RAPPORT TECHNIQUE DE LA RÉALISATION DU SERVICE “FAIR BEANS”

Réalisé par
Julien Bellavance (111 104 607)
Léandre Gagnon-Lewis (111100481)
Joice Kariya (111023195)

Présenté à
M. Richard Khoury

Dans le cadre du cours
GLO-2005: Modèles et langages des bases de données pour ingénieurs

17 avril 2019

Table des matières

Énonciation du problème et des exigences

Bien qu'il existe de nombreux magasins de cafés ayant déjà une réputation et une portée colossale, le marché du café équitable demeure peu exploité par les boutiques en lignes. Il est généralement vendu par de petites boutiques spécialisées, en petites quantités fixes. La boutique Fair Beans viendrait ainsi occuper une niche intéressante en étant la première boutique de café équitable en ligne permettant d'acheter son café en vrac. Cela permettrait aux clients de choisir exactement la quantité dont ils ont besoin et aux connaisseurs de créer des mélanges personnalisés à leur goût.

Fair Beans étant une boutique spécialisée, le système devra permettre d'acheter différentes variétés de cafés en précisant la quantité voulue. De plus, le site devrait offrir une fiche d'information détaillée de chaque variété (contenant entre autres le nom, le prix au gramme, la compagnie, la provenance, la torréfaction et l'espèce). Les clients devront avoir la possibilité de se créer un compte dans le magasin en ligne. Le magasin devra offrir un système de panier pour les achats, ainsi qu'un système de navigation dans le catalogue pour trouver un café spécifique en fonction de critères de recherche et de mots-clés. Il devrait également avoir une page d'accueil affichant les nouveautés, les rabais et les offres spéciales à durée limitée.

Ce rapport détaille les fonctionnalités du système Fair Beans, tant sur un plan technique que conceptuel. Les premières sections expliquent les modèles entité-relation et relationnel ayant servi à élaborer le système. Les sections suivantes vont en détail dans les fonctionnalités de chaque niveau du système ainsi que dans les aspects d'optimisation et de sécurité. Enfin, le rapport se termine par une courte perspective sur les fonctionnalités qui n'ont pu être implémentées ou qui sortaient du cadre de ce projet.

Outils d'organisation et division des tâches

Afin de s'assurer du bon déroulement du travail, les tâches doivent être équitablement réparties entre les membres de l'équipe de développement. L'outil GitLab est donc utilisé pour permettre une division efficace du travail et permettre aux membres de l'équipe de suivre la

progression du projet en tout temps. Chaque étape et fonctionnalité à implémenter sont définies par des « issues », qui sont assignées au fur et à mesure de l'avancement du travail.

Cela est complémenté par des rencontres hebdomadaires permettant à tous de se mettre à niveau et de discuter de la suite du projet. Il est prévu que chaque membre de l'équipe travaille sur chaque niveau du projet.

Modélisation des données de la BD

Cette section présente une vue conceptuelle du système Fair Beans. Deux types de modèles sont utilisés : le modèle entité-relation, et le modèle relationnel.

Modèle entité-relation

Ce modèle représente le fonctionnement abstrait du système Fair Beans. Les entités les plus importantes sont "Produit" et "Client". Toutes les autres entités sont donc organisées autour de ces deux dernières. Comme Fair Beans est une boutique spécialisée, les produits doivent être détaillés au maximum. "Produit" a donc un grand nombre d'attributs, dont notamment un numéro d'identification qui servira de clé primaire afin d'identifier les variétés de cafés. L'entité "Client" n'a pour attribut qu'une adresse courriel, servant de clé primaire, car il n'est pas souhaitable que plusieurs clients utilisent la même adresse, et le mot de passe. Le courriel sert donc d'identifiant et le client n'a pas besoin de donner son nom ou son adresse à ce stade.

Le reste du modèle montre comment le client pourra interagir avec les produits. Il doit d'abord pouvoir remplir un panier avec les produits de son choix. Le panier indique également le prix total avant taxe des articles qu'il contient. Le client peut également passer une commande. À ce stade, il doit donner une adresse de livraison et le nom associé celle-ci. Chaque commande est identifiée uniquement par son numéro d'identification, afin de garder un historique de transaction, et possède également une date de transaction.

