

Adresse : 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse

RAPPORT DE STAGE



Effectué chez : UpCoop

Du : 03 février 2025 au : 28 mars 2025

Réalisé par : Yanis Rahmani – Stagiaire en développement back-end

Encadré par :

Mamadou Nafo et Sahar Benkhoud

REMERCIEMENTS

Avant de débiter ce rapport de stage, je veux témoigner ma sincère gratitude envers toutes les personnes qui m'ont accompagné et soutenu pendant les huit semaines passées au sein de UpCoop, du 3 février au 28 mars 2025. Une expérience à la fois enrichissante, exigeante, formatrice, tant sur le plan technique qu'humain.

Je souhaite tout d'abord remercier Monsieur Marc BONINO, mon enseignant référent à l'I.U.T. de Villetaneuse, pour son encadrement, sa disponibilité et ses conseils durant ce stage.

Je remercie ensuite mes deux tuteurs de stage qui m'ont été attribués chez UpCoop, Monsieur Mamadou NAFO et Madame Sahar BENKHOUD, pour leur accompagnement, leur écoute et la confiance réciproque établie dans le cadre de mon stage. Au gré de leurs retours, j'ai pu progresser de façon efficace, bien comprendre les enjeux techniques, m'engager dans des projets concrets en travaillant en autonomie.

Je tiens à exprimer mes remerciements à Madame Virginie LINARD, représentante de l'organisme d'accueil pour son accueil chaleureux.

Mes remerciements s'adressent également aux membres de l'équipe du Squad Bénéficiaire de UpCoop qui m'ont permis d'affiner très rapidement mes compétences, d'intégrer l'équipe et de me sentir pleinement impliqué dans les missions qui m'étaient confiées, étant entendu que la gentillesse de mes collaborateurs et l'esprit d'équipe ont beaucoup facilité la réalisation de mes tâches.

Enfin, je souhaite remercier l'ensemble du corps enseignant du BUT 2 Informatique de l'I.U.T. de Villetaneuse. Leur enseignement m'a apporté les fondations solides nécessaires pour aborder ce stage avec rigueur et curiosité.

SOMMAIRE

Introduction

- Présentation du stage
 - Objectifs du stage (scolaire et professionnels)
 - Aperçu général de l'entreprise d'accueil
-

1. Présentation de l'entreprise (pages 1-4)

- 1.1. Historique, statut et positionnement dans le secteur
 - 1.2. Organisation interne et fonctionnement des équipes
 - 1.3. Rôle de la Squad Bénéficiaire
 - 1.4. Outils, technologies et environnement de travail
-

2. Environnement économique et stratégique (pages 5-7)

- 2.1. Activité de l'entreprise et modèle économique
 - 2.2. Cartes privatives, titres spéciaux et partenaires (MasterCard, Conecs)
 - 2.3. Concurrence et transformation numérique du secteur
 - 2.4. Veille concurrentielle – Février 2025
-

3. Contexte et déroulement du stage (pages 8-11)

- 3.1. Accueil et intégration au sein de l'équipe
 - 3.2. Prise en main des outils et environnement technique
 - 3.3. Modalités de suivi avec les tuteurs
 - 3.4. Méthodologie de travail (tickets, branches Git, agilité...)
-

4. Missions réalisées (pages 11-14)

- 4.1. Présentation générale des tâches confiées
 - 4.2. Script **PurgeAccountsWithoutCard**
 - 4.3. Script **DeleteSuppressedAccounts**
 - 4.4. Script **CleanExpiredCards**
 - 4.5. Problèmes rencontrés et solutions apportées
-

5. Compétences développées (pages 15-17)

- 5.1. Compétences techniques acquises (Laravel, SQL, Git...)
 - 5.2. Compétences méthodologiques (organisation, rigueur, autonomie)
 - 5.3. Liens avec les enseignements du BUT Informatique
-

Conclusion (pages 17-18)

- Bilan de l'expérience
 - Impact professionnels et personnels
 - Perspectives pour la suite de mon parcours
-

Annexes (pages 19-24)

INTRODUCTION

Présentation du stage

Au cours de ma deuxième année de B.U.T. Informatique à l'IUT de Villetaneuse, j'ai eu l'opportunité d'effectuer un stage de huit semaines, du 3 février au 28 mars 2025, au sein de l'entreprise UpCoop, plus précisément dans l'équipe technique appelée Squad Bénéficiaire. Ce stage fait partie intégrante de ma formation : il a pour objectif l'acquisition de compétences techniques à travers leur mise en œuvre sur un problème réel, tout en me confrontant aux réalités du monde du travail.

La Squad Bénéficiaire est l'équipe en charge de la gestion des comptes utilisateurs finaux chez UpCoop : création de compte, rattachement des cartes, envoi de notifications importantes, suppression des comptes, et conformité au RGPD (Règlement Général sur la Protection des Données). Elle gère donc tout ce qui touche à la relation entre l'entreprise et les bénéficiaires de ses services (titres-restaurant, cartes cadeaux, chèques culture, etc.).

Ce stage m'a offert une immersion complète dans un univers professionnel dynamique, centré sur le développement web, avec de réels enjeux techniques, un travail en équipe exigeant et une organisation fondée sur la rigueur et la responsabilisation.

Objectifs du stage (scolaires et professionnels)

Mon objectif principal était de perfectionner mes compétences techniques dans un cadre professionnel concret. En rejoignant une équipe de développeurs expérimentés, j'ai pu mobiliser mes acquis en développement backend tout en explorant des outils et méthodes employés en entreprise (revue de code, tests automatisés, intégration continue, gestion de tickets, etc.).

Sur le plan scolaire, ce stage était un moyen de valider ma capacité à appliquer mes connaissances dans des situations concrètes, mais aussi de développer mes

compétences humaines : autonomie, communication, gestion du temps et adaptabilité à une organisation agile.

Sur le plan personnel, je souhaitais confirmer mon intérêt pour le développement web et me rapprocher de la réalité du métier de développeur. Travailler au contact d'une équipe technique impliquée m'a permis de mieux comprendre les enjeux et les responsabilités de ce domaine.

Aperçu général de l'entreprise d'accueil

UpCoop est une entreprise française qui développe des solutions de titres de paiement (chèques cadeaux, titres-restaurant, cartes mobilité, etc.), ainsi que leur distribution et leur gestion, pour les entreprises et les collectivités, et pour les particuliers, dans l'objectif de faire bénéficier d'un impact social, afin de préserver le pouvoir d'achat des salariés et de faire vivre le commerce de proximité.

Entreprise à mission, UpCoop est organisée autour de plusieurs équipes techniques, les squads, chacune ayant un périmètre fonctionnel dans le cadre du développement et de l'exploitation des solutions. La Squad Bénéficiaire, dans laquelle j'ai été intégré, est dédiée à la gestion des utilisateurs finaux (bénéficiaires), à la sécurité de ces derniers, à l'expérience qu'ils en retirent, à l'adéquation de ce qui est mis en place avec la tenue des données personnelles au regard des exigences de la norme RGPD.

En intégrant cette équipe, j'ai pris part à des missions qui y sont liées, à savoir la gestion automatisée des comptes : suppression des comptes inactifs, notification de l'utilisateur avant leur suppression, nettoyage des données liées aux cartes périmées, pour ne mentionner que quelques-unes des tâches essentielles à l'exécution du service.

1. Présentation de l'entreprise

1.1 – Historique, statut et positionnement dans le secteur

Créée en 1964 sous le nom de Chèque Déjeuner, l'entreprise est devenue par la suite Groupe Up, avant d'adopter son nom actuel : UpCoop. Elle est spécialisée dans la gestion de titres de paiement à impact social : titres-restaurant, cartes cadeaux, chèques culture, ou encore chèques d'accompagnement personnalisés. Ces solutions permettent d'améliorer le quotidien des salariés, de soutenir l'économie locale et de renforcer le pouvoir d'achat.

Ce qui distingue UpCoop des autres acteurs du marché, c'est son statut coopératif. Elle est structurée en SCOP (Société Coopérative et Participative), un modèle dans lequel les salariés peuvent devenir sociétaires et participer aux décisions stratégiques de l'entreprise. Ce fonctionnement repose sur des valeurs fortes : égalité, solidarité et ancrage local, dans la lignée de l'Économie Sociale et Solidaire (ESS).

Présente en France mais aussi à l'international, UpCoop est implantée dans plusieurs pays d'Europe (Allemagne, Belgique, Espagne, Italie...) et en Afrique (Mali, Sénégal, Maroc, Tunisie...). Elle s'adresse aux entreprises, collectivités et particuliers, en leur proposant des solutions numériques et des dispositifs d'aide sociale pour les publics fragiles.

Sur un marché très concurrentiel, face à des groupes comme Edenred ou Sodexo, UpCoop se démarque par son modèle coopératif engagé, sa proximité avec les bénéficiaires et sa volonté d'innover en continu, notamment via la digitalisation de ses services.

1.2 – Organisation interne et fonctionnement des équipes

UpCoop est structurée en plusieurs grandes directions : Direction Générale, Marketing, Technique, Systèmes d'Information, Ressources Humaines, etc.

J'ai été intégré à la Direction Technique, qui fonctionne en squads, c'est-à-dire de petites équipes autonomes et multi-disciplinaires. Une squad regroupe généralement :

- Des **développeurs backend** (chargés de la logique côté serveur),
- Des **développeurs frontend** (interface utilisateur),
- Des **testeurs QA** (assurance qualité),
- Un ou plusieurs **chefs de projet**,
- Et parfois des **UX designers** (conception de l'expérience utilisateur).

