Projet CSI - L3 Miage

Système d’achat en ligne & drive

CHABOISSIER Maxime

PAPELIER Romain

BESSON Léonard

BURTEAUX Pierre

Table des matières

[Modèle Conceptuel des Données 4](#_Toc418021385)

[MCD 4](#_Toc418021386)

[Explication de certains choix 5](#_Toc418021387)

[Dictionnaire de données 6](#_Toc418021388)

[Fonctionnalités 7](#_Toc418021389)

[Coté Client 7](#_Toc418021390)

[Navigation 7](#_Toc418021391)

[Gestion du compte 8](#_Toc418021392)

[Gestion des commandes 8](#_Toc418021393)

[Coté administrateur 8](#_Toc418021394)

[Business intelligence 8](#_Toc418021395)

[Modèles Conceptuels des Traitements 9](#_Toc418021396)

[Arrivée d’un client sur le site / connexion 9](#_Toc418021397)

[Validation du panier 10](#_Toc418021398)

[Annulation d’une commande 11](#_Toc418021399)

[Edition des bilans 11](#_Toc418021400)

[Ajout d’un produit au panier 12](#_Toc418021401)

[Modèles organisationnels des traitements 12](#_Toc418021402)

[Vérification des informations bancaires 12](#_Toc418021403)

[Edition des bilans financiers 13](#_Toc418021404)

[Contraintes d’intégrité 13](#_Toc418021405)

[Modèle Physique des Données 14](#_Toc418021406)

[Technologies et Techniques utilisées 15](#_Toc418021407)

[Langages du web 15](#_Toc418021408)

[gestion du projet 16](#_Toc418021409)

[Répartition du travail 16](#_Toc418021410)

[durée de réalisation 16](#_Toc418021411)

[Méthode de travail 16](#_Toc418021412)

[Fonctionnement du site 17](#_Toc418021413)

[LA PAGE D’ACCUEIL 17](#_Toc418021414)

[LA CONNEXION DU CLIENT 17](#_Toc418021415)

[L’AJOUT D’UN PRODUIT AU PANIER 17](#_Toc418021416)

[LE PANIER 18](#_Toc418021417)

[LE CHOIX DES HORAIRES 18](#_Toc418021418)

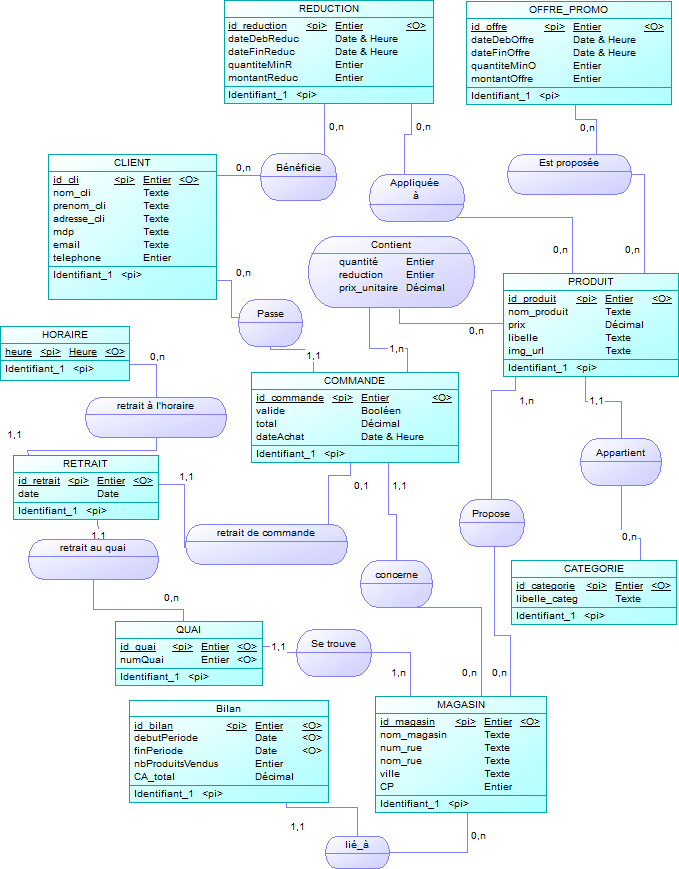
[LA GENERATION DU BILAN 18](#_Toc418021419)

