Graphical user interface, application

Description automatically generated

UNIVERSITÉ LAVAL

FACULTÉ DES SCIENCES ET GÉNIE

Modèles et langages des bases de données pour ingénieurs

GLO-2005

**Plateforme d’achat en ligne pour des bières sans alcool**

Travail d'équipe remis à M. Richard Khoury

Par

**ÉQUIPE #34**

Jerena Mihary

536952010

Simard Liam

111236843

Christopher Bélanger

111063575

Hiver 2022

Table of Contents

[Énonciation du problème et des exigences 3](#_Toc101183971)

[Spécifications du système et des responsabilités des trois niveaux 4](#_Toc101183972)

[Modélisation des données et de la BD 5](#_Toc101183973)

[Modèle entité-relation 5](#_Toc101183974)

[Modèle relationnel 6](#_Toc101183975)

[Gestion des données 8](#_Toc101183976)

[Logique de l’application 8](#_Toc101183977)

[Interface de l’utilisateur 9](#_Toc101183978)

[Gestion de l’équipe et division du travail. 10](#_Toc101183979)

# Énonciation du problème et des exigences

L’application web que nous avons décidé de concevoir est celle d’une plateforme d’achat en ligne de bière sans alcool. Par bière sans alcool, il est sous-entendu qu’il s’agit d’un pourcentage d’alcool inférieur à . À l’image d’un magasin en ligne traditionnel le client pourra naviguer parmi différentes bières et plusieurs fournisseurs. Un utilisateur qui souhaite passer une commande via l’application aura besoin de se créer un compte afin d’enregistrer ses commandes. Une personne, une fois connectée à son compte, pourra se créer un panier de commande en ajoutant les bières sans alcool qu’il ou elle désire commander et pourra venir les chercher en centre de collecte des commandes. Les clients auront également la possibilité de noter les bières sans alcool.

Le contexte de notre application est décrit par la liste suivante

* Chacune des bières recensées sur notre site internet est accompagné de son nom, sa description, son prix, son contenant donc sa quantité (canette, bouteille, fut), sa marque, sa provenance, son type de malte, sa catégorie (IPA, blonde, brune, rousse, etc.), son arôme (s’il en a un), sa date de création et d’une note qui seraient la moyenne des notes attribués par les clients.
* Pour acheter les bières, les clients ont la possibilité de se créer un compte de fidélité pour accumuler des points. Ces comptes clients comporteront le nom, le prénom, la date de naissance (uniquement les personnes âgées de plus de 18 ans peuvent créer un compte, la boutique en ligne est enregistré avec la province de Québec), son numéro de téléphone, son mail, son nombre de points de fidélité, d’une adresse de facturation, l’historique de ses commandes, et optionnellement sa carte de crédit avec son numéro et sa date d’expiration. Le client a la possibilité de noter chaque bière commandée.
* Afin de gérer les stocks, il est important de mettre à jour l’état des stocks avec la quantité disponible pour chaque produit dès qu’une vente est réalisée, qu’une livraison est reçue. Les commandes sont à récupérer directement au comptoir de commande, qui est à même l’entrepôt.
* Il faut également conserver l’historique des bons de commandes fournisseurs. Ses bons doivent regrouper les produits et quantités reçues, le coût de la commande, la date de la dernière livraison ainsi que le délai de livraison. Cet historique permet de faciliter le suivi entre les différents fournisseurs.

# Spécifications du système et des responsabilités des trois niveaux

Les fonctionnalités de notre application seront les suivantes :

* Création d’un compte client
* Connexion à son compte
* Un client peut consulter son historique d’achat
* Trier les bières selon leur sorte
* Choisir la quantité d’une bière à commander
* Ajouter une bière dans son panier de commande
* Retirer un élément de son panier
* Modifier la quantité de son panier
* Passer la commande de son panier

# Modélisation des données et de la BD

## Modèle entité-relation

**A close up of a calculator

Description automatically generated with medium confidence**

## Modèle relationnel

data\_beers (id: INT,

Name: VARCHAR (100),

Brewery: VARCHAR (100),

Style: VARCHAR (100),

Alcohol\_content: FLOAT (1),

Price: FLOAT (2),

Rating: INT)

data\_customers (pseudo: VARCHAR (50),

last\_name: VARCHAR (500),

first\_name: VARCHAR (500),

birth\_date: DATE,

email: VARCHAR (500),

phone\_number: DECIMAL,

Billing\_address: VARCHAR (500),

Credit\_card: VARCHAR (16))

credit\_Card (CC\_number: VARCHAR (16),

CC\_expiration\_date DATETIME)

