Version Web)** : Réduire la taille des images du menu pour faciliter la navigation, surtout avec l'augmentation des items.
3. **Recherche par Item** : Permettre la recherche d'items spécifiques dans la barre de recherche, pas seulement les noms de cafés.
4. **Affichage des commandes terminées** : Simplifier l'affichage en montrant le nom du café et le total de la commande, avec une option pour voir les détails en cliquant dessus.

Exigences Spécifiques à la version mobile

1. **Clarté de l'horaire du café** : Rendre l'horaire du café plus clair en affichant les jours de la semaine en ligne plutôt qu'en colonne.
2. **Options de tri et filtres** : Ajouter des options de tri (par exemple, prix croissant/décroissant) et un filtre pour afficher uniquement les produits en stock.
3. **Modification des informations du compte** : Permettre aux utilisateurs de modifier leur nom/prénom en cas d'erreur ou de changement.
4. **Alignment des logos** : Assurer l'alignement des logos dans les informations du café.

J'ai conçu deux types de prototypes : jetables et évolutifs, chacun servant des objectifs différents dans le processus de développement.

Prototypes jetables

Ces prototypes ont été utilisés pour tester rapidement des idées et des fonctionnalités spécifiques sans l'intention de les intégrer directement dans la version finale de l'application. Ils me permettaient d'explorer des options créatives et de fonctionnalité, évaluant leur impact et leur acceptation sans risques majeurs pour la stabilité de l'application existante.

Prototypes évolutifs

Contrairement aux prototypes jetables, les prototypes évolutifs, faits avec du HTML/CSS/Javascript ont été conçus pour être améliorés et intégrés progressivement dans l'application. Cette approche a

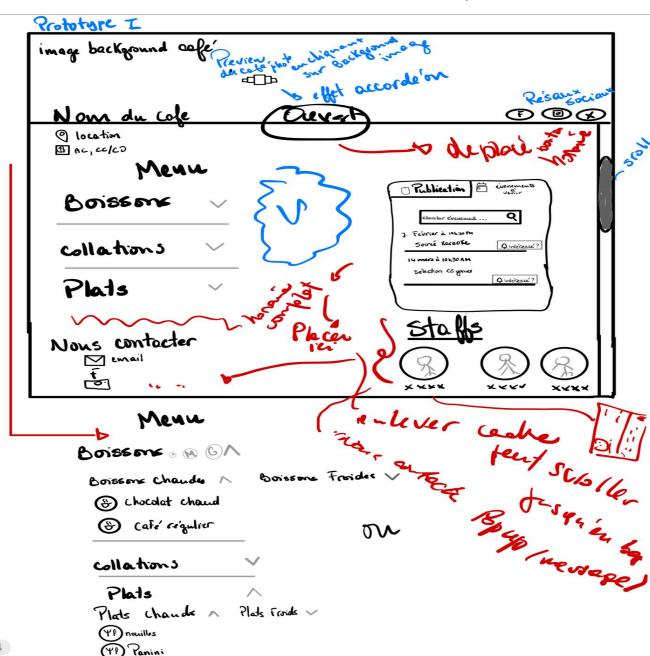
permis d'incorporer les retours des utilisateurs de manière continue et de faire évoluer les prototypes en versions de plus en plus complètes et fonctionnelles. Ces prototypes ont tendance à devenir la base sur laquelle la version finale de l'application est construite. [Voir annexe 2](#).