[CE QU’IL RESTE A FAIRE 18](#_Toc418021420)

[Conclusion 18](#_Toc418021421)

# Modèle Conceptuel des Données

## MCD

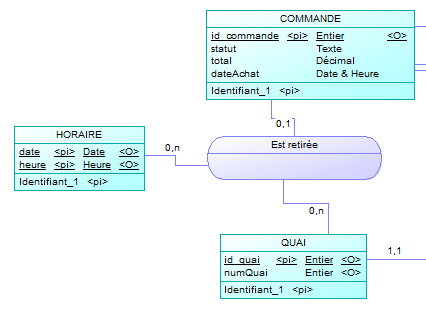


## Explication de certains choix

Nous avons choisi de représenter les magasins que possède la société, en effet, une chaine de grande distribution peut posséder plusieurs magasins dans plusieurs villes. Nous avons donc découpé l’adresse d’un magasin pour une meilleure possibilité de recherche d’un magasin à l’arrivée sur le site par exemple.

Nous avons lié les produits à des magasins, car certains magasins peuvent proposer des produits qu’on ne trouve pas dans d’autres, comme par exemple certains « Produits du Monde », beaucoup plus diversifiés dans un magasin présent dans une grande ville.

Par ailleurs, nous avons également trouvé intéressant de lier un bilan à un magasin, afin d’avoir un bilan des ventes pour chaque magasin. Cela permet de comparer les magasins par « rentabilité » selon leur implantation géographique, dans une optique Business Intelligence. Toujours dans cette optique, le fait de lier un Produit d’une Commande à un Magasin permet également de voir les taux de ventes des produits en fonction de leur localisation, et de voir ainsi s’il faut arrêter la vente d’un produit très peu vendu dans tel magasin.

Nous avions également une relation ternaire entre Commande, Quai et Horaire dans un premier lieu, qui devait respecter les contraintes Commande -> horaire, quai et horaire, quai -> commande.

Mais cette relation, une fois passé en MPD, faisait que l’on pouvait avoir une même commande retirée dans deux quais différents au même horaire.

Nous avons donc créé une entité Retrait qui fait office d’intermédiaire, comme indiqué sur le MCD actuel. Cela alourdit le MCD, mais une fois le MPD généré, on obtient toujours une seule table qui relie Commande, Horaire et Quai.

Le panier est représenté par une commande en cours, c’est-à-dire une commande non validée (attribut « validé » = false). Comme il n’y a qu’un seul panier possible par client, il n’y a qu’une seule commande non validée par client, ce qui la rend facile à retrouver avec une simple requête SQL. Nous avons choisi ce système par rapport à une entité Panier à part, étant donné qu’il faudrait, à la validation de chaque panier, recréer une entité Commande avec les mêmes relations vers les mêmes produits, ce qui peut faire beaucoup de requêtes de modification d’un seul coup et ralentir l’exécution. Dans notre cas, la validation d’un panier change simplement la valeur de l’attribut « validé » dans la commande en cours, et son annulation entraine la suppression de cette commande et de ses relations avec les produits.

Concernant les réductions et les offres promotionnelles, elles sont bien deux offres distinctes : une offre promotionnelle s’applique à tous les clients sur un ou plusieurs produits, alors qu’une réduction s’applique sur certains produits et pour certains clients (récompense fidélité, etc). Nous avons choisi de ne pas lier Client, Reduction et Produit par un ternaire, car une seule et même réduction peut être appliquée sur une gamme de produits, et certains clients peuvent bénéficier de cette même réduction.