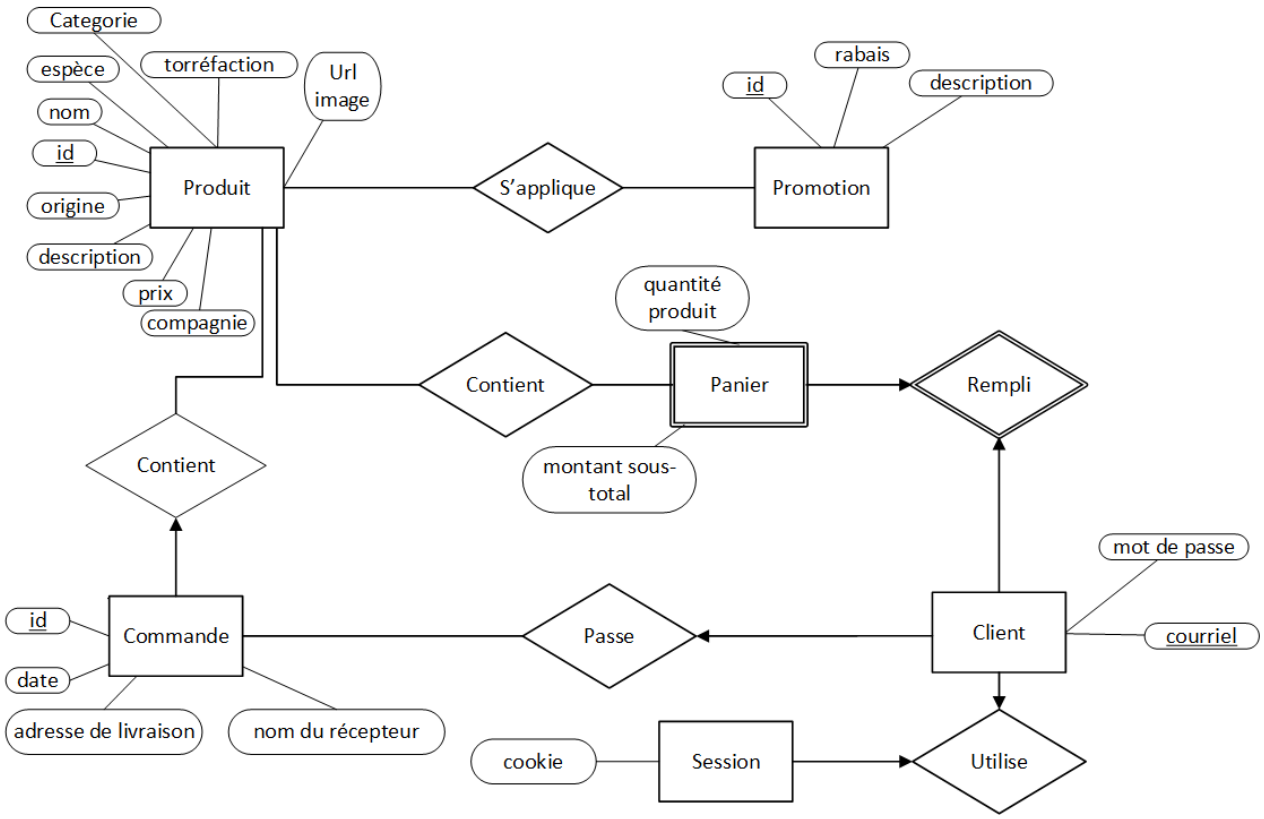

Figure 1 Le modèle entité-relation du système Fair Beans

Des promotions peuvent éventuellement s'appliquer à un ou plusieurs produits. Ces promotions sont identifiées dans la BD par un numéro unique afin d'en garder la trace. Un rabais en pourcentage et une description complètent la description de cette entité. Enfin, les utilisateurs se connectent au site internet par leur session. Afin de leur permettre de se reconnecter automatiquement, un cookie est enregistré dans la BD et sur le navigateur de l'utilisateur.

Le modèle entité-relation est donc très utile pour visualiser le fonctionnement du système. En complément, le modèle relationnel qui suit permettra de détailler les tables de la BD.

Modèle relationnel

Le modèle relationnel du système est représenté par une série de tableaux. Il permet de visualiser comment l'information est stockée dans la BD. Dans la table produit, on retrouve donc simplement les informations nécessaires pour une fiche de produit. Les images ne peuvent pas être enregistrées dans la BD sans utiliser des méthodes qui dépassent l'envergure de ce projet. Elles sont donc conservées par leur URL.