Sélection et raffinement du prototype

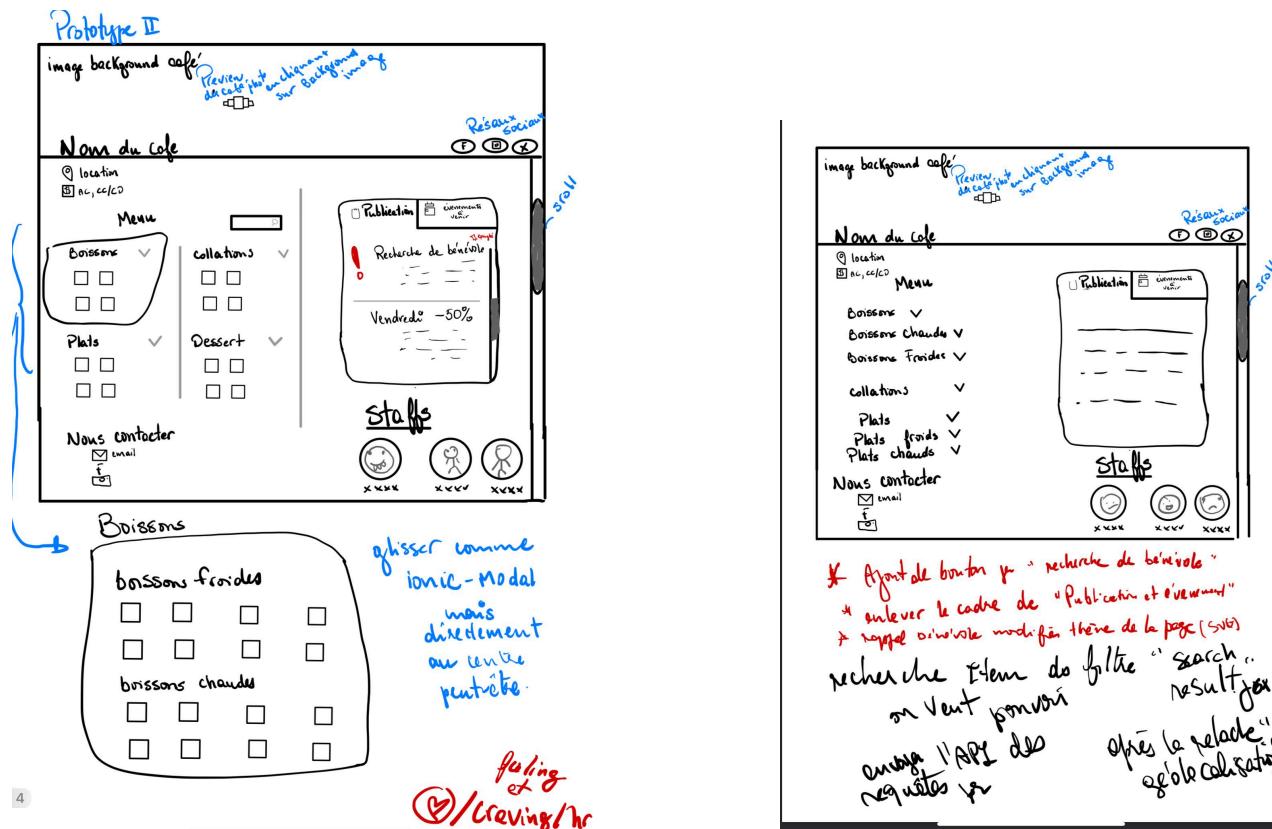
Un processus de sélection rigoureux a été mis en place, impliquant le gérant du cafés Tore et Fraction pour choisir le prototype le plus adapté. Cette collaboration a assuré l'alignement des choix de conception avec les besoins réels des utilisateurs finaux et a renforcé notre partenariat avec les parties prenantes clés.

Description des Prototypes

- Prototype I :** Axé sur une refonte complète de la page du café, intégrant des éléments sociaux comme les annonces et les événements, et révisant le menu pour améliorer la navigation.



- Prototype II, III et prototype de recherche avancée :** Ces versions ont testé différentes conceptions de layout et l'intégration de fonctionnalités de recherche avancée et de réseautage social qui permettaient aux utilisateurs de créer et partager des événements, visant à augmenter l'engagement des utilisateurs et le temps passé sur l'application.



Architecture des composants

La structure de l'application web, suit une structure standard de projet React. Le point d'entrée principal est `main.jsx`, où les routes de l'application sont définies. À ce niveau, plusieurs dossiers courants de React sont présents:

- **assets** : Ce dossier contient les ressources statiques telles que les icônes utilisées dans l'application.
- **components** : C'est dans ce dossier que sont regroupés les composants React réutilisables. Chaque composant est conçu pour accomplir une tâche spécifique, favorisant la modularité du code. Ils sont regroupés en sous-catégories en fonction de leur utilité, par exemple : `Cafe`, `Items`, `Search`, `Orders`, `events`, `annonces`. Dès qu'une partie d'une page devient complexe, ou est réutilisée dans plusieurs pages, elle est transformée en composant séparé, afin de séparer la logique et avoir un code plus clair.
- **helpers** : Des fonctions logiques sont regroupées ici. Ces `helpers` sont des modules réutilisables facilitant la cohérence et la lisibilité du code. Il peut s'agir de la logique permettant de restreindre certaines pages aux gérants ou membres d'un café seulement, ou gérer les requêtes authentifiées à l'API.
- **hooks** : Les `hooks` personnalisés sont regroupés dans ce dossier. Les `hooks` permettent de réutiliser de la logique d'état dans les composants fonctionnels et sont une partie intégrante

de React. Il peut s'agir de logique pour gérer un compte à rebours de 1h avant d'annuler une commande, ou changer l'UI en fonction de si un élément est visible sur l'écran de l'utilisateur.