Chaque squad est responsable d'un périmètre fonctionnel, d'un domaine spécifique, ce qui favorise la spécialisation et la réactivité.

Le fonctionnement repose sur des **méthodes agiles**, organisées en **sprints** (cycles courts de 2 semaines). Chaque sprint comprend plusieurs étapes :

- **Écriture de tests** (pour éviter les bugs récurrents),
- **Revue de code** (relecture par un autre développeur),
- **Intégration et déploiement continu** (mise à jour automatisée du code),
- Et des réunions régulières comme les :
 - **Daily meetings** (brève réunion chaque matin),
 - **Plannings** (préparation du sprint),
 - **Rétrospectives** (bilan à la fin du sprint).

La communication interne repose sur plusieurs outils :

- **Microsoft Teams** pour les échanges quotidiens,
- **Git** pour la gestion du code source (versionnage),
- **Jira** pour suivre les tâches, les bugs et l'avancement global du projet.

1.3 – Rôle de la Squad Bénéficiaire

La Squad Bénéficiaire, à laquelle j'ai été affecté, gère toute la relation technique avec les utilisateurs finaux, appelés ici bénéficiaires. Ce sont eux qui utilisent les solutions d'UpCoop (titres-restaurant, cartes cadeaux, etc.).

Ses missions principales :

- **Créer les comptes utilisateurs,**
- **Associer les cartes ou titres à ces comptes,**
- **Envoyer des notifications importantes** (alertes, changements de statut),
- **Assurer la conformité RGPD,**
- **Supprimer les comptes inactifs ou non conformes.**

Elle manipule des données sensibles et intervient sur des opérations critiques utilisées quotidiennement par des milliers de bénéficiaires. C'est donc une équipe centrale dans la qualité de service de l'entreprise.

1.4 – Outils, technologies et environnement de travail

Durant ce stage, j'ai évolué dans un environnement technique moderne, principalement basé sur l'écosystème Laravel, un framework PHP orienté backend. Les outils et technologies que j'ai utilisés quotidiennement sont :

- **PHP / Laravel** : pour développer la logique côté serveur,
- **MySQL** : base de données relationnelle utilisée pour stocker les informations (comptes, cartes, statuts...),
- **Git** : outil de versionnement du code pour travailler efficacement en équipe,

- **GitLab** et **SourceTree** : pour gérer les branches, commits, et suivre l'évolution du code.

L'organisation du travail s'appuie sur :

- Des **daily meetings** pour faire le point chaque matin,
- Des **revues de code** systématiques,
- Des **tests automatisés** pour vérifier les fonctionnalités existantes,
- Et une **intégration continue**, permettant des déploiements fluides et fiables.

Enfin, j'ai utilisé :

- **Microsoft Teams** pour les échanges avec l'équipe,
- **Jira** pour suivre les tâches et leur avancement,
- **DBBear** pour interagir visuellement avec les bases de données.

Un poste de travail (ordinateur Dell sous Linux) m'a été fourni, entièrement configuré avec les outils nécessaires.

J'ai également eu la possibilité de travailler à distance, tout en maintenant une communication régulière avec l'équipe.

2. Environnement économique et stratégique

2.1. Activité de l'entreprise et modèle économique

UpCoop est une entreprise française qui conçoit et commercialise des titres de paiement à impact social : titres-restaurant, cartes cadeaux, chèques culture ou encore chèques d'accompagnement personnalisé. Ces solutions visent à améliorer le quotidien des bénéficiaires tout en soutenant le commerce local, à destination des entreprises, des collectivités et des associations.

Son activité repose sur un modèle économique hybride, à la fois performant et engagé. Le chiffre d'affaires provient principalement de la vente de titres (au format papier ou numérique) et des frais de gestion liés à leur distribution et à leur utilisation. L'entreprise propose également des services complémentaires : plateformes de gestion, solutions RH ou dispositifs d'accompagnement social.

En parallèle, UpCoop investit fortement dans le numérique pour faciliter l'accès aux services : applications mobiles, portails web, automatisation des processus, etc. Cette transformation digitale répond aux attentes des bénéficiaires et aux exigences réglementaires, notamment en matière de protection des données (RGPD).

Ce modèle s'inscrit dans une démarche d'Économie Sociale et Solidaire. En tant que SCOP, UpCoop réinvestit ses excédents au service du collectif : amélioration des services, innovation, conditions de travail, ou projets d'intérêt général. L'objectif n'est pas la maximisation du profit, mais la création de valeur partagée et durable.

2.2 – Cartes privatives, titres spéciaux et partenaires (Mastercard, Conecs)

UpCoop émet différents types de supports de paiement, physiques ou numériques, selon les usages visés et les partenariats techniques établis.

Les **cartes privatives** sont des cartes à usage limité, valables uniquement dans un réseau d'enseignes partenaires ou sur des plateformes précises. Elles permettent de flécher un budget vers un usage particulier (restauration, cadeaux, culture...) tout en restant maîtrisées en termes d'utilisation. Ces cartes sont souvent utilisées pour les titres-restaurant, cartes cadeaux ou autres avantages sociaux dématérialisés.

UpCoop propose aussi des titres spéciaux, comme les chèques d'accompagnement personnalisé (CAP), destinés à des publics en situation de précarité. Ces dispositifs sont généralement mis en œuvre en partenariat avec les collectivités locales ou des associations, dans le cadre de politiques d'action sociale.

Sur le plan technique, UpCoop s'appuie sur deux principaux partenaires :

- **Mastercard** : pour la gestion des cartes à puce interbancaires, notamment pour les titres-restaurant. Cela permet une utilisation plus souple et universelle chez les commerçants acceptant ce réseau.
- **Conecs** : une plateforme numérique spécialisée dans la dématérialisation des titres prépayés. Elle assure l'interopérabilité et le traitement des paiements électroniques pour les solutions telles que les cartes UpDéjeuner.

Ce système de partenariats permet à UpCoop d'assurer une large compatibilité tout en garantissant la sécurité, la traçabilité et la conformité des paiements.

2.3 – Concurrence et transformation numérique du secteur

Le secteur des titres de paiement à impact social est dominé par quelques grands acteurs comme **Edenred** (Ticket Restaurant) ou **Sodexo** (Pass Restaurant), qui bénéficient d'une forte notoriété et de réseaux de partenaires étendus. À leurs côtés, des entreprises comme **Swile** ou **Pluxee** ont émergé plus récemment, misant sur des plateformes 100 % digitales et des expériences utilisateurs repensées pour capter une clientèle plus jeune et plus connectée.

Dans ce contexte, UpCoop doit faire face à une pression concurrentielle importante, à la fois sur les fonctionnalités proposées, la simplicité d'usage, et l'innovation continue. Les clients attendent des solutions intuitives, disponibles sur mobile, avec une gestion fluide et des intégrations possibles dans leurs outils RH. Les bénéficiaires, eux, souhaitent un usage rapide, sécurisé et flexible de leurs titres.

Désormais, la transformation numérique est bel et bien devenue un levier stratégique à part entière. Le secteur des titres dématérialisés (cartes, applications mobiles, espace personnel en ligne, etc.) est devenu un standard du marché. Les entreprises doivent donc s'intégrer à des systèmes de gestion interopérables, conformes aux normes de sécurité (RGPD, authentification forte...), tout en adoptant une logique évolutive.

C'est dans cette dynamique qu'UpCoop investit fortement dans le digital : portails dédiés aux employeurs, interfaces simplifiées pour les bénéficiaires, automatisation des processus métiers, suivi en temps réel, etc. L'objectif : répondre aux attentes actuelles d'un acteur innovant, sans renier son modèle coopératif.

2.4 – Veille concurrentielle – Février 2025

Dans un secteur où la concurrence est dominée par des géants comme Edenred ou Sodexo, UpCoop doit constamment s'adapter pour conserver sa position tout en respectant ses valeurs coopératives.

En février 2025, la veille concurrentielle a mis en lumière plusieurs tendances clés :

- Edenred continue de renforcer sa présence sur le digital, avec le développement de sa plateforme mobile "MyEdenred", qui centralise titres-restaurant, offres partenaires et suivi de dépenses. L'entreprise mise également sur l'intelligence artificielle pour personnaliser les services proposés aux utilisateurs.
- Sodexo, de son côté, investit dans l'écosystème "Sodexo Pass", avec un accent sur les partenariats et l'intégration de services annexes comme la gestion RH et les programmes de fidélité. Sa stratégie repose sur la diversification et la fidélisation des entreprises clientes.

Ces concurrents ont une approche plus commerciale, souvent guidée par la performance boursière et la recherche de rentabilité. À l'inverse, UpCoop mise sur un modèle coopératif, ancré dans l'économie sociale et solidaire, avec un investissement régulier dans le numérique, mais aussi dans l'accompagnement de ses clients et bénéficiaires.

La veille a aussi permis d'identifier l'arrivée de nouveaux acteurs digitaux, plus agiles, spécialisés dans les solutions 100 % dématérialisées. Cela pousse UpCoop à accélérer sa propre transformation, tout en conservant son identité fondée sur la gouvernance partagée, la transparence, et l'impact social.

3. Contexte et déroulement du stage

3.1 – Accueil et intégration au sein de l'équipe

Dès mon premier jour chez UpCoop, j'ai été accueilli par mon tuteur, Mamadou, qui m'a remis mon badge d'accès et m'a fait visiter les locaux. L'équipe technique étant située au 5^e étage du bâtiment, j'ai pu découvrir les différents espaces de travail, les salles de réunion, ainsi que les lieux de vie partagés avec les autres pôles. Cette première visite m'a permis de mieux visualiser l'environnement dans lequel j'allais évoluer pendant les huit semaines à venir.