# Dictionnaire de données

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | **Code** | **Type de données** | **Description** |
| adresse\_cli | ADRESSE\_CLI | Texte | adresse du client |
| CA\_total | CA\_TOTAL | Décimal | chiffre d'affaire total du magasin |
| CP | CP | Entier | code postal du magasin |
| date | DATE | Date | date de retrait de la commande |
| dateAchat | DATEACHAT | Date & Heure | date de validation (achat) de la commande |
| dateDebOffre | DATEDEBOFFRE | Date & Heure | date de début de la période de l'offre promo |
| dateDebReduc | DATEDEBREDUC | Date & Heure | date de début de la période de la réduction |
| dateFinOffre | DATEFINOFFRE | Date & Heure | date de fin de la période de l'offre promo |
| dateFinReduc | DATEFINREDUC | Date & Heure | date de fin de la période de la réduction |
| debutPeriode | DEBUTPERIODE | Date | début de la période du bilan |
| email | EMAIL | Texte | email du client (login de connexion) |
| finPeriode | FINPERIODE | Date | fin de période du bilan |
| heure | HEURE | Heure | heure de retrait de la commande |
| id\_bilan | ID\_BILAN | Entier | id du bilan |
| id\_categorie | ID\_CATEGORIE | Entier | id de la catégorie de produit |
| id\_cli | ID\_CLI | Entier | id du client |
| id\_commande | ID\_COMMANDE | Entier | id de la commande |
| id\_magasin | ID\_MAGASIN | Entier | id du magasin |
| id\_offre | ID\_OFFRE | Entier | id de l'offre promo |
| id\_produit | ID\_PRODUIT | Entier | id du produit |
| id\_quai | ID\_QUAI | Entier | id du quai |
| id\_reduction | ID\_REDUCTION | Entier | id de la réduction |
| id\_retrait | ID\_RETRAIT | Entier | id du retrait de commande |
| img\_url | IMG\_URL | Texte | url de l'image du produit |
| libelle | LIBELLE | Texte | description du produit |
| libelle\_categ | LIBELLE\_CATEG |  | libelle de la catégorie |
| mdp | MDP | Texte | mot de passe du client |
| montantOffre | MONTANTOFFRE | Entier | montant de la remise due à l'offre promo en pourcentage |
| montantReduc | MONTANTREDUC | Entier | montant de la remise due à la réduction en pourcentage |
| nbProduitsVendus | NBPRODUITSVENDUS | Entier | nombre de produits vendus dans le bilan |
| nom\_cli | NOM\_CLI | Texte | nom du client |
| nom\_magasin | NOM\_MAGASIN | Texte | nom du magasin |
| nom\_produit | NOM\_PRODUIT | Texte | nom du produit |
| nom\_rue | NOM\_RUE | Texte | nom de la rue du magasin |
| num\_rue | NUM\_RUE | Texte | numéro de la rue du magasin |
| numQuai | NUMQUAI | Entier | numéro du quai |
| prenom\_cli | PRENOM\_CLI | Texte | prénom du client |
| prix | PRIX | Décimal | prix unitaire du produit |
| quantiteMinO | QUANTITEMINO | Entier | quantité minimale d'un produit acheté requise pour l'offre promo |
| quantiteMinR | QUANTITEMINR | Entier | quantité minimale d'un produit acheté requise pour la réduction |
| quantité | QUANTITE | Entier | quantité d'un produit dans une commande |
| reduction | REDUCTION | Entier | réduction finale appliquée à la commande (offre OU réduc) |
| statut | STATUT | Texte | statut de la commande (en cours, validée) |
| telephone | TELEPHONE | Entier | numéro de téléphone du client |
| total | TOTAL | Décimal | total à payer pour la commande |
| ville | VILLE | Texte | ville dans laquelle se trouve le magasin |

# Fonctionnalités

Nous avons listé les différentes fonctionnalités attendues pour la future application de gestion du Drive à la fois pour le client et pour l’entreprise (administrateur).

## Coté Client

### Navigation

Rechercher un produit

Naviguer dans les catégories

Afficher la page d’un produit

Trier par prix, par nom

### Gestion du compte

Inscription

Connexion

Déconnexion

### Gestion des commandes

Afficher historique des commandes

Valider/régler une commande

Annuler une commande

Choisir heure de retrait de la commande

Ajouter un produit dans la commande en cours (panier)

Supprimer un produit dans la commande en cours (panier)

## Coté administrateur

### Business intelligence

Consulter bilans de ventes

# Modèles Conceptuels des Traitements

## Arrivée d’un client sur le site / connexion

/C1 & C2 & C3

(/C1 & C2 & /C3) ou (/C1 & /C2)

C1

/C4

C4

/C1 & /C2

CHOIX CONFLIT MAGASIN

* Si utilisateur choisit de supprimer le panier (C4)
  + Création nouveau panier
* Sinon (/C4)
  + Corrige le magasin et restaure le panier en session