- *routes* : Ce dossier contient les définitions des routes de l'application. Chaque route est associée à un composant spécifique à afficher lorsqu'elle est atteinte. Il y donc ici toutes les pages qui ont leur propre URL, tel que *Home*, *Cafe*, *Login*, *Signup*, *events*, *announcements*, etc.
- *utils* : Les utilitaires généraux, tels que des fonctions permettant de manipuler des dates, des horaires, ou définir des constantes, sont stockés ici. Ces utilitaires permettent de réutiliser des valeurs fixes ou des petites fonctions très simples dans plusieurs composants.

Cette organisation structurelle facilite la maintenance, la compréhension, et favorise l'évolution de l'application en suivant les meilleures pratiques de développement React. Chaque dossier joue un rôle défini, permettant aux développeurs de travailler de manière collaborative et efficace sur différentes parties de l'application. Voir [annexe 4](#).

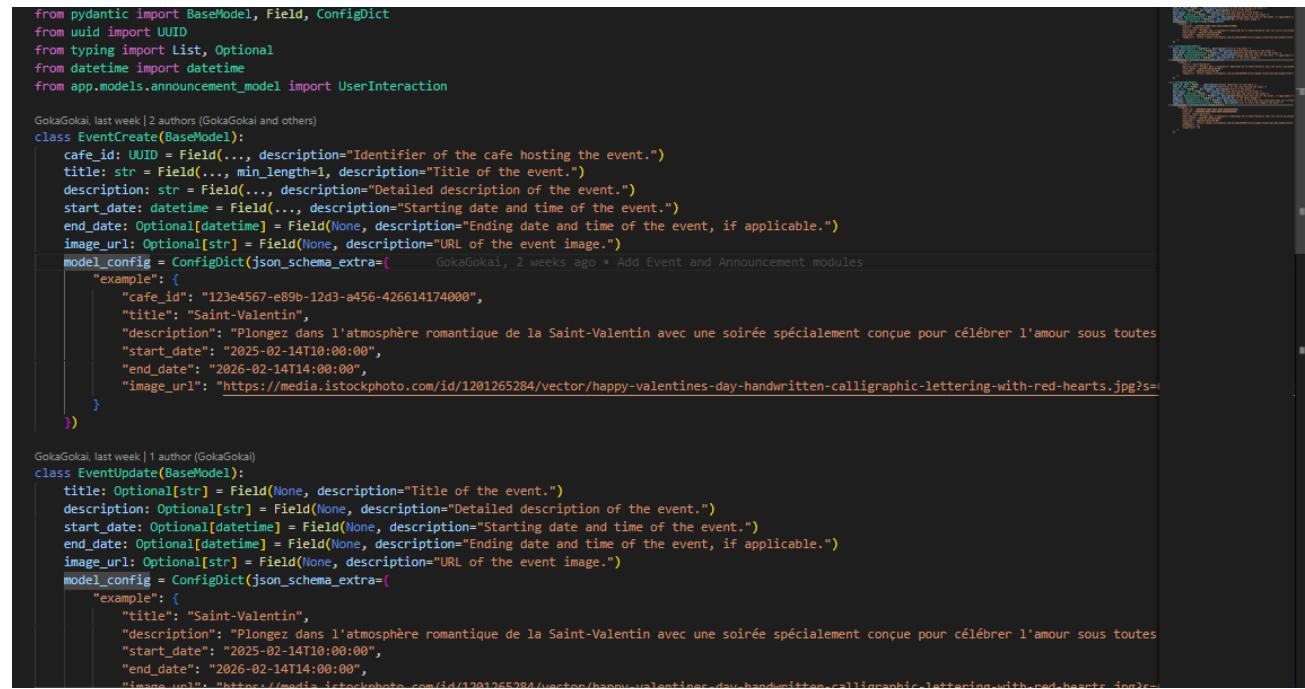
Implémentation

Développement Backend

Pour le backend, j'ai amélioré le composant de recherche, [search.py](#), en y intégrant des fonctionnalités de recherche par tag et item de menu, ce qui permet une navigation plus intuitive et ciblée. Les modifications incluent des ajustements dans [search.schema](#) et [search.service](#) pour mieux répondre aux requêtes du frontend. En parallèle, des composants pour gérer les événements ([event.py](#), [event.schema](#), et [eventmodel.py](#)) et les annonces ont été développés sur le même modèle, augmentant ainsi la modularité et la facilité de maintenance du code.