Mamadou a pris le temps de m'expliquer le fonctionnement général de l'entreprise, les interactions entre les différentes directions et le rôle de la Squad Bénéficiaire dans l'écosystème global d'UpCoop. Cette présentation m'a permis de rapidement comprendre les enjeux, le fonctionnement et le positionnement de l'équipe dans laquelle j'étais intégré.

Dès mon arrivée, un bureau attitré m'a été attribué, équipé de deux écrans que je pouvais connecter à l'ordinateur fourni via un port HDMI. Le poste était prêt à l'emploi, ce qui a facilité ma prise de fonction dès les premières heures. L'ambiance dans l'open space était calme et studieuse, mais les échanges amicaux entre collègues ont vite instauré un climat chaleureux et accessible.

Mon intégration a également été marquée par la bienveillance de l'équipe. Chacun a pris le temps de me saluer, de se présenter, et certains m'ont rapidement proposé leur aide si j'en avais besoin. En peu de temps, je me suis senti intégré, à la fois sur le plan humain et professionnel.

Enfin, quelques jours après mon arrivée, j'ai été invité à me présenter officiellement lors d'un daily meeting de l'équipe. Ce moment, simple mais important, m'a permis d'expliquer mon parcours, mon rôle au sein du stage et de nouer des premiers liens concrets avec les autres membres de la squad. Cette dynamique collective a posé les bases d'un travail d'équipe fluide pour la suite du stage.

3.2 – Prise en main des outils et de l'environnement technique

Après mon installation au sein de l'équipe, j'ai commencé à configurer mon environnement de travail avec l'aide de plusieurs collègues. On m'a guidé dans l'installation des outils essentiels : Git pour la gestion du code, Composer pour les dépendances PHP, SourceTree pour la visualisation des branches ainsi que pour

push mon travail, Visual Studio Code pour le code et l'édition, ainsi que DBBear pour la manipulation des bases de données. J'ai également été ajouté aux différentes conversations Teams et j'ai reçu les identifiants nécessaires pour accéder aux services internes d'UpCoop.

Pour m'accorder les droits d'accès à la base de données ainsi que les droits administrateurs nécessaires au bon déroulement de mon stage, Mamadou, mon tuteur de stage, a ouvert un ticket, c'est-à-dire une demande officielle envoyée au groupe technique en charge des accès. Ce processus, courant mais indispensable, m'a permis d'avoir toutes les autorisations requises pour travailler de manière autonome sur les projets de la Squad.

C'est aussi à ce moment-là que j'ai commencé à découvrir Laravel, le framework PHP utilisé au sein de l'entreprise. Je dois l'avouer, au début cela n'a pas été simple. Laravel est très complet, puissant, mais dense, avec sa propre logique, sa structure bien définie (routes, contrôleurs, modèles, vues, middlewares, etc.), et un jargon spécifique à comprendre. J'ai donc dû passer par une phase d'apprentissage intense, en m'appuyant sur des tutoriels, des documentations et les conseils de mes collègues. Progressivement, j'ai assimilé les fondamentaux : l'organisation d'un projet Laravel, le rôle des fichiers `.env`, les migrations de base de données, ou encore la gestion des routes et des contrôleurs.

Cela dit, je suis encore loin de maîtriser ce framework en profondeur : il faudrait sans doute plusieurs mois de formation dédiée pour atteindre le niveau d'aisance des développeurs confirmés de l'équipe. Mais cette prise en main m'a permis de poser des bases solides pour la suite de mon stage, en me familiarisant avec un outil incontournable du développement web moderne.

3.3 – Modalités de suivi avec les tuteurs

Tout au long de mon stage, j'ai été encadré par Mamadou, développeur expérimenté au sein de la Squad Bénéficiaire. En tant que tuteur de stage, il a été mon référent principal : il m'a accompagné dès les premiers jours, m'a aidé à comprendre les enjeux techniques de l'équipe et m'a progressivement confié des tâches à réaliser en autonomie.

Mamadou a toujours été disponible pour répondre à mes questions, débloquer des situations, ou simplement me donner des conseils sur la manière d'aborder une problématique. Lors de nos échanges, il prenait souvent le temps d'expliquer les bonnes pratiques du métier, notamment en matière de rigueur dans le code, d'organisation dans les branches Git, ou encore d'approche méthodologique dans la

résolution de bugs. Ce suivi régulier, mêlant technique et bienveillance, m'a permis d'évoluer dans un cadre à la fois exigeant et rassurant.

J'ai également été suivi à distance par Sahar, ma tutrice, basée en Tunisie. Elle travaille en télétravail avec l'équipe depuis l'étranger, et malgré la distance, elle a joué un rôle important dans mon suivi. En plus de m'accompagner sur les aspects administratifs liés au stage, Sahar est une développeuse experte en Laravel, ce qui a été d'une grande aide au fil des semaines. Nos échanges sur Teams étaient fréquents et enrichissants : elle m'a souvent orienté dans ma compréhension du framework et dans la lecture du code existant.

Ce double accompagnement, à la fois local avec Mamadou et distant avec Sahar, a été pour moi un vrai pilier. Il m'a permis d'avoir un réel support sur les projets, et de bénéficier d'un soutien constant, que ce soit sur le plan technique, humain ou organisationnel.

3.4 – Méthodologie de travail (tickets, branches Git, agilité...)

Lors de mon stage, la gestion des tâches s'est faite via un système de tickets, dans lequel chaque tâche ou fonctionnalité à développer était décrite de manière détaillée. Ces tickets étaient suivis et gérés avec Jira, un outil centralisé qui permet à toute l'équipe de visualiser l'avancement des projets.

Concernant le développement en lui-même, l'équipe utilise Git pour la gestion du code, et chaque développeur travaille sur sa propre branche. Cela permet de développer des fonctionnalités sans affecter le travail des autres. Une fois qu'une fonctionnalité est prête, le développeur crée une Pull Request (PR). Grossièrement, c'est une demande pour intégrer (ou « fusionner ») les modifications de sa branche dans la branche principale du projet. Avant la fusion, les autres membres de l'équipe révisent le code pour s'assurer qu'il respecte les bonnes pratiques et qu'il n'introduit pas de bugs.

Les sprints agiles sont au cœur de notre organisation : chaque sprint dure environ deux semaines, et à la fin de chaque sprint, un ensemble de tâches doit être terminé. Les Daily Meetings nous permettent de faire un point rapide chaque matin sur l'avancement de chacun et de résoudre les problèmes éventuels en équipe. Ces réunions sont un bon moyen de rester sur la même longueur d'onde et d'avancer ensemble de manière fluide.

En résumé, le travail se fait de manière très collaborative, avec une attention constante portée à la qualité du code et à l'intégration continue, grâce aux revues de code et aux outils comme Git et Jira. Cette méthodologie m'a permis de bien m'intégrer à l'équipe et de comprendre les bonnes pratiques à adopter pour un développement efficace et cohérent.

4. Missions réalisées

4.1. Présentation générale des tâches confiées

Tout au long de mon stage, j'ai pu travailler sur plusieurs projets stratégiques intégrés à la Squad Bénéficiaire d'UpCoop. Ceux-ci m'ont permis d'entrer dans le vif du sujet des processus techniques pratiques, et de contribuer à l'amélioration continue des services proposés par l'entreprise. Les principales actions que j'ai réalisées sont liées à la gestion des comptes d'utilisateurs et à la mise en place de scripts permettant d'optimiser les processus de nettoyage et de gestion des données.

4.2. Script `PurgeAccountsWithoutCard`

L'une des missions principales de mon stage a été le développement d'un script intitulé **`PurgeAccountsWithoutCard`**, conçu pour automatiser la suppression des comptes utilisateurs n'ayant plus de carte associée. Ce script, écrit en PHP avec le framework Laravel, interagit directement avec la base de données pour identifier et supprimer les comptes inutilisés ou incomplets.

L'objectif de ce script est d'améliorer la gestion des données en nettoyant la base de données des comptes obsolètes. Il applique des filtres pour trouver les utilisateurs n'ayant pas de carte rattachée depuis une certaine période (7 ou 14 jours), et selon l'ancienneté, il envoie des alertes ou supprime définitivement les comptes. Ce processus garantit la conformité des données et réduit les risques de stockage d'informations inutiles.

Le développement de ce script m'a permis de mieux comprendre l'utilisation de Laravel, notamment les migrations et les modèles Eloquent, qui facilitent l'interaction avec la base de données. En utilisant les migrations, j'ai pu définir la structure

nécessaire à l'exécution du script, et les modèles Eloquent ont simplifié les requêtes pour accéder et manipuler les données des utilisateurs.

Annexe [1] : Voici une capture d'écran du code du script **PurgeAccountsWithoutCard**, qui illustre comment il participe au nettoyage des comptes utilisateurs sans carte associée et montre l'utilisation de Laravel pour interagir avec la base de données.

4.3 – Script DeleteSuppressedAccounts

Un autre travail important réalisé durant mon stage a été la création du script **DeleteSuppressedAccounts**. Ce script permet de gérer la suppression des comptes utilisateurs ayant été placés dans un état "Supprimé" dans la base de données. Son objectif est de restaurer la propreté de la base en supprimant définitivement ces comptes après un délai de latence déterminé, qui permet de conserver les données valides avant leur suppression définitive. Ce processus garantit que la base de données reste propre et optimisée.

La démarche suivie dans ce script est simple : il identifie les comptes utilisateurs ayant le statut "SUPPRESS" et qui sont restés dans cet état pendant un certain temps, généralement 30 jours. Une fois ces utilisateurs détectés, le script les supprime en respectant les règles de confidentialité et de sécurité.