Pwd (pseudo: VARCHAR (50),

Motdepasse: VARCHAR (300))

supplier\_order (id INT,

delivery\_date DATE,

Product\_id INT,

Cost FLOAT (2),

Quantity INT)

customer\_Order (Order\_id: INT,

Client\_id: VARCHAR (50),

Beer\_id: INT,

Quantity: INT,

Total\_price: FLOAT (2))

order\_item (Client\_id: VARCHAR (50),

Order\_id: INT,

Beer\_id: INT,

Quantity: INT,

Total\_price: FLOAT (2))

Stock (Beer\_id: INT,

Quantity: INT)

Rating (Beer\_id: INT,

Customer\_id: VARCHAR (50),

Rating: TINYINT)

# Gestion des données

La gestion des données est implémentée en utilisant PyCharm via le système de gestion de base de données MySQL. Avec ce niveau, toutes les données nécessaires au bon fonctionnement de l’application y sont sauvegardées. Afin d’importer les données contenues dans les fichiers .csv, un script Python est lancé par l’utilisateur afin de remplir les tables.

* Les données sur les bières (nom, type, brasserie, prix, etc.)
* Les données sur les comptes des utilisateurs (Pseudo, adresse courriel, carte de crédit, etc.)
* Les commandes des clients (nom de la bière, quantité, prix, etc.)
* Les commandes aux fournisseurs (date de livraison, prix, nom de la bière, etc.)
* Les données sur les cartes de crédit des clients (numéro, date d’expiration)

Ce premier niveau répond aux requêtes du deuxième niveau, soit la logique de l’application, qui lui-même répond aux requêtes du troisième niveau, l’interface utilisateur. En effet, lorsqu’un client se créer un compte par exemple (interface utilisateur), il rentre des informations (Nom, prénom, date de naissance, adresse courriel, pseudo, mot de passe, etc.) qui, via le niveau logique de l’application, seront transmises puis emmagasinées par le premier niveau, gestion des données, en utilisant un INSERT. Par la suite, lorsqu’un client cherche une bière en triant parmi les sortes disponibles, le niveau de gestion des données sélectionnera, à l’aide d’un SELECT, les bières répondant au filtre appliqué par le client. Enfin, quand un client ajoutera les bières à son panier et passera sa commande, la carte de crédit du client s’affichera, via un SELECT, (si le client avait choisi de sauvegarder sa carte). Une fois la commande passée, la table contenant les commandes des clients se mettra à jour avec la nouvelle commande.

Aussi, lorsqu’un client donne une note à une bière, la table qui contient les notes se mettra à jour et en utilisant une gâchette, la note afficher sur le site web se mettra elle aussi à jour.

# Logique de l’application

La logique d’affaire correspond au niveau le plus profond du programme de bière sans alcool. Il a un impact direct sur la performance des requêtes que l’utilisateur demande au server. Les différentes requêtes sont effectuées par python sur la base de données MySQL. Dans le fichier database.py, le programme se connecte à cette base de données puis effectue les différentes requêtes. Ils trient les bières par type, vas chercher les utilisateurs et les vérifies. Il ajoute des nouveaux utilisateurs, a la base de données, ajoutes des commandes au panier de la base de données. Ces différentes requêtes sont liées à des routes, Flask. Les routes Flask sont utiles pour passe de l’information. Dans le fichier server.py, les différentes routes pour transférer l’information pour les futurs requêtes y sont présente. On peut y voir les routes de vérification de compte, par exemple, qui vérifie l’utilisateur et le mot de passe et afficher une nouvelle page html, si la connexion est réussie ou échoué. Même les pages de base comme la page contacter-nous, sont créé par une route. Ce sont Ces routes qui forment les Template html. Il passe l’information de la base de données à l’utilisateur. Cela est utilisé pour connecter les demandes. Beaucoup d’erreur se sont manifesté à cet étape du processus, nous avons fait de notre mieux et rendue un programme fonctionnel dans son ensemble. Les routes communiquent directement avec le javascript de la partie interface utilisateur.

# Interface de l’utilisateur

L’interface utilisateur correspond à notre premier niveau. Il s’agit du seul niveau sur lequel l’utilisateur à un impact direct. Sur notre site internet, les utilisateurs ont la possibilité de créer leur compte, de se connecter, de visualiser les produits, d’ajouter les produits à leur panier et enfin de confirmer la réservation des bières.