CONNEXION

* Vérifier login et mot de passe
* Si mot de passe ou login incorrect (C1)
  + Afficher erreur de connexion
* Sinon (/C1)
  + Vérifier existence d’un panier
  + Si un panier existe déjà (C2)
    - Si le panier concerne un magasin différent du magasin choisi (C3)
      * Proposer à l’utilisateur de supprimer l’ancien panier ou de choisir le magasin du panier existant
    - Sinon (/C3)
      * Restaure le panier en session
  + Sinon (/C2)
    - Création d’un nouveau panier

B

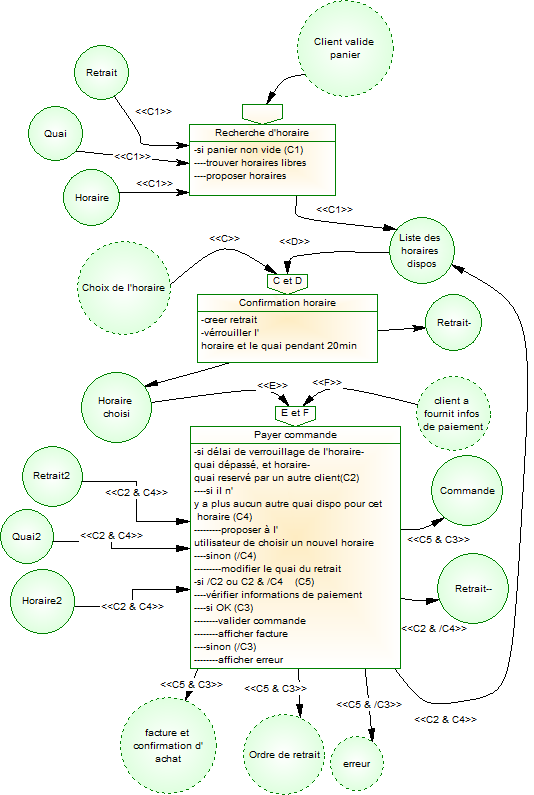
A et B

A

CHOIX DU MAGASIN

* afficher la liste des magasins

## Validation du panier



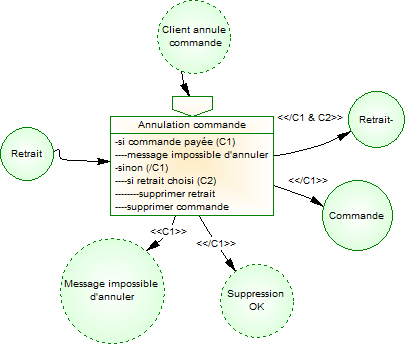
L’idéal lors de la validation du panier est de choisir l’horaire avant de payer, comme c’est le cas sur le site Carrefour par exemple. En effet, la plupart des sites marchands fonctionnent déjà sur le principe du « payé – terminé », où une fois que les informations bancaires sont saisies et validées, l’utilisateur reçoit automatiquement un mail de confirmation et peut quitter le site.

L’idée de payer avant de choisir l’horaire ne sera donc pas évidente pour tout le monde, et il risque donc d’y avoir des commandes payées, mais sans horaire retrait, tant l’utilisateur va valider sa commande puis quitter la page sans valider l’horaire.

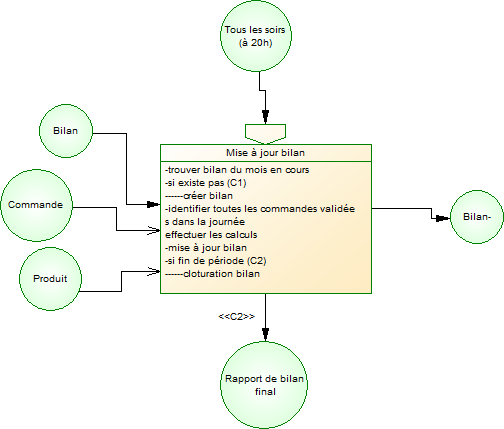
Le mécanisme sera donc proche de celui de la réservation de billets chez SNCF : l’utilisateur choisit un horaire, ce qui verrouille cet horaire et le quai associé pendant 20 minutes. Si l’utilisateur n’a pas payé sa commande dans les 20 minutes, l’horaire et le quai sont remis à disposition d’autres utilisateurs.