Pour cela, j'ai utilisé Laravel Eloquent pour interagir avec la base de données, ce qui simplifie la gestion des utilisateurs et rend le code plus lisible. Le script effectue une requête filtrant les utilisateurs par leur statut et la date de leur abandon, avant de les supprimer de manière sécurisée.

Le script **DeleteSuppressedAccounts** fait partie d'un processus de gestion des données et d'optimisation de la base de données, et il a été conçu pour être automatisé.

Annexe [2] : Voici une capture d'écran du code du script **DeleteSuppressedAccounts**, illustrant son rôle dans la suppression des comptes utilisateurs supprimés et son intégration avec la base de données à l'aide de Laravel.

4.4 – Script CleanExpiredCards

Un des projets les plus intéressants et pratiques que j'ai réalisés pendant ce stage a été la création du script **CleanExpiredCards**. Ce script est conçu pour nettoyer la base de données en supprimant automatiquement les cartes qui ont expiré. C'est une tâche indispensable pour maintenir la base de données à jour et éviter qu'elle ne devienne encombrée de données obsolètes, ce qui pourrait ralentir les performances du système.

L'idée est simple : le script parcourt les cartes enregistrées dans la base de données, identifie celles dont la date d'expiration est dépassée, puis les supprime en toute sécurité. C'est un bon exemple de l'automatisation de tâches répétitives, ce qui permet de gagner du temps et d'éviter des erreurs humaines.

J'ai utilisé Laravel Eloquent pour interagir avec la base de données, ce qui a rendu la gestion des cartes beaucoup plus fluide et le code plus lisible. Cela m'a permis de créer des requêtes efficaces pour cibler uniquement les cartes expirées, et de m'assurer que seules celles-ci étaient supprimées, tout en préservant la sécurité et l'intégrité des autres données.

Cette tâche m'a non seulement permis d'approfondir ma compréhension de Laravel, mais aussi de contribuer concrètement à l'amélioration continue des processus chez UpCoop. C'est ce genre de projet qui m'a fait comprendre à quel point l'automatisation et la gestion de données sont importantes dans un environnement professionnel.

Annexe [3] : Voici une capture d'écran du code du script **CleanExpiredCards**, illustrant son rôle dans le nettoyage des cartes expirées et son intégration avec la base de données à l'aide de Laravel.

4.5 – Problèmes rencontrés et solutions apportées

Tout au long du développement des scripts, plusieurs défis techniques se sont présentés à moi. Ces problèmes, bien que parfois frustrants, ont été des occasions d'apprendre et de grandir. Et sans l'aide précieuse de mes tuteurs, je n'aurais pas pu surmonter certains obstacles.

Le premier gros problème que j'ai rencontré concernait la gestion des transactions dans les scripts `PurgeAccountsWithoutCard` et `DeleteSuppressedAccounts`. En effet, ces scripts étaient cruciaux car ils supprimaient des comptes utilisateurs, et une erreur aurait pu compromettre l'intégrité des données. Pour résoudre ce problème,

Mamadou m'a orienté vers l'utilisation des transactions DB de Laravel. Ce mécanisme garantit que toutes les suppressions sont effectuées correctement ou, en cas d'échec, qu'elles sont annulées pour maintenir la cohérence des données. Sans son aide, je n'aurais pas pu saisir la subtilité de cette approche et l'implémenter correctement.

Ensuite, j'ai rencontré des lenteurs dans le script `CleanExpiredCards`. La requête pour récupérer les cartes expirées était trop lente, surtout avec une grande quantité de données. Mamadou m'a conseillé d'ajouter des index sur les colonnes fréquemment utilisées. Cela a permis de réduire considérablement le temps d'exécution de la requête. C'est grâce à ses conseils avisés que j'ai pu optimiser cette partie du code et résoudre ce problème de performance.

Lors de la mise en place du système de notifications dans le script `PurgeAccountsWithoutCard`, je me suis heurté à un problème de lenteur dans l'envoi des mails. Sahar, ma tutrice basée à distance, m'a suggéré d'utiliser un système de file d'attente pour envoyer les notifications en arrière-plan. Elle m'a expliqué que cela permettrait de garder les performances du script intactes tout en assurant que les utilisateurs reçoivent bien leur notification. Grâce à ses explications, j'ai pu résoudre ce problème efficacement.

Un autre défi est survenu lorsque j'ai testé le script. Mamadou et Sahar m'ont rapidement fait comprendre l'importance de bien journaliser les erreurs et les succès des suppressions. Ils m'ont conseillé d'ajouter un système de logs détaillé pour pouvoir suivre facilement le bon déroulement des opérations. Cela m'a permis de peaufiner le script et d'avoir un suivi en temps réel des erreurs ou des réussites, ce qui est devenu essentiel lors des tests et de la mise en production.

Enfin, la sécurité des opérations a été un point crucial, surtout lorsqu'il s'agissait de supprimer des utilisateurs. Mes tuteurs m'ont guidé sur la nécessité d'ajouter des vérifications de sécurité avant chaque suppression, pour m'assurer que seules les suppressions légitimes aient lieu. Ils m'ont expliqué à quel point il est important de valider chaque action pour respecter les meilleures pratiques de sécurité. Grâce à eux, j'ai pu renforcer la fiabilité du code et m'assurer qu'il répondait aux exigences de sécurité.

Ces difficultés, bien qu'assez complexes, ont été des occasions d'apprendre et de progresser. Ce n'est qu'avec l'aide constante de mes tuteurs, Mamadou et Sahar, que j'ai pu résoudre ces problèmes. Leur soutien m'a non seulement permis de trouver des solutions techniques, mais m'a aussi donné la confiance nécessaire pour affronter de futurs défis. Sans leur aide, je n'aurais pas pu surmonter certains blocages.

5. Compétences développées

5.1. Compétences techniques acquises (Laravel, SQL, Git, etc.)

Les compétences techniques acquises durant ce stage sont désormais des bases solides pour mon avenir professionnel. J'ai eu la chance de travailler avec des outils essentiels dans le développement web et l'informatique, tels que Laravel, SQL, Git, SourceTree, et DBear. Ces technologies sont incontournables dans le milieu du développement, et bien qu'elles aient exigé de moi beaucoup d'apprentissage, elles me fournissent des fondations solides.

Le travail avec Laravel, notamment pour le développement de scripts et la gestion des bases de données, m'a permis de comprendre comment structurer efficacement un projet PHP. Avec SQL, j'ai renforcé mes compétences en gestion des données, tandis que Git m'a permis de collaborer efficacement avec l'équipe, en apprenant à gérer les branches, faire des commits et résoudre les conflits. En utilisant SourceTree et DBear, j'ai aussi découvert l'importance des outils de gestion visuelle et de l'interaction avec les bases de données.

Cela dit, même si je suis satisfait des progrès réalisés, je sais qu'il me reste encore beaucoup à apprendre pour maîtriser parfaitement ces outils et les intégrer pleinement dans mes projets futurs. Ce stage m'a permis non seulement de renforcer mes compétences techniques, mais aussi de comprendre à quel point ces outils sont cruciaux pour construire des applications solides et performantes.

Aujourd'hui, je me sens prêt à relever les défis techniques qui m'attendent dans ce domaine passionnant, bien que je sache qu'il me reste encore beaucoup à faire pour perfectionner mes compétences. Ce stage a été une expérience clé pour poser les bases de ma carrière dans le développement web et je suis plus que motivé à continuer à progresser.

5.2. Compétences méthodologiques (organisation, rigueur, autonomie)

Au-delà des compétences techniques, ce stage m'a permis de développer des compétences méthodologiques essentielles pour réussir dans le monde professionnel. Tout au long de cette expérience, j'ai appris à travailler de manière plus structurée et organisée, à adopter une rigueur nécessaire pour le développement de projets complexes, et à gagner en autonomie.

Une des compétences les plus précieuses que j'ai développées est la gestion du temps et le respect des délais. En travaillant avec des équipes expérimentées, j'ai appris à mieux planifier, à évaluer le temps nécessaire pour chaque étape, et à m'assurer que je pouvais respecter les objectifs fixés. Cette capacité d'organisation est d'autant plus importante dans un projet où chacun a sa part de travail à gérer et à coordonner avec les autres.

J'ai par ailleurs acquis beaucoup de connaissances sur la rigueur nécessaire dans le développement pour éviter les erreurs de code, bien organiser ses fichiers et ses branches Git, et livrer un produit de qualité. J'ai compris qu'en développement, rien ne doit être laissé au hasard et que la rigueur est essentielle pour produire un bon travail.

L'autonomie a également été un point central de ce stage. Même si l'équipe m'a toujours soutenu, j'ai dû apprendre à gérer mes tâches seul, à résoudre des problèmes seul, et à m'organiser pour atteindre mes objectifs.

Cette autonomie m'a permis de gagner en confiance, et je me sens désormais capable de prendre en charge des projets plus complexes, avec moins de supervision.

En somme, ce stage m'a appris à allier la rigueur technique à une organisation méthodique, tout en développant mon autonomie. Ces compétences méthodologiques sont désormais des atouts précieux pour ma carrière future.

5.3 Liens avec les enseignements du BUT Informatique

Ce stage a été une véritable immersion dans le monde du développement professionnel, et en y repensant, je me rends compte que tout ce que j'ai appris au cours de mon BUT Informatique m'a permis de franchir les premières étapes. Les bases que j'ai acquises en SQL, JavaScript et gestion des bases de données ont constitué un point de départ solide, mais la réalité du travail en entreprise m'a vite montré que l'aspect pratique est bien plus complexe et exigeant que ce que j'avais imaginé.