Le panier a pour clé primaire une super-clé formée de l'adresse courriel et du numéro de produit, pour s'assurer de l'unicité de cette combinaison. En effet, il ne peut pas y avoir deux fois le même produit dans le panier, seule la quantité du produit peut être modifiée. Lorsqu'une commande est passée, toutes les informations pertinentes sont enregistrées dans la table commande. L'id_commande fait le lien entre cette table et produit_commandé. Dans la table produit_commandé, la clé primaire sera la combinaison de id_commande et id_produit, afin de préserver l'unicité de cette combinaison.

Le modèle relationnel de la base de données Fair Beans :

produit (product)									
id	nom	categorie	origine	espèce	torréfaction	fournisseur	prix ($/kg)	description	image URL

promotion (promotion)				
id	réduction	attribut	selecteur	description

session (session)	
token	email

utilisateur (user)	
email	mot de passe

panier (cart)		
email	id_produit	quantité

produit_commandé (product_in_command)		
id_commande	id_produit	quantité

commandes (command)								
id	date	courriel	prénom	nom	adresse de rue	ville	code postal	pays

Les promotions sont enregistrées lors de leur création et font effet immédiatement, jusqu'à ce qu'elles soient retirées. Un sélecteur permet d'identifier quels produits sont touchés par la promotion, et est typiquement un attribut des produits.

Les utilisateurs sont identifiés par leur adresse courriel, et leur mot de passe est enregistré dans la BD après avoir été haché par le serveur, afin de le protéger. De plus, les sessions de connexions sont identifiées par le cookie généré pour l'utilisateur. Le cookie sera donc la clé primaire de cette table, mais une adresse courriel ne devrait y apparaître qu'une seule fois.

Notons enfin que certaines tables et certains attributs ne sont pas implémentés, bien qu'ils auraient eu leur place dans ce système. D'abord, il n'y a pas de table pour les employés. Étant donné que Fair Beans commencerait ses opérations en tant que start-up de petite envergure, il a été convenu qu'il n'y aurait pas d'autres employés que les créateurs du système. Il n'est donc pas difficile pour eux de manipuler directement la BD, et une interface pour employés n'a donc pas été implémentée. Par conséquent, il n'est pas nécessaire d'enregistrer les employés dans la BD.

Cela conclut la section conceptuelle du rapport. Les modèles montrés devraient ainsi fournir une compréhension générale du système. Les sections suivantes iront plus en détail dans l'implémentation et les aspects pratiques du projet.

Fonctionnalités du niveau serveur de Base de Données

Cette section traite de l'implémentation technique de la base de données et de son fonctionnement. Les routines et les triggers utilisés pour implémenter les fonctionnalités nécessaires au système seront expliqués.

La base de données implémente 7 tables, présentées dans le modèle relationnel. Les clés primaires ont déjà été expliquées avec le modèle relationnel, mais certaines tables utilisent également des clés étrangères. La table session utilise les courriels de la table utilisateurs puisqu'une session ne doit nécessairement exister que pour un utilisateur existant. De la même façon, la table commande a un attribut courriel référençant les courriels de la table utilisateur, puisque chaque commande doit être affiliée à un client. La table panier utilise les courriels d'utilisateurs et les id de produits afin de lier chaque panier à un client et chaque produit à sa description. Finalement la table produit_commandé fait référence à l'id de la commande associée, et à l'id de chaque produit dans

la commande. Les produits sont enregistrés indépendamment de la commande afin de pouvoir associer plusieurs produits à un même id.

Une routine, sous forme de “trigger”, est également implémentée dans la BD. Celle-ci permet d’ajouter automatiquement les produits du panier d’un client à la table produit_commandé lorsqu’une commande est passée. Les commandes sont donc générées automatiquement. La même fonction permet également de vider le panier du client une fois la commande passée.

Des mesures ont également été prises afin d’optimiser les requêtes de la BD. Celles-ci sont discutées en détail dans la section suivante.

Optimisation de la Base de Données

Afin d’offrir le service le plus rapide et efficace possible, la base de données de Fair Beans doit être optimisée. Pour ce faire, les tables ont été normalisées et des index ont été créés.

D’abord, la normalisation des tables permet d’organiser de manière logique la BD et simplifie les opérations entre les tables. Comme chaque attribut des tables ne contient que des valeurs uniques, la première forme normale est respectée. Notez que certaines chaînes de caractères dans la table produits sont des listes, ce qui est normalement exclu en 1FN. Cependant, dans ce cas ces listes sont considérées comme un seul élément, servant à décrire un produit. Ensuite, les tables sont toutes sous forme FNBC, car toutes les dépendances sont en fonction d’une clé primaire, donc uniques dans la table.