Intégration Frontend-Backend

L'implémentation réelle de la fonctionnalité de recherche dans l'application a nécessité des ajustements continus pour assurer que le frontend pouvait appeler le backend efficacement, en tenant compte des spécificités des requêtes et des réponses attendues.



```
from pydantic import BaseModel, Field, ConfigDict
from uuid import UUID
from typing import List, Optional
from datetime import datetime
from app.models.announcement_model import UserInteraction

GokaGokai, last week | 2 authors (GokaGokai and others)
class EventCreate(BaseModel):
    cafe_id: UUID = Field(..., description="Identifier of the cafe hosting the event.")
    title: str = Field(..., min_length=1, description="Title of the event.")
    description: str = Field(..., description="Detailed description of the event.")
    start_date: datetime = Field(..., description="Starting date and time of the event.")
    end_date: Optional[datetime] = Field(None, description="Ending date and time of the event, if applicable.")
    image_url: Optional[str] = Field(None, description="URL of the event image.")

    model_config = ConfigDict(json_schema_extra={
        "example": {
            "cafe_id": "123e4567-e89b-12d3-a456-426614174000",
            "title": "Saint-Valentin",
            "description": "Plongez dans l'atmosphère romantique de la Saint-Valentin avec une soirée spécialement conçue pour célébrer l'amour sous toutes",
            "start_date": "2025-02-14T10:00:00",
            "end_date": "2026-02-14T14:00:00",
            "image_url": "https://media.istockphoto.com/id/1201265284/vector/happy-valentines-day-handwritten-calligraphic-lettering-with-red-hearts.jpg?size=1200x1200"
        }
    })

GokaGokai, last week | 1 author (GokaGokai)
class EventUpdate(BaseModel):
    title: Optional[str] = Field(None, description="Title of the event.")
    description: Optional[str] = Field(None, description="Detailed description of the event.")
    start_date: Optional[datetime] = Field(None, description="Starting date and time of the event.")
    end_date: Optional[datetime] = Field(None, description="Ending date and time of the event, if applicable.")
    image_url: Optional[str] = Field(None, description="URL of the event image.")

    model_config = ConfigDict(json_schema_extra={
        "example": {
            "title": "Saint-Valentin",
            "description": "Plongez dans l'atmosphère romantique de la Saint-Valentin avec une soirée spécialement conçue pour célébrer l'amour sous toutes",
            "start_date": "2025-02-14T10:00:00",
            "end_date": "2026-02-14T14:00:00",
            "image_url": "https://media.istockphoto.com/id/1201265284/vector/happy-valentines-day-handwritten-calligraphic-lettering-with-red-hearts.jpg?size=1200x1200"
        }
    })
```

Projet Café sans-fil

```
import React, { useState, useEffect } from "react";
import { useParams, useNavigate } from "react-router-dom";
import Container from "@/components/Container";
import useApi from "@/hooks/useApi";
import { Helmet } from "react-helmet-async";
import Breadcrumbs from "@/components/Breadcrumbs";
import LoadingSpinner from "@/components>LoadingSpinner";
import AdminOnly from "@/helpers/AdminOnly";
import EventCard from "./Board/EventCard";
import toast from 'react-hot-toast';
import { Link } from "react-router-dom";
import authenticatedRequest from "@/helpers/authenticatedRequest";

const EditEvent = () => {
  const { id: cafeSlug } = useParams();
  const navigate = useNavigate();
  const { data: cafeData, isLoading: cafeIsLoading, error: cafeError } = useApi('/cafes/${cafeSlug}');
  const { data: eventsData, isLoading: eventsIsLoading, error: eventsError, refetch: refetchEvents } = useApi('/events/');
  const [events, setEvents] = useState([]);
  const [eventData, setEventData] = useState({
    cafe_id: '',
    title: '',
    description: '',
    start_date: '',
    end_date: '',
    image_url: ''
  });

  useEffect(() => {
    if (cafeData) {
      setEventData(prevState => ({
        ...prevState,
        cafe_id: cafeData.cafe_id,
      }));
    }
  }, [cafeData]);

  useEffect(() => {
    if (eventsData) {
      setEventData(prevState => ({
        ...prevState,
        title: eventsData[0].title,
        description: eventsData[0].description,
        start_date: eventsData[0].start_date,
      }));
    }
  }, [eventsData]);
}

export default EditEvent;
```