À l'université, on nous enseigne les concepts et la théorie, et c'est précieux. Mais lorsque tu arrives dans un environnement où les projets sont réels, où chaque ligne de code peut avoir un impact concret, la différence est saisissante. Les outils comme Laravel, SourceTree, et DBBear, que j'avais effleurés à l'IUT, sont devenus des éléments centraux de mon quotidien ici. Et même si je les avais vus en théorie, leur utilisation dans un cadre professionnel m'a pris un temps d'adaptation que je n'avais pas anticipé.

Au début, cela a été un défi. L'écart entre la théorie et la pratique était frappant. J'ai dû me battre pour surmonter les moments de doute, mais au fur et à mesure, j'ai compris à quel point l'expérience terrain est essentielle. Ce stage a été une occasion de combler ces lacunes, de voir l'importance de la rigueur dans chaque tâche, de comprendre qu'au-delà du code, il y a une vraie organisation, une vraie équipe, des enjeux concrets.

Ce que je retiens de cette expérience, c'est qu'à l'université, on m'a donné les outils pour réussir, mais c'est ici, dans le monde réel, que j'ai appris à les utiliser de manière à ce qu'ils aient un véritable impact. Cette expérience m'a ouvert les yeux sur tout ce que je dois encore apprendre, mais elle m'a aussi fait sentir que je suis sur la bonne voie, prêt à relever les défis à venir avec encore plus de passion et de détermination.

Conclusion

Mon stage chez UpCoop a été une expérience incroyablement enrichissante, à la fois sur le plan professionnel et personnel. Dès le début, j'ai pu plonger dans un environnement stimulant, où j'ai été confronté à des défis techniques concrets tout en étant accompagné par une équipe compétente et bienveillante. J'ai découvert à quel point la rigueur, l'autonomie et la capacité à travailler en équipe sont essentielles dans le développement d'applications fiables et efficaces. Grâce à la Squad Bénéficiaire, j'ai appris non seulement à manipuler des outils et des technologies mais aussi à comprendre leur place dans un projet global.

Si l'école m'a apporté des bases solides, j'ai compris que le monde réel du travail est bien plus complexe. Les compétences techniques acquises en cours m'ont permis d'appréhender les outils utilisés dans l'entreprise, mais l'aspect pratique et la gestion des projets sont bien plus intenses. À l'université, nous abordons la théorie de manière plus abstraite, tandis que chez UpCoop, j'ai appris à appliquer ces connaissances dans des situations concrètes, à résoudre des problèmes, à gérer des délais, et à adapter mes méthodes pour répondre aux besoins du projet.

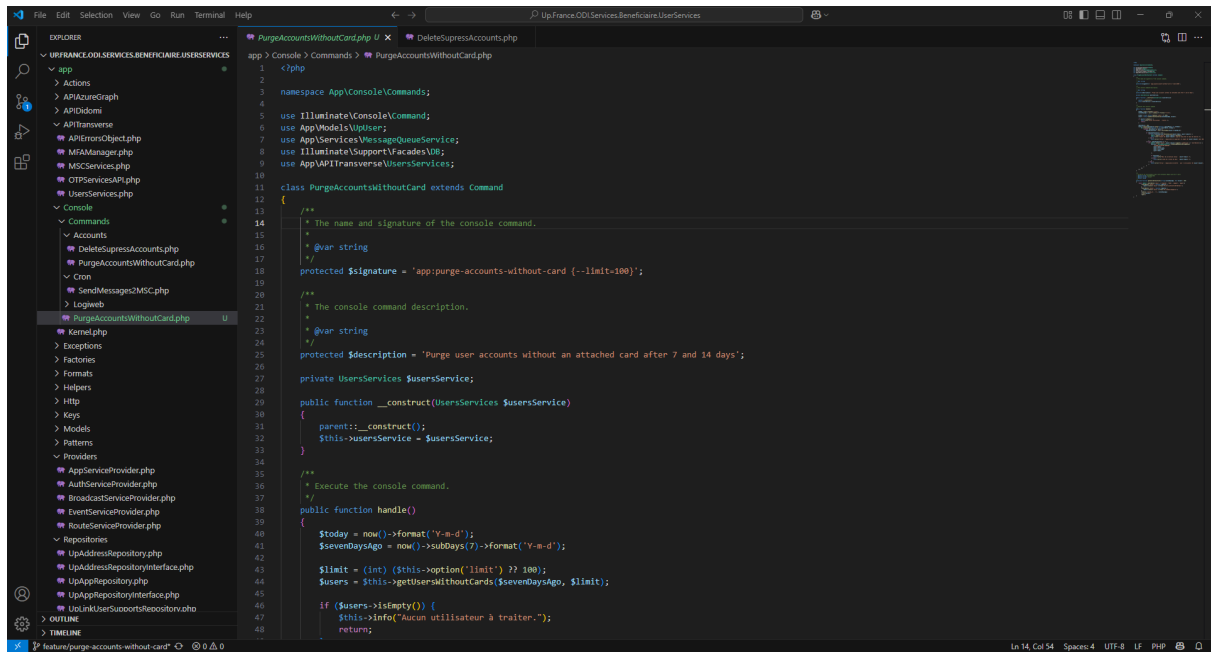
Le plus grand enseignement de ce stage a été de comprendre la nécessité de la rigueur. En informatique, chaque détail compte. Le fait d'être exigeant avec soi-même, de ne jamais laisser de place à l'approximation et d'être capable de revoir constamment son travail font partie des clés de la réussite. C'est aussi une leçon d'humilité : la technique est un terrain mouvant, et chaque défi est une occasion d'apprendre, de progresser et de repousser ses limites.

Sur le plan humain, ce stage a été une véritable aventure. Les échanges avec mes tuteurs, Mamadou et Sahar, ont été déterminants. Leur expertise, leur patience et leur capacité à me guider m'ont permis de surmonter des obstacles techniques, mais aussi de grandir en tant que professionnel. Leur soutien constant m'a montré l'importance d'avoir un mentor, quelqu'un qui croit en tes capacités et qui t'accompagne pour évoluer.

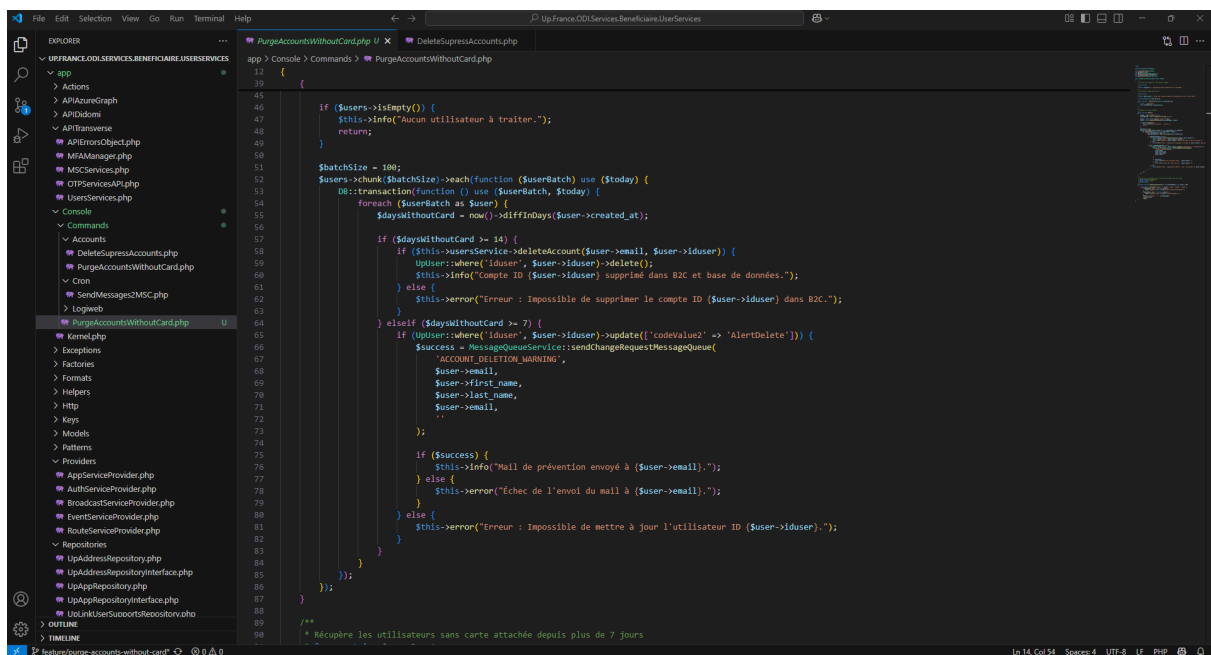
Enfin, ce stage m'a confirmé ma volonté de poursuivre une carrière dans le développement web. Le monde de l'informatique est un univers fascinant, plein de défis, mais aussi de possibilités. J'ai découvert que l'apprentissage est un processus continu, que l'on n'arrête jamais d'apprendre, et que chaque projet nous permet d'affiner nos compétences, d'élargir notre champ de connaissances, et de devenir un meilleur professionnel. Bien que j'aie encore beaucoup de choses à apprendre, je me sens désormais prêt à aborder de nouveaux projets avec confiance et enthousiasme.