Si le quai choisi au départ a été attribué à une autre commande une fois que l’utilisateur a payé la sienne, on cherche un autre quai libre pour cet horaire. Si aucun autre quai n’est libre pour cet horaire, on invite l’utilisateur à choisir un nouvel horaire.

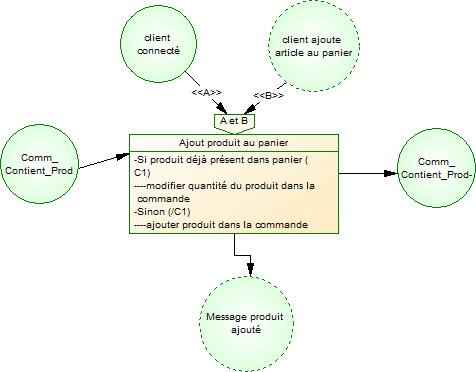
## Annulation d’une commande



## Edition des bilans



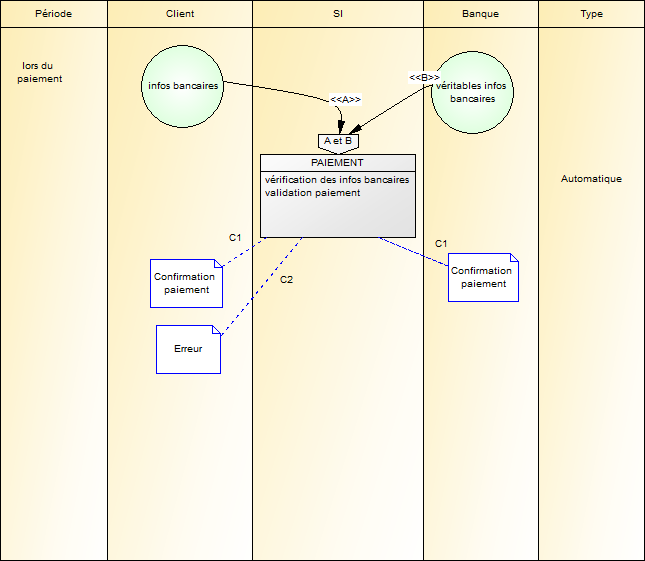
# Ajout d’un produit au panier



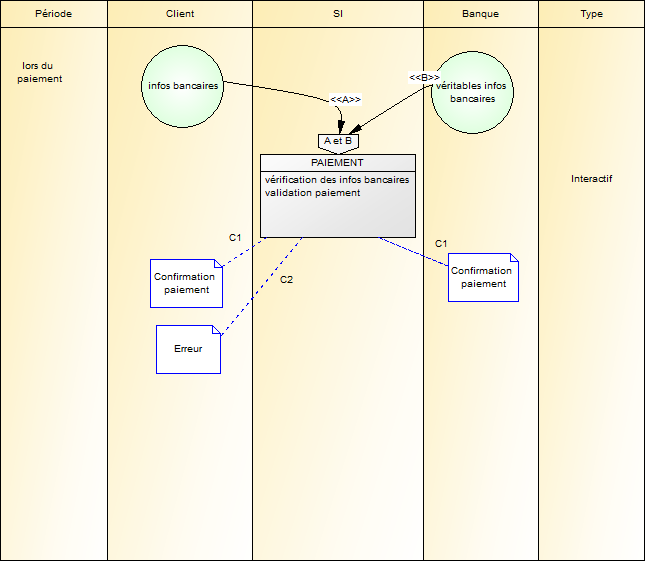
Pour pouvoir ajouter un produit, il faut que d’un côté le client clique le bouton d’ajout du produit, et que de l’autre le client soit bien connecté (l’objet interne « client connecté » symbolise la suite de l’opération « CONNEXION » si la connexion se passe bien. Il ne représente pas une condition interne à l’opération d’Ajout au panier, puisque cette opération ne peut pas exister si le client n’est pas préalablement connecté (ex : bouton d’ajout au panier « grisé » dans ce cas).