Difficultés rencontrés

L'implémentation des nouvelles fonctionnalités dans l'application Café sans-fil a présenté plusieurs défis, notamment la compréhension du codebase existant, l'intégration de nouveaux composants, et l'apprentissage approfondi des technologies utilisées comme React et TailwindCSS, ainsi que la maîtrise de FastAPI pour le backend.

J'ai rencontré des difficultés initiales pour comprendre la structure du code précédent, savoir comment ajouter et séparer les composants que j'utiliserais, ainsi que pour naviguer dans les complexités de React et TailwindCSS.

Pour surmonter ces obstacles, j'ai adopté plusieurs stratégies efficaces :

Communication avec les anciens Collaborateurs: J'ai pris contact avec les anciens développeurs de Café sans-fil pour discuter du codebase qu'ils avaient créé. Cette communication directe m'a permis de mieux comprendre les choix de conception et les méthodologies qu'ils avaient appliqués.

Ressources éducatives: Pour améliorer ma compréhension de React et de TailwindCSS, j'ai consulté des vidéos éducatives et suivi des tutoriels qui m'ont aidé à apprêhender les nuances de ces technologies.

Documentation officielle et API : J'ai régulièrement consulté la documentation officielle de React, TailwindCSS, et FastAPI, ainsi que la documentation de l'API de l'application pour comprendre comment les routes ont été construites et comment les utiliser efficacement dans le développement frontend et backend. Cela m'a aidé à résoudre des problèmes spécifiques et à apprendre à utiliser

Projet Café sans-fil

efficacement ces outils dans le contexte de mon projet.

Grâce à ces approches, j'ai non seulement réussi à intégrer les fonctionnalités nécessaires dans l'application mais aussi à enrichir significativement mon expérience de développement. Les fonctionnalités récemment implémentées, comme la publication d'événements et d'annonces, ont déjà commencé à apporter une dimension sociale enrichie à l'application. Ces aspects continueront d'être peaufinés dans la nouvelle version de l'application pour maximiser l'engagement des utilisateurs et améliorer l'interaction au sein de la communauté étudiante.

Cette démarche proactive a non seulement permis de surmonter les défis techniques mais a également contribué à préparer l'application pour des tests d'utilisabilité futurs, essentiels pour valider et affiner l'interface et les fonctionnalités avant le déploiement final.

Tests

Les tests via une API temporaire ont joué un rôle clé pour valider les nouvelles fonctionnalités avant leur intégration complète. Ces tests ont aidé à identifier et résoudre les problèmes de compatibilité et de performance, garantissant ainsi que les fonctionnalités fonctionnent harmonieusement une fois déployées. La finalisation de l'intégration de la recherche par tag et item est un exemple de la manière dont ces tests ont facilité une navigation plus efficace à travers l'application.

Tests fonctionnels

J'ai principalement utilisé Postman et FastApi pour tester les différentes routes de notre API. Cela a permis de simuler des requêtes client et de vérifier les réponses de l'API, assurant ainsi que toutes les fonctions fonctionnent comme prévu. Postman et FastApi offrent une interface conviviale pour créer, partager et documenter les tests d'API, ce qui facilite la collaboration entre les membres de l'équipe.

Tests utilisateur

Les tests utilisateurs pour la nouvelle version de l'application Café sans-fil sont prévus une fois que l'implémentation sera terminée. Cette étape cruciale permettra de s'assurer que les améliorations et nouvelles fonctionnalités répondent aux attentes et besoins des utilisateurs. En préparation, des scénarios de test exhaustifs seront développés pour évaluer de manière approfondie toutes les interactions clés au sein de l'application. Un échantillon diversifié d'utilisateurs finaux sera recruté pour obtenir un retour représentatif et précis. L'analyse de ces données permettra d'identifier d'éventuelles améliorations et d'itérer sur la conception avant le lancement final, assurant ainsi une expérience utilisateur optimale et fonctionnelle dès le déploiement.