Annexes :

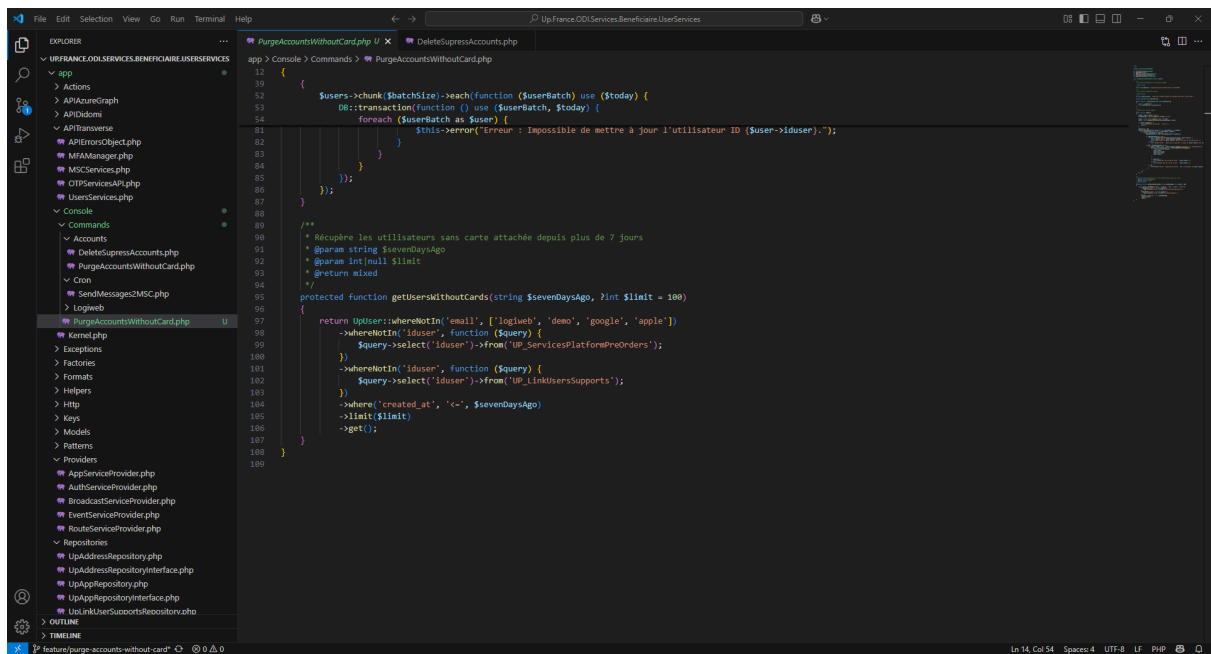
Annexe 1 : Script PurgeAccountsWithoutCard – Gestion des comptes utilisateurs sans carte associée



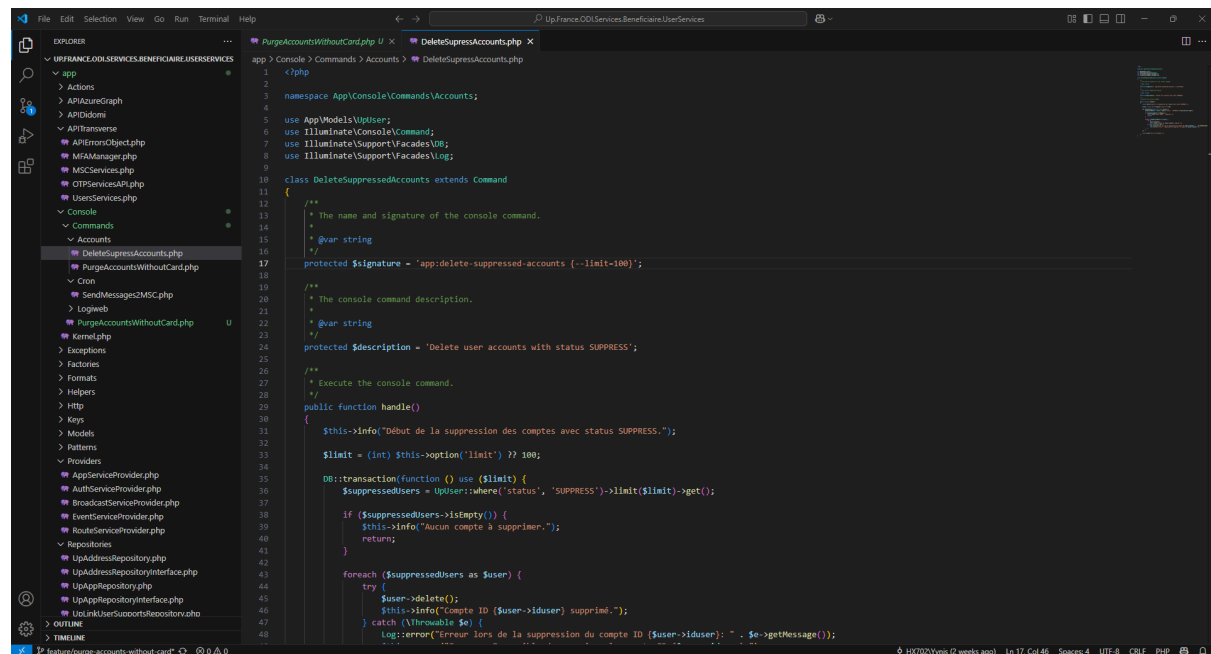
```
1 <?php
2
3 namespace App\Console\Commands;
4
5 use Illuminate\Console\Command;
6 use App\Models\User;
7 use App\Services\MessageQueueService;
8 use Illuminate\Support\Facades\DB;
9 use App\APITransverseUsersServices;
10
11 class PurgeAccountsWithoutCard extends Command
12 {
13     /**
14      * The name and signature of the console command.
15      *
16      * @var string
17      */
18     protected $signature = 'app:purge-accounts-without-card [--limit=100]';
19
20     /**
21      * The console command description.
22      *
23      * @var string
24      */
25     protected $description = 'Purge user accounts without an attached card after 7 and 14 days';
26
27     private UsersServices $userService;
28
29     public function __construct(UsersServices $userService)
30     {
31         parent::__construct();
32         $this->userService = $userService;
33     }
34
35     /**
36      * Execute the console command.
37      *
38      * @return void
39      */
40     public function handle()
41     {
42         $today = now()->format('Y-m-d');
43         $sevenDaysAgo = now()->subDays(7)->format('Y-m-d');
44
45         $limit = (int) ($this->option('limit') ?? 100);
46         $users = $this->getUsersWithoutCards($sevenDaysAgo, $limit);
47
48         if ($users->isEmpty()) {
49             $this->info("Aucun utilisateur à traiter.");
50             return;
51         }
52     }
53 }
```

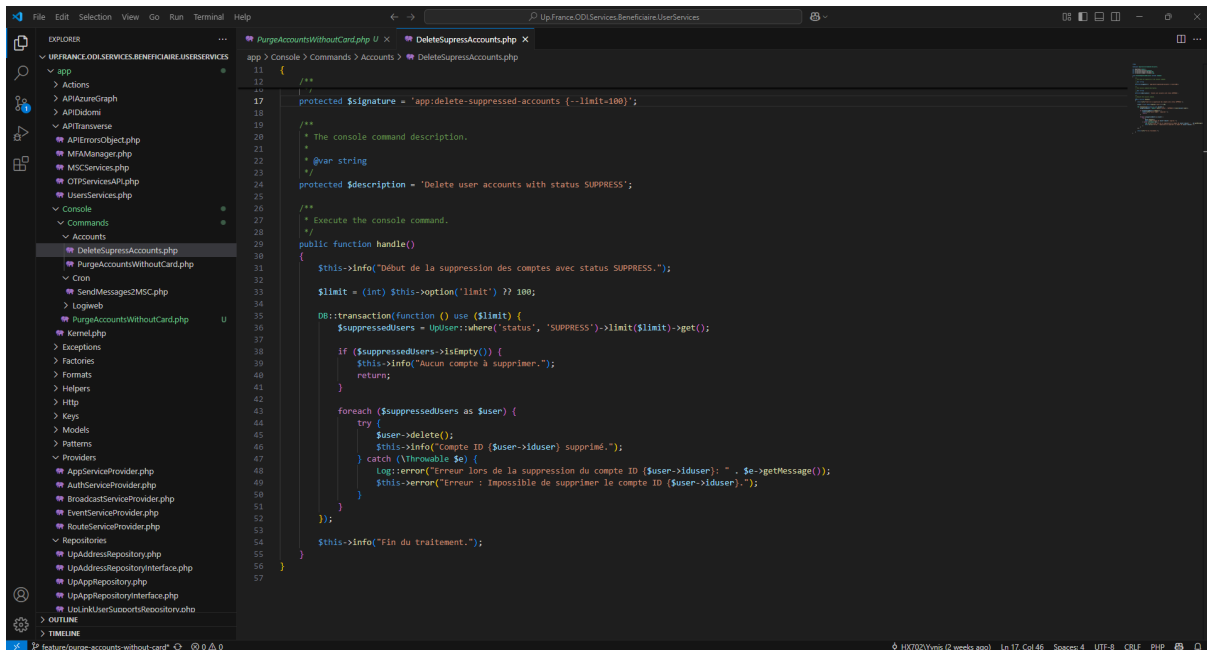


```
54 {
55     if ($users->isEmpty()) {
56         $this->info("Aucun utilisateur à traiter.");
57         return;
58     }
59
60     $batchSize = 100;
61     $users->chunk($batchSize)->each(function ($userBatch) use ($today) {
62         DB::transaction(function () use ($userBatch, $today) {
63             foreach ($userBatch as $user) {
64                 $daysWithoutCard = now()->diffInDays($user->created_at);
65
66                 if ($daysWithoutCard >= 14) {
67                     if ($this->userService->deleteAccount($user->email, $user->idUser)) {
68                         $this->info("compte ID ($user->idUser) supprimé dans B2C et base de données.");
69                     } else {
70                         $this->error("Erreur : Impossible de supprimer le compte ID ($user->idUser) dans B2C.");
71                     }
72                 } elseif ($daysWithoutCard >= 7) {
73                     if ($this->userService->update(["codeValue" => 'AlertDelete'])) {
74                         $success = MessageQueueService::sendChangeRequestMessageQueue(
75                             "ACCOUNT_DELETION_WARNING",
76                             $user->email,
77                             $user->first_name,
78                             $user->last_name,
79                             $user->email,
80                             ...
81                         );
82
83                     if ($success) {
84                         $this->info("Mail de prévention envoyé à ($user->email).");
85                     } else {
86                         $this->error("Échec de l'envoi du mail à ($user->email).");
87                     }
88                 } else {
89                     $this->error("Erreur : Impossible de mettre à jour l'utilisateur ID ($user->idUser).");
90                 }
91             }
92         });
93     });
94 }
```

