

Cahier des charges d'un restaurant

Dans le petit restaurant des COCO Dingo, dirigé par M.CHHUON, il a besoin d'un outil de gestion afin d'identifier et d'optimiser ses ventes. Pour cela, deux axes d'études sont à prendre en compte : le client et le produit. Afin de réaliser ces deux tâches, nous allons établir un **historique des commandes** comportant à la fois les noms des clients et les plats qu'ils ont commandés.

Tout d'abord, notre patron cherche à identifier ses **clients fidèles** pour leur apporter une attention particulière. Comme tout restaurant qui se respecte, le client est roi. C'est donc pour cela qu'il faut profiler et déterminer les plus fidèles. A l'aide de l'historique, nous allons pouvoir relever le **nombre d'occurrence d'un nom/prénom et les stocker dans une base de données**. Puis, il ne restera plus qu'à **comparer les différentes occurrences** entre elles pour déterminer les clients fidèles. L'obtention d'un **numéro de téléphone** d'un client pourrait aussi être utile pour supprimer un grand nombre de contraintes et de quiproquos. D'autre part, l'étude de **l'âge** peut être intéressante. Un étudiant et un fonctionnaire n'auront pas les mêmes moyens financiers. Ces informations concernant le client formeront la classe **CLIENT** du programme.

Une classe **DATE** va être créée pour nous permettre d'amplifier la précision des informations à propos du passage d'un client. De plus, pour les occurrences cela va nous faciliter le comptage.

Ensuite, chercher à identifier les **plats favoris** des clients est essentiel pour un restaurant cela permet au patron d'établir sa carte pour optimiser ses recettes. Pour cela, l'historique va associer à un nom tous les plats qu'il a pris c'est-à-dire **l'entrée, le plat de résistance, le dessert et la boisson**. A l'aide de cette étude, on pourra déterminer les « **Best Sellers** » du restaurant, comme pour la recherche des clients fidèles, il faudra **relever la plus grande occurrence d'un plat** dans la base de données. Ces objets formeront la classe **PRODUITS** du programme.

Ces deux classes vont être associées dans une classe **COMMANDE** pour faciliter la mise en place de l'historique. L'étude du restaurant sera donc faite dans la classe COMMANDE où les comparera l'occurrence des données qu'on cherche (clients fidèles, Best Sellers, etc...).

Dans notre société et notre temps, le pouvoir d'achat est devenu très instable. Une autre étude peut être faite par notre programme : **l'étude des prix**. Notre programme va associer **le prix total de chaque commande à un nom** dans l'historique. Le prix d'un produit peut aussi influencer le client. Par cette étude, le patron pourra adapter ses menus et ajuster les prix si cela est possible. Dans la classe COMMANDE, le prix total sera donc indiqué. Et dans la classe REPAS, chaque objet sera accompagné de son prix.

La contrainte majeure du programme réside dans le relevé des commandes. Une grande commande de plusieurs personnes peut être faite par une seule personne. L'exemple type est le cas de la famille, en général le père paye, et donc, il commande pour toute la famille. Dans ces cas-là, on devra associer tous les plats à une même personne. Et si par la suite, un des enfants revient seul sans son père, notre programme ne considérera pas le fils comme un client fidèle. Une autre contrainte du programme est qu'il n'y a pas la possibilité à l'erreur. Toute modification est impossible après avoir entrée les données.

HISTORIQUE
Command m_commande;
historique(Command commande);
listo stockageC(listo malisteC, Command m_commande); liprod stockageP(liprod malisteP, Command m_commande); listo rechercheListo(listo malisteC, Command m_commande); void rechercheProd(liprod malisteP, Command m_commande); int occurrenceC(listo malisteC, string Nom, std::string Prenom); int occurrenceP(liprod malisteP, std::string NomProduit);

DATE
Int m_jour Int m_mois Int m_année Date(int jour, int mois, int annee) int Askjour(); int Askmois(); int Askannee(); void afficher();

COMMANDE
Client m_client; Produit m_product; Date m_date; Command(Client client, Produit product, Date date); string get_clientL(); string get_clientF(); string get_productN(); int get_productP(); int get_dateJ(); int get_dateM(); int get_dateA(); int prixTotal(); void affiche();

PRODUIT
string m_nom; float m_prix; Produit(string name, float price); string get_name(); float get_price(); void afficher();

CLIENT
string m_first_name string m_last_name; int m_age int m_tel; Client(string first_name, string last_name, int age,int tel) string get_first_name(); string get_last_name(); int get_age(); int get_tel(); void afficher();