Documentation

Comme mentionné dans les besoins du projet, la documentation joue un rôle crucial dans la compréhension, l'utilisation et la maintenance de l'application. Pour documenter le projet, un wiki³ (référence officielle du projet) tenu à jour est maintenu tout au long du projet et disponible sur le répertoire GitHub du projet. Il contient également, sous forme de fichier README, des instructions détaillées sur le déploiement et la maintenance de l'application sur Render ainsi que les procédures (guides) d'installation et de d'exécution pour le *front-end* et le *back-end*.

La documentation de l'API est générée à partir de *Swagger UI* et *Redoc*. La documentation⁴ fournie par *Swagger UI* permet aux utilisateurs de visualiser et d'interagir avec notre API. Elle offre la possibilité de tester directement les routes de l'API, facilitant ainsi la compréhension de ses fonctionnalités. La documentation⁵ fournie par *Redoc* présente une version plus épurée de la documentation de l'API, offrant une lisibilité améliorée pour une consultation rapide.

³ <https://github.com/ceduni/cafe-sans-fil/wiki>

⁴ <https://cafesansfil-api.onrender.com/docs>

⁵ <https://cafesansfil-api.onrender.com/redoc>

Conclusion

Au cours de ce projet Café sans-fil, j'ai acquis une expérience précieuse en gestion de projet de bout en bout, similaire à ce que l'on peut rencontrer en milieu professionnel. Cette expérience m'a permis de développer des compétences avancées en utilisant des outils tels que Git et GitHub, et d'approfondir ma maîtrise du développement front-end et back-end. En transformant efficacement les besoins des utilisateurs en prototypes fonctionnels, j'ai pu tester et valider ces solutions avec de véritables clients, garantissant ainsi que les améliorations apportées étaient bien alignées avec leurs attentes.

En outre, l'implémentation récente des fonctionnalités de publication d'événements et d'annonces a été conçue avec l'objectif d'apporter une dimension sociale enrichie à l'application. Cette fonctionnalité vise à faciliter et encourager la communication et l'interaction au sein de la communauté étudiante, en permettant aux cafés et aux étudiants de partager des informations pertinentes et d'organiser des événements. Dans la nouvelle version de l'application, ces fonctionnalités continueront d'être développées et affinées pour améliorer encore plus l'engagement et la participation des utilisateurs, renforçant ainsi la dynamique sociale et communautaire au sein de l'université.

Remerciements

Je tiens à exprimer ma gratitude profonde envers tous ceux qui ont contribué au succès du projet Café sans-fil, en particulier Louis-Édouard LAFONTANT pour sa supervision attentive et son soutien constant tout au long du projet. Son accompagnement a été essentiel pour naviguer à travers les différentes phases du développement et assurera une transition solide vers de futures opportunités professionnelles.

Je remercie également Simon, le responsable du café Tore et Fraction, pour son implication directe dans la définition des exigences et le test des prototypes de la version initiale 0.2.1 de l'application. Son expertise et son temps consacré ont été cruciaux pour aligner notre produit avec les besoins réels des utilisateurs.

Un grand merci aussi à tous les testeurs qui ont évalué le prototype de l'ancienne version de l'application. Leurs feedbacks précieux ont été fondamentaux pour identifier les améliorations nécessaires et les nouveaux besoins. À mesure que j'ai progressé dans l'implémentation de la nouvelle version de Café sans-fil, ces testeurs continueront de jouer un rôle vital en évaluant les nouvelles fonctionnalités et en assurant que l'application répond bien aux attentes des utilisateurs. La contribution de chacun a été inestimable, et je suis enthousiaste à l'idée de voir l'impact positif que Café sans-fil aura sur la vie de campus à l'Université de Montréal.

Bibliographie

“Documentation.” *Tailwind CSS*, 17 June 2021, <https://v2.tailwindcss.com/docs>.

Friis, Rikke, and Teo Yu. “Design Thinking: Get Started with Prototyping | IxDF.” *The Interaction Design Foundation*, 20 July 2020,

<https://www.interaction-design.org/literature/article/design-thinking-get-started-with-prototyping>.