En plus de la normalisation, un index est implémenté pour la table « utilisateur » pour accélérer les opérations de recherche. Dans « utilisateur », la requête la plus courante est une sélection par email. Lorsqu’un utilisateur se connecte à la base de données, il doit entrer son adresse courriel et son mot de passe. Comme le courriel est unique pour chaque utilisateur, la sélection accède directement à la valeur de hachage donnée, sans avoir à parcourir toute la liste des valeurs de hachage possibles. Des tests réalisés avec cinquante mille entrées ont montré que sans index, la recherche prend environ 0,02s. Avec l’index, la recherche prend plutôt 0,00s. Comme Fair Beans vise à couvrir toute la clientèle canadienne, et éventuellement l’international, l’indexation est cruciale pour accélérer les opérations de manipulation de données. Pour que chacun ait accès à son

historique de commande plus efficacement, deux index de hachage ont été ajoutés pour la recherche d'id dans la table produit_commandé et pour la recherche d'email dans la table commande. Un exemple courant qui justifie ces index est quand un utilisateur va voir les détails d'une des commandes qu'il a passées. La première requête appelée serait:

"SELECT * FROM command WHERE email = <email>",

puis l'utilisateur choisirait une commande et une deuxième requête serait lancée:

"select id, name, category, origin, species, roast, roaster, price, flavours, imageUrl, quantity from product_in_command inner join product on product_in_command.product_id = product.id WHERE command_id = <id>'"

Dans la table session, les tokens de cookie sont couramment l'objet de recherche. Cependant, cette table effectue également une insertion à chaque fois qu'un utilisateur ouvre une session et une suppression lors de sa déconnexion. Comme la table subira plus d'opérations de modification que de recherche, il est plus important de réduire le temps des modifications que des recherches. Pour ces raisons, un index sur token n'a pas été implémenté.

Des index en arbre B+ pourraient aussi être ajoutés dans la table produits, car les recherches par produits filtrés sont très courantes. Cependant, la base des données de Fair Beans, par sa nature de boutique spécialisée, ne contiendra jamais plus que quelques centaines de produits. Pour cette raison, la création d'un index ne permettrait pas d'accélérer la recherche.

Cela conclut l'examen du niveau base de données. La section suivante se penchera ensuite sur le prochain niveau du système, le serveur Python.

Fonctionnalités du niveau serveur Python

Le serveur python a pour but principal de servir de pont entre l'interface utilisateur et la base de données. Dans ce but, il utilise le framework flask pour offrir un REST api auquel le front-end peut se connecter. Il utilise Mysql-Connector Driver pour python afin de se connecter à la base de données. En plus de servir de pont, le serveur python peut garder en caches certaines données sur les produits pour éviter des appels à la BD. Par exemple, quand l'utilisateur veut appliquer des

filtres sur les produits, il peut simplement filtrer ses produits en cache plutôt que de refaire une requête. Finalement, le serveur a aussi la responsabilité de hacher les mots de passe et de vérifier si un mot de passe entré correspond bel et bien à celui dans la BD.

Fonctionnalités du niveau interface utilisateur

L'interface utilisateur sert de plateforme d'achat. Ses fonctionnalités peuvent être décomposées en quatre catégories: accueil, magasin, compte et achat. La page d'accueil offre une barre de navigation avec des liens vers les différents composants du site. On retrouve aussi un carrousel affichant les promotions en cours et offrant des liens vers les produits touchés. Le magasin quant à lui, affiche les produits et permet à l'utilisateur d'appliquer différents filtres de recherche. Une seconde barre de navigation permet de faire des recherches sur les produits, et de trier les produits par prix ou par ordre alphabétique. Finalement, le magasin permet à l'utilisateur de choisir un nombre maximum de produits par page entre 16, 32 et 64. Les fonctionnalités du compte permettent à l'utilisateur de se créer un compte client, de se connecter à son compte, de se déconnecter et de consulter l'historique de ses commandes. Les fonctionnalités d'achat offrent une vue au client sur les différents produits dans son panier. Elles lui permettent ainsi de modifier les quantités des produits dans son panier ou de les supprimer. Aussi, elles permettent de compléter la commande avec la saisie des informations nécessaires.

Sécurité du système

Cette section porte sur les mesures de sécurité mise en place dans le système. Afin de protéger la confidentialité des utilisateurs et l'intégrité du système, il est important de considérer comment prévenir les actions malveillantes à son égard. Notamment, il est important de prévenir contre les injections sql et les accès non autorisés au système de BD et aux comptes des clients.

Il est important de noter que cette version du programme n'offre pas de connexion sécurisée entre le serveur et le client, et que cela pose une faille de sécurité majeure, puisque les mots de passe peuvent être interceptés et que des injections sql pourraient être faites en communiquant directement avec le serveur. Il était hors de notre portée pour ce projet cependant d'acheter un

certificat SSL valide. Dans le cas où ce projet irait en production, il est certain qu'une connexion HTTPS serait de mise.

Les comptes clients sont quant à eux protégés par le hachage des mots de passe par le serveur python lors de la création du compte. Ainsi, même si un intrus parvenait à accéder à la table des utilisateurs, il n'obtiendrait pas les mots de passe des clients. Les mots de passe sont également vérifiés par le client avant la création du compte afin qu'ils respectent certaines mesures de sécurité de base, soient une longueur minimale de 6 caractères et la présence d'un caractère autre qu'une lettre.

Le système est également protégé des injections sql directement dans le programme client. Une fonction est appelée sur tous les champs où un utilisateur peut saisir du texte et vérifie si ce texte contient des mots-clés qui se retrouvent dans les injections sql courantes et, le cas échéant, affiche un message d'erreur et bloque la requête.

Finalement, les accès à des pages réservées aux membres sont bloqués sur le client tant que les utilisateurs ne sont pas authentifiés avec le serveur.

Cette section conclut la partie pratique du rapport. Bien que plusieurs concessions ont étés faites lors de l'implémentation du système, soit par contraintes de temps ou de complexité, Fair Beans demeure un système complet et fonctionnel dans ses tâches principales. Les fonctions de sécurité en place devraient donc, en pratique, empêcher toute intrusion dans le système.

Perspective et projets futurs

Comme il a déjà été dit, Fair Beans a été conçu pour être fonctionnel. Le système peut donc effectuer la plupart des tâches qui lui sont confiées tel que l'on s'attend d'un produit commercial. Cependant, il reste de nombreuses fonctionnalités qui n'ont pas pu être implémentées. Les plus importantes sont donc expliquées ici.

Le système de paiement n'est pas fonctionnel, ce qui représente la plus grande fonctionnalité qui ne peut absolument pas manquer au système pour une version commerciale. Lier l'interface utilisateur et le serveur à un ou plusieurs api de systèmes de paiement en ligne devrait donc être la priorité pour des travaux futurs sur le projet.

Dans la section sur le modèle relationnel, il a été mentionné qu'il n'y avait aucune information sur les employés dans la BD. Une table employée dans la BD, contenant l'identifiant et le mot de passe de chaque employé, ainsi qu'une page réservée aux employés sur le site, serait des additions nécessaires. Cela permettrait à des employés n'ayant pas l'autorisation d'accéder directement à la BD d'y faire des modifications quand même, par l'entremise d'une interface utilisateur.

Les informations sur les clients ne sont pas enregistrées, ce qui force les clients à entrer leur adresse et leurs options de paiement à chaque commande qu'ils effectuent. Leur permettre d'enregistrer ces informations améliorerait beaucoup leur expérience avec le système.

Enfin, de nombreuses mesures de sécurité pourraient être ajoutées, par exemple l'authentification à deux facteurs ou le cryptage des adresses courriel. Plusieurs autres fonctionnalités pourraient également être ajoutées, mais les plus importantes ont été indiquées dans cette section, dans le but d'expliquer ce qui a été considéré, mais ultimement coupé du projet.

Conclusion

Ceci conclut le rapport sur la création du système Fair Beans. En tant que boutique spécialisée en vente de café équitable, Fair Beans imposait plusieurs critères sur le design du système final. Il a d'abord été conceptualisé que le magasin offrirait un nombre limité de produits, avec des fiches de description détaillées pour chacun, et que les clients seraient identifiés par leur adresse courriel. Le reste de la conceptualisation a découlé de ces choix. L'implémentation des trois niveaux du système a permis de créer un produit réel et fonctionnel répondant aux exigences déterminées au début du projet.

Le système n'est pas complètement terminé, puisqu'il manque entre autres un système de paiement fonctionnel. Le projet tel qu'il a été monté est cependant une base solide et correctement programmée, qui pourrait sans problème être utilisée pour créer un véritable produit commercial. De plus, la niche que Fair Beans tient à occuper, soit le marché du café équitable en vrac, demeure encore inexploitée. Le système Fair Beans est donc dans une excellente position pour être déployé rapidement et prendre avantage d'un marché intéressant.