# Modèles organisationnels des traitements

## Vérification des informations bancaires



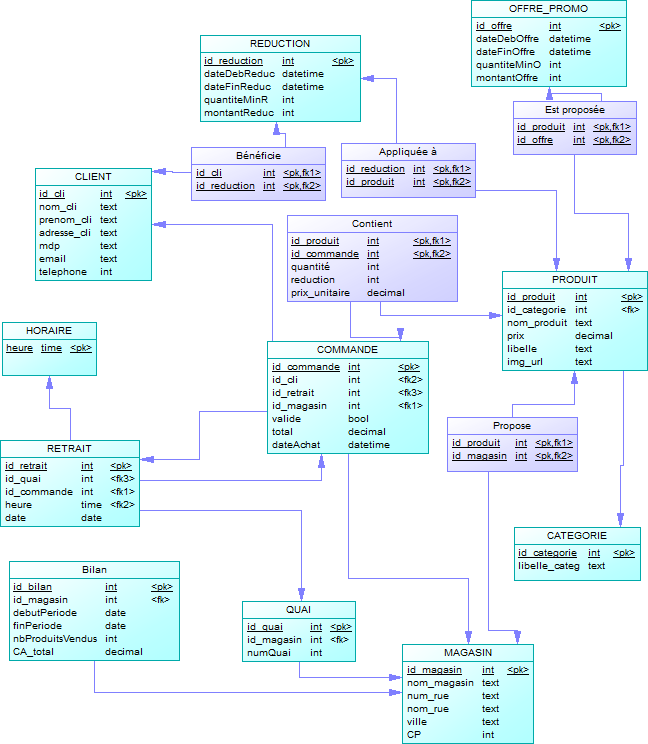
## Edition des bilans financiers



# Contraintes d’intégrité

|  |  |
| --- | --- |
| **Forte** | **Faible** |
| Une seule réduction ou offre promotionnelle peut être appliquée sur un produit à l'ajout au panier |  |
| Le montant total de la commande devra être égal aux prix des produits (après les réductions) |  |
| Un client ne peut ajouter ou supprimer un produit à une commande validée |  |
| Une commande ne peut être retirée en deux fois (deux horaires différents ou deux quais différents) |  |

# Modèle Physique des Données



# Technologies et Techniques utilisées

- Pour réaliser ce projet, nous avions le choix entre le langage PHP et le langage Java. Notre choix c’est porté sur le PHP, que nous trouvions plus pratique pour ce type de projet et que nous maîtrisons. De plus, nous avons déjà utilisé ce langage pour la réalisation de plusieurs projets. Enfin, ce langage s’intègre facilement sous EasyPHP. Il permet également de gérer les requêtes SQL de manière efficace.

## Langages du web



- Nous utilisons MySQL car il est adapté au modèle Active Record et convient parfaitement pour ce projet de part sa gratuité et son efficacité. Son langage possède de plus de nombreuses interactions avec le langage PHP, qui nous permet de mieux gérer les requêtes SQL.  De plus nous avons déjà réalisés plusieurs projet avec ces technologies contrairement à JDBC.

Programmateur d’événements: Cette fonctionnalité permet de programmer des évènements qui vont déclencher des requêtes SQL, des procédures stockées. Nous utilisons le programmateur d’événements pour la création du bilan, ainsi que la gestion des réservations de quai.

- Nous utilisons le modèle MVC (Modèle/Vue/Contrôleur) car il est depuis longtemps maintenant un modèle de référence dans le développement d’applications. Il permet un développement simple sur 3 axes. De plus, la maintenance et la mise à jour de l’application est aisé car il n’est pas nécessaire de devoir modifier tout le code.

Active Record :  L’Active Record est un design pattern pour lire les données d'une [base de données](http://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es). Les attributs d'une [table](http://fr.wikipedia.org/wiki/Table_(alg%C3%A8bre_relationnelle)) sont encapsulés dans une [classe](http://fr.wikipedia.org/wiki/Classe_(informatique)). Ainsi l'objet, instance de la classe, est lié à un [tuple](http://fr.wikipedia.org/wiki/Tuple) de la base.

Ajax : L’Ajax est destiné à réaliser de rapides mises à jour du contenu d'une page Web, sans qu'elles nécessitent de rechargement visible par l'utilisateur de la page Web.

Transmission des informations: L’utilisation des variables de session étaient indispensable pour ce projet. En effet, de nombreuses informations transitent entre les pages et doivent êtres stockées. Les méthodes GET et POST ont aussi été utilisé pour l’envoi de données.