“Guide to prototyping in Figma – Figma Learn - Help Center.” *Figma help*,

<https://help.figma.com/hc/en-us/articles/360040314193-Guide-to-prototyping-in-Figma>. Accessed 5 May 2024.

Johnson, Jeff. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Elsevier Science, 2014.

“Quick Start – React.” *React*, <https://react.dev/learn>.

“What is Paper Prototyping? — updated 2024 | IxDF.” *The Interaction Design Foundation*, 30 June 2017,

<https://www.interaction-design.org/literature/topics/paper-prototyping>.

Annexes

Annexe 1: Application version 0.2.0

Page d'un café (début de la page)

 **café sans-fil** preview

Connexion  1

Cafés > Tore et fraction



Tore et fraction

● Fermé · Ouvre à 08:00 lun. ▾

Café 100% bénévoles, vue du coucher de soleil magnifique, on remplit votre tasse pour 0,50\$.

📍 Pavillon André-Aisenstadt, local AA-1221
💳 Carte de débit (min 8,00 \$) 💳 Carte de crédit (min 5,00 \$)



Nouveau produit
Nouveau produit à 02/02/2024

Atelier
Atelier à 03/02/2024

Menu

Page d'un café (suite de la page) - Cette section a été rognée à des fins de présentation

Menu

Grilled Cheese



LE MOZZA
2,00 \$



LE MARBRÉ
2,00 \$



LE BLANC
2,00 \$



LE BRIE
2,00 \$

Boissons chaudes



Café régulier
0,75 \$



Thé
0,75 \$



Espresso
1,50 \$



Espresso Double
2,00 \$



Page d'un café (fin de la page)

Nous contacter

📞 (514)-998-9008

✉ jeremy.dinh@droit.coop

💻 www.droit.coop

🌐 Facebook

📷 Instagram

Projet Café sans-fil

Annexe 2: Prototype de la version 0.2.1

Page d'un café (complète) - Cette présentation a été choisie pour l'implémentation

Tore et fraction

Ouvert • Ferme à 17:00

CAFÉ TORE ET FRACTION

X F @

Menu

Carte de débit (min 8.00 \$) Carte de crédit (min 5.00 \$)

Cafés	thés
Café filtre 1.00\$	Menthe bio 1.00\$
Café mocha (expresso) 2.00\$	Nespresso 2.00\$
Espresso 2.00\$	Café latte 2.50\$
Americano glacé 2.00\$	Latte glacé 2.50\$
Chocolat chaud 1.00\$	

Boissons	Snack
Liqueur/Jus 1.50\$	Chocolat 1.00\$
Gatorade/Pure Leaf 1.75\$	Friandises 1.50\$
Starbucks 2.25\$	Noix 2.00\$
	Ramen 3.00\$
	Samoses 3.25\$
	Pizza 4.00\$
	Salade 6.25\$

Tore et Fraction

Pavillon André-Aisenstadt, local AA-1221

Cafe 100% bénévoles, vue du coucher de soleil magnifique, on remplit votre tasse pour 0.50\$.

Appareils

Micro-ondes Presse panini Machine à café

Staff

Devenir bénévole

Jeremy Gérant Simon Gérant + 54 bénévoles

annonces

Distribution de Chocolat Gratuit 4 jours

Soirée Saint-Valentin au Café Campus! 10 jours

Tableau d'affichage

Saint-Valentin
14 Février 19h30
7077 Ave du Parc Agora du 5e étage

VIENS CÉLÉBRER LA ST-VALENTIN AVEC NOUS
4@7 FESTIF X

Soirée karaoke
10 Mars 19h30
7077 Ave du Parc Agora du 5e étage

Soirée Karaoke

CHALET
2 Mai 17h30
7077 Ave du Parc Agora du 5e étage
50 billets restants
Prix: 62 \$

Le moment tant attendu est enfin arrivé pour vous annoncer l'arrivée du chalet d'informatique ! Cette session, il se déroule au Domaine du Rocher La première semaine de mise en vente des billets est réservée uniquement aux membres de l'AÉDIROUM (ceux qui ont payé leur cotisation). Après une semaine, s'il reste des places, les +1 seront les bienvenus. La vérification va être faite.