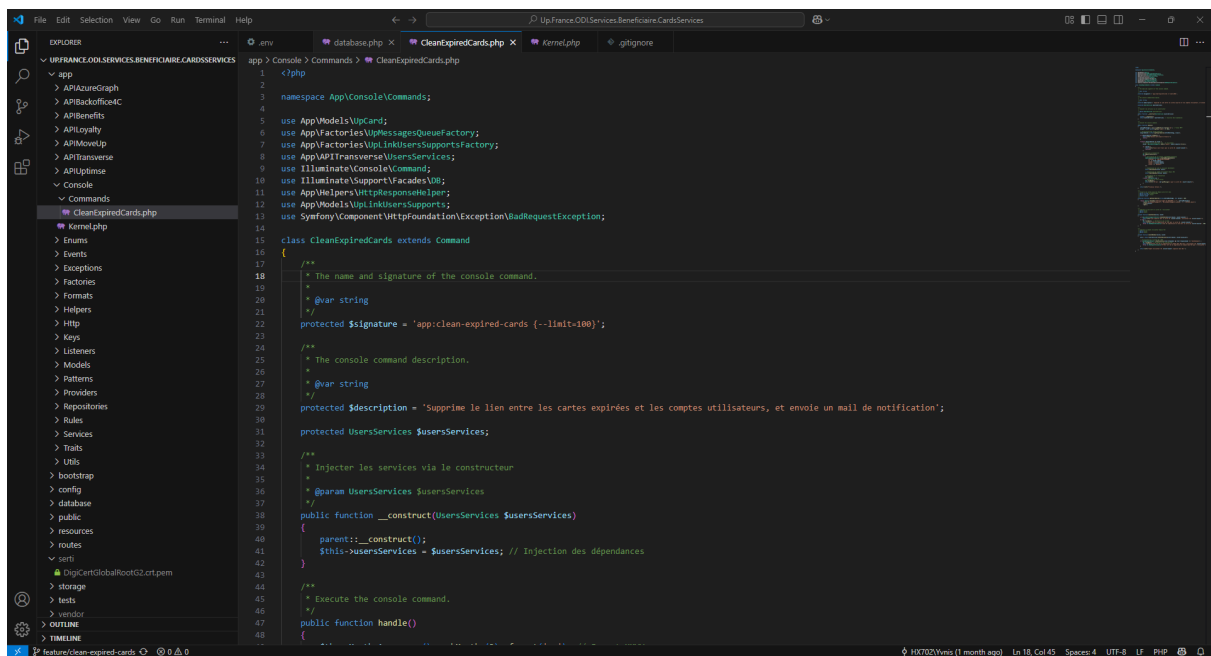


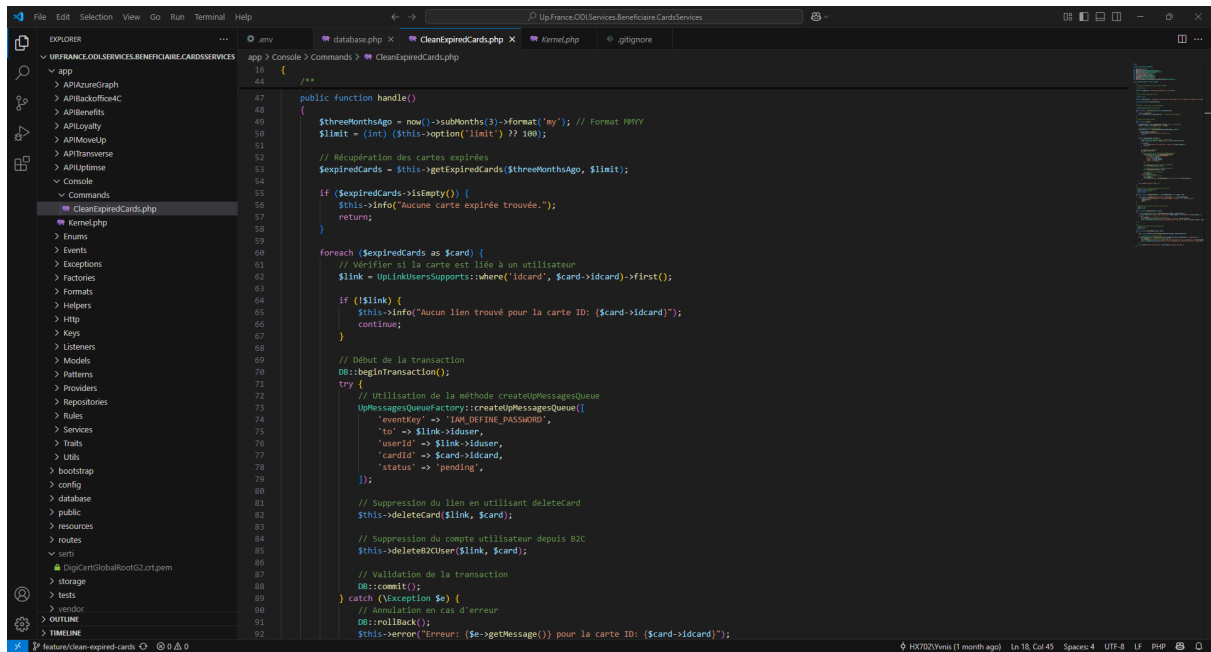
Annexe 2 : Script DeleteSuppressedAccounts – Gestion des comptes utilisateurs avec statut "Supprimé"



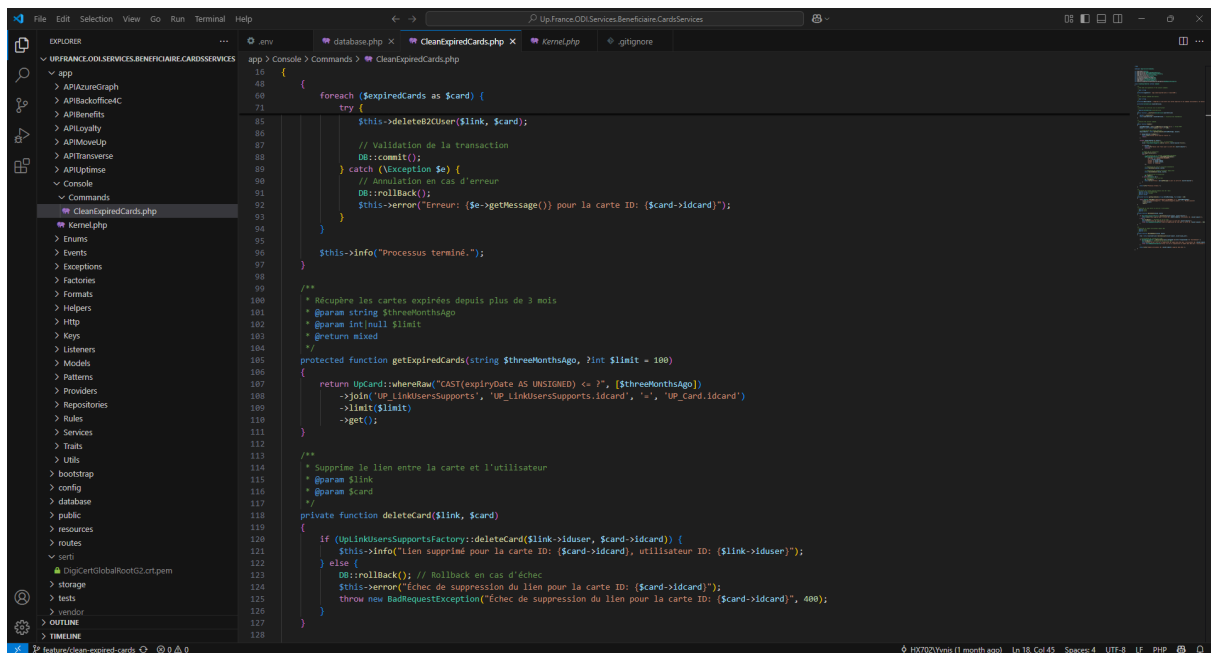


Annexe 3 : Script CleanExpiredCards – Nettoyage des cartes expirées et gestion des données associées