Pour réaliser toutes ses actions, le client aura accès à la page principale sur laquelle la marque se présente. Ensuite, s’il se rend sur l’onglet suivant « Magasin », toutes les bières apparaissent. Nous avons fait appel à du JavaScript pour les récupérer et les afficher. Pour chaque bière, le client potentiel à accès à son nom, sa brasserie, son style, son prix et sa note. Le client a la possibilité de filtrer les bières en fonction de leur style ou de faire défiler la page vers le bas pour toutes les voir. Pour trier sur le style de bière, nous avons opter pour des boutons radios pour que l’utilisateur sache qu’il ne peut sélectionner qu’un seul type à la fois. En cliquant sur une bière, le client est invité à saisir la quantité qu’il souhaite acheter et à l’ajouter au panier ou annuler la démarche et ainsi retourner à la liste de bières. Il ne lui reste plus qu’à payer en ligne. Dans le cadre du projet, nous n’avons pas gérer le paiement sur internet, nous considérons que dès que le produit est ajouté au panier, le paiement est directement validé et la commande apparait alors dans Panier. Nous faisons de nouveau appel à JavaScript pour afficher l’historique des commandes du client connecté.

Dans l’onglet Panier, le client va pouvoir retrouver toutes l’historique de ses commandes.

Dans le troisième onglet, « Compte », le client peut soit se créer un compte soit se connecter s’il dispose déjà d’un pseudo et d’un mot de passe. Pour récupérer la saisie des utilisateurs, nous avons utilisé des formulaires HTML en adaptant le type de donnée attendu, par exemple pour les informations tels que le nom, le prénom nous avons demandé du texte, pour les informations comme la date de naissance, nous avons opté pour un type date pour faciliter la gestion de l’information dans les autres niveaux et enfin pour le mot de passe, nous avons utilisé un type mot de passe pour que les utilisateurs puissent se connecter à l’abris des regards indiscrets. Initialement, nous avons fait le choix d’attribuer nous même un mot de passe à chaque utilisateur mais cela nous a ensuite handicapé pour crypter le mot de passe stocker. Enfin pour les données comme les coordonnées bancaires, nous demandons un nombre.

Finalement, une fois connecté, le message d’accueil affiche « Bonjour » suivit du pseudo de l’utilisateur. A noter que, la création du compte ne fonctionne que si l’utilisateur rempli tous les champs correctement. Et un message lui confirme que le compte à bien été créé et que le mot de passe que nous lui avons déterminé lui sera transmis.

Pour la création d’un compte, nous somme passé par des routes qui faisait références à des fonctions pythons qui venaient modifier les bases de données. De même pour vérifier l’existence d’un compte. Enfin, dans le dernier onglet, « Contacter-nous », l’utilisateur pourra retrouver nos coordonnées. Concernant l’apparence de notre site, nous somme rester sur un CSS primaire, nous n’avons malheureusement pas pu travailler cet aspect de notre site internet.

# Gestion de l’équipe et division du travail.

Le travail a été séparé en attribuant des niveaux de difficulté aux tâches à effectuer. Ensuite, ces tâches ont été séparées de manière que chaque membre de l’équipe contribue au même nombre de points et à une difficulté similaire. Il a aussi été décidé d’imposer un délai à la réalisation des tâches.

Tableau 1- Répartition des tâches de chaque membre de l’équipe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tâches | Points | Niveau de difficulté | Nom | Date limite |
| L’énonciation du problème et de ses exigences | 1 | 1 | JM | 16/03 |
| Le modèle entité-relation et relationnel du système | 2 | 2 | LS | 18/03 |
| Le modèle relationnel du système | 1 | 1 | JM | 18/03 |
| Les fonctionnalités du niveau serveur de BD : Création des relations | 3 | 1 | LS | 20/03 |
| Les fonctionnalités du niveau serveur de BD : Requêtes et routines | 3 | 2 | LS | 20/03 |
| Les fonctionnalités du niveau serveur de BD : Indexation et optimisation | 1 | 3 | CB | 01/04 |
| Les fonctionnalités du niveau serveur de BD : Normalisation des relations | 1 | 2 | JM | 01/04 |
| Les fonctionnalités de la logique d’affaire | 2 | 1 | JM | 15/04 |
| Les fonctionnalités de l’interface utilisateur | 2 | 3 | JM | 27/03 |
| Les fonctionnalités de l’interface utilisateur | 2 | 2 | CB | 27/03 |
| La sécurité du système | 1 | 3 | CB | 15/04 |
| L’organisation et la gestion de l’équipe, et division des tâches | 1 | 0 |  |  |
| Démo vidéo | 4 | 3 | LS | 15/04 |