Page d'un café (présentation alternative)


**café
sans-fil**

🔍
Tore et Fraction

Bag
User icon

Ouvert • Ferme à 18:00




**CAFÉ TORE
ET FRACTION**

X
F
I



Menu

· Carte de débit (min 8.00 \$) · Carte de crédit (min 5.00 \$)

Sandwich



boissons chaudes



Tore et Fraction

📍 Pavillon André-Aisenstadt, local AA-1221

Horaire ⓘ

Café 100% bénévoles, vue du coucher de soleil magnifique, on remplit votre tasse pour 0,50\$.

Appareils

Micro-ondes | Presse panini | Machine à café

Staff ✉



Snack



boissons froides



Années **Événements** ⓘ

Distribution de Chocolat Gratuit Il y a 4 jours

Tore et Fraction: Pour égayer votre journée, passez au Café Étudiant ce mercredi! Nous distribuons des chocolats gratuits pour tous nos visiteurs. Une petite douceur pour accompagner vos études et vos pauses café

Soirée Saint-Valentin au Café Campus! Il y a 10 jours

Tore et Fraction: Rejoignez-nous pour une soirée romantique pleine de surprises et de douceur. Amenez votre valentin(e) ou venez rencontrer votre âme sœur. Musique douce, ambiance chaleureuse et boissons spéciales vous attendent ce 14 février.

Projet Café sans-fil

Annexe 3: Capture d'écran de l'application (version 0.2.1)

The screenshot displays the main dashboard of the Café sans-fil application. At the top, there's a header with the logo "café sans-fil", a search bar, and a user icon. A banner for "CAFÉ ANTHROPOLOGIE" is prominently displayed against a background of purple and blue leaves. Below the banner, the text "Fermé - Ouvre à 08:00 lun." is shown. The main content area is divided into several sections:

- Café Anthropologie**: Includes social media links (Facebook, Instagram) and a note about a minimum \$3.00 debit card requirement.
- Menu**: A grid of food and drink items.
 - Grilled Cheese**: LE BRIE (2,00 \$), LE GOUDA (2,75 \$), LE BONJOUR (3,00 \$), LE CHEEBAR FORT (2,50 \$).
 - Boissons chaudes**: Espresso (1,50 \$), Latte ou Cappuccino (3,00 \$).
 - Boissons froides**: Coca-Cola (1,25 \$), Arizona (1,00 \$), Limonade maison (1,50 \$), Soda au gingembre (1,75 \$).
 - Collations**: Barre Spécial K (0,25 \$), Barre Val Nature (1,00 \$), Barre de chocolat (1,50 \$), Brooksie (0,50 \$).
- Café Anthropologie**: Details about the establishment (Pavillon Lionel-Groulx, local C-3029) and its welcoming and affordable atmosphere. It lists equipment (Micro-ondes, Presse panini, Machine à café), staff (Marthe Briand, Robert Moulin), and a "Devenir bénévole" button.
- annonces**: An announcement for a wine tasting event on May 16, 2024, from 09:23 to 11:23. It includes a "PROMO" button, a "WEEKEND" button, and a "SPECIAL" button. A "Like" button with a count of 1 is also present.
- Tableau d'affichage**: Three event cards:
 - Dégustation de vins**: Explorez notre sélection exclusive de vins lors de notre prochaine soirée dégustation. Date: 16 mai 2024, Heure: 09:23 - 11:23. Buttons: "Je viens!", "Je supporte!".
 - Nuit de musique live**: Venez vivre une expérience unique avec notre soirée musique live ce vendredi! Date: 26 mai 2024, Heure: 09:23 - 11:23. Buttons: "Je viens!", "Je supporte!".
 - Soirée cinéma**: Rejoignez-nous pour une projection exclusive d'un classique du cinéma sous les étoiles. Date: 31 mai 2024, Heure: 09:23 - 13:23. Buttons: "Je viens!", "Je supporte!".

Menu ouvert

This screenshot shows the detailed menu for the "CAFÉ ANTHROPOLOGIE". The main title "Menu" is at the top. The "Grilled Cheese" section contains three items:

- LE BONJOUR**: Un grilled-cheese au fromage camembert 3,00 \$
- LE MARBRÉ**: Un grilled-cheese au fromage cheddar marbré 2,00 \$
- LE PROVOLONE**: Un grilled-cheese au fromage provolone doux 2,50 \$

The "Boissons chaudes" and "Boissons froides" sections are partially visible at the bottom.

Annexe 4: Diagramme de dépendances des composants de l'application