# gestion du projet

## Répartition du travail

Pour réaliser ce projet nous avons combiné séances de travail en groupe et travail individuel. Lorsque tous les membres du groupe étaient présents sur Nancy, nous avons organisé des séance de travail en groupe, essentiellement pour poser les bases du projets. Ensuite, nous avons utilisé Github qui nous a permis de centraliser le travail de chacun.

## durée de réalisation

Durée total :

Nous avons organisé notre première réunion le 19 février. Deux rapports intermédiaires ont été déposés. Le projet avançait donc de façon régulière. Nous l'avons terminé le 26 avril, soit une durée totale d'environ deux mois.

Répartition du temps par tâches :

* MCD, MCT, MOT, contraintes d’intégrités : 15%
* Développement : 75%
* Rapport : 10%

## Méthode de travail

* Répartition des tâches
* Travail individuel sur les tâches à effectuer
* Travail en commun sur les tâches à effectuer
* Propositions d’idées
* Mise en commun sur un dépôt GitHub

# Fonctionnement du site

Nous avons crée un site basé principalement sur les fonctionnalités et donc possédant un style très épuré. Ce site est découpé en deux pages HTML, la première étant une page permettant de choisir un magasin dans une liste. ensuite, nous avons une page index qui sera similaire pour chacun des choix. Cette page répertorie nos différents produits.

## LA PAGE D’ACCUEIL

La page d’accueil propose l’intégralité des produits vendus par le magasin, sans distinction des catégories. Le choix des catégories se fait à partir d’une liste affichée sur la gauche de cette page.

## LA CONNEXION DU CLIENT

Tant que l’utilisateur n’est pas connecté, celui-ci a uniquement comme possibilité de consulter les produits. Nous avons donc insérer à cette page d’accueil un bloc permettant de se connecter en tant que client de nos magasins à l’aide de son email et de son mot de passe. Une fois le client connecté, un panier est crée pour ce client.

## L’AJOUT D’UN PRODUIT AU PANIER

Une fois connecté, le client peut remplir son panier. Pour ajouter un produit, il suffit de cliquer sur celui-ci, afin d’avoir une description détaillée du produit. Puis le client doit sélectionner la quantité et cliquer sur “Ajouter au panier”.

S’il se déconnecte et se reconnecte le système va chercher dans la base la dernière commande portant le numéro du client connecté et qui n’est pas encore validé pour afficher son panier.

Une fois des produits ajoutés au panier, nous pouvons cliquer sur ce dernier qui nous permettra d’avoir un résumé de tous les produits que nous avons ajouté au panier.

## LE PANIER

Le panier contient le nom des produits, la quantité, le prix unitaire, et le prix total. Le client pourra alors valider la commande à l’aide d’un bouton, nous choisissons alors le jour de retrait de notre commande, et ensuite nous pouvons choisir l’horaire de retrait.

## LE CHOIX DES HORAIRES

Des tests sont effectués pour savoir si un quai est disponible à l'horaire donnée, sinon le client est averti. Après avoir crée un retrait, un évènement temporel est crée : le client à 20 minutes pour valider son retrait sinon ce dernier est annulé pour permettre de laisser le créneau a une éventuelle autre commande.

## LA GENERATION DU BILAN

La génération du bilan se fait via un évènement MySQL (creationBilan.sql). La création du bilan se fait chaque mois à une heure définie. L’événement insère dans la table BILAN de la base la période couverte par le bilan, la quantité de produit vendus et le montant total des ventes.

## CE QU’IL RESTE A FAIRE

La fonctionnalité concernant la génération des réductions et des offres promotionnelles n’a pas été implémentée, faute de temps.

# Conclusion

Ce projet était très intéressant dans le sens où nous devions concevoir un site commercial de A à Z, de la phase de conception (MCD, MCT, …) à la phase de production, en passant par la réalisation du code. Il nous a permis de nous poser de vraies questions concernant l’importance de bien concevoir nos modèles avant de programmer, dans le but de gagner du temps et savoir dans quelle direction aller lors de l’implémentation des fonctionnalités.

Notre site est actuellement hébergé sur un serveur web :

http://romainpapelier.fr/CSI/