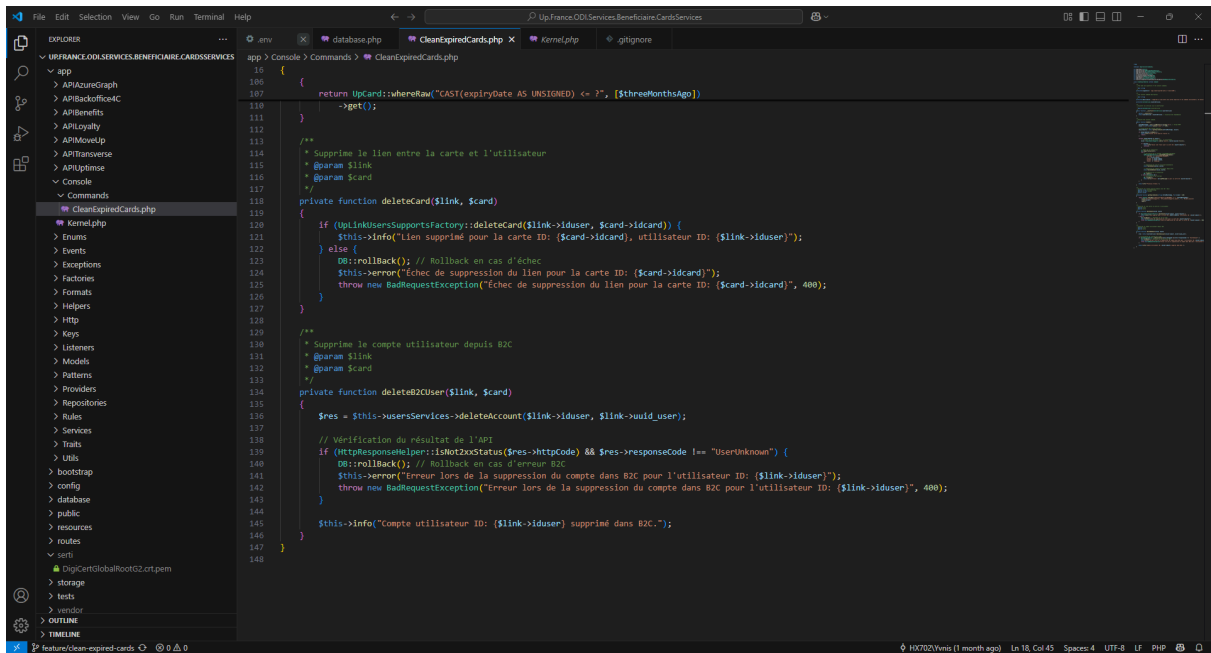




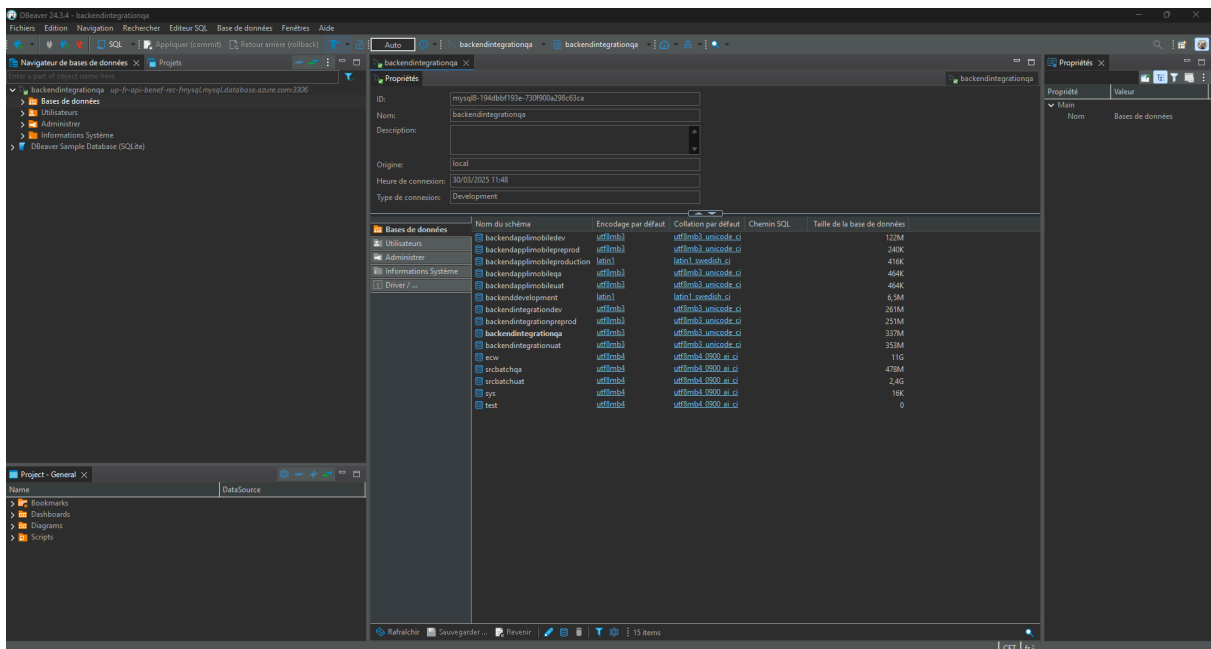
```
16 {
44 /**
47 public function handle()
48 {
49     $threeMonthsAgo = now()-subMonths(3)->format('my'); // Format MMYY
50     $limit = (int) ($this->option('limit')) ?? 100;
51
52     // Récupération des cartes expirées
53     $expiredCards = $this->getExpiredCards($threeMonthsAgo, $limit);
54
55     if ($expiredCards->isEmpty()) {
56         $this->info("Aucune carte expirée trouvée.");
57         return;
58     }
59
60     foreach ($expiredCards as $card) {
61         // Vérifier si la carte est liée à un utilisateur
62         $link = UpLinkUsersSupports::where('idcard', $card->idcard)->first();
63
64         if (!$link) {
65             $this->info("Aucun lien trouvé pour la carte ID: {$card->idcard}");
66             continue;
67         }
68
69         // Début de la transaction
70         DB::beginTransaction();
71         try {
72             // Utilisation de la méthode createMessagesQueue
73             UpMessageQueueFactory::createMessagesQueue([
74                 'eventKey' => 'IAM_DEFINE_PASSWORD',
75                 'to' => $link->idUser,
76                 'userId' => $link->idUser,
77                 'cardId' => $card->idcard,
78                 'status' => 'pending',
79             ]);
80
81             // Suppression du lien en utilisant deleteCard
82             $this->deleteCard($link, $card);
83
84             // Suppression du compte utilisateur depuis B2C
85             $this->deleteB2CUser($link, $card);
86
87             // Validation de la transaction
88             DB::commit();
89         } catch (\Exception $e) {
90             // Annulation en cas d'erreur
91             DB::rollback();
92             $this->error("Erreur: {$e->getMessage()} pour la carte ID: {$card->idcard}");
93         }
94     }
95
96     $this->info("Processus terminé.");
97 }
98
99 /**
100 * Récupère les cartes expirées depuis plus de 3 mois
101 * @param string $threeMonthsAgo
102 * @param int|null $limit
103 * @return mixed
104 */
105 protected function getExpiredCards(string $threeMonthsAgo, ?int $limit = 100)
106 {
107     return UpCard::whereRaw("CAST(expiryDate AS UNSIGNED) < ?", [$threeMonthsAgo])
108         ->join('Up_LinkUsersSupports', 'Up_LinkUsersSupports.idcard', '=', 'Up_Card.idcard')
109         ->limit($limit)
110         ->get();
111 }
112
113 /**
114 * Supprime le lien entre la carte et l'utilisateur
115 * @param $link
116 * @param $card
117 */
118 private function deleteCard($link, $card)
119 {
120     if (UpLinkUsersSupportsFactory::deleteCard($link->idUser, $card->idcard)) {
121         $this->info("Lien supprimé pour la carte ID: {$card->idcard}, utilisateur ID: {$link->idUser}");
122     } else {
123         DB::rollback(); // Rollback en cas d'échec
124         $this->error("Échec de suppression du lien pour la carte ID: {$card->idcard}");
125         throw new BadRequestException("Échec de suppression du lien pour la carte ID: {$card->idcard}", 400);
126     }
127 }
128 }
```



```
16 {
48 {
49     foreach ($expiredCards as $card) {
50         try {
51             $this->deleteB2CUser($link, $card);
52
53             // Validation de la transaction
54             DB::commit();
55         } catch (\Exception $e) {
56             // Annulation en cas d'erreur
57             DB::rollback();
58             $this->error("Erreur: {$e->getMessage()} pour la carte ID: {$card->idcard}");
59         }
60     }
61
62     $this->info("Processus terminé.");
63 }
64
65 /**
66 * Récupère les cartes expirées depuis plus de 3 mois
67 * @param string $threeMonthsAgo
68 * @param int|null $limit
69 * @return mixed
70 */
71 protected function getExpiredCards(string $threeMonthsAgo, ?int $limit = 100)
72 {
73     return UpCard::whereRaw("CAST(expiryDate AS UNSIGNED) < ?", [$threeMonthsAgo])
74         ->join('Up_LinkUsersSupports', 'Up_LinkUsersSupports.idcard', '=', 'Up_Card.idcard')
75         ->limit($limit)
76         ->get();
77 }
78
79 /**
80 * Supprime le lien entre la carte et l'utilisateur
81 * @param $link
82 * @param $card
83 */
84 private function deleteCard($link, $card)
85 {
86     if (UpLinkUsersSupportsFactory::deleteCard($link->idUser, $card->idcard)) {
87         $this->info("Lien supprimé pour la carte ID: {$card->idcard}, utilisateur ID: {$link->idUser}");
88     } else {
89         DB::rollback(); // Rollback en cas d'échec
90         $this->error("Échec de suppression du lien pour la carte ID: {$card->idcard}");
91         throw new BadRequestException("Échec de suppression du lien pour la carte ID: {$card->idcard}", 400);
92     }
93 }
94 }
```



Annexe 4 : Utilisation de DBeaver pour la gestion et l'interaction avec les bases de données



Annexe 5 : Utilisation de SourceTree pour la gestion des branches Git et des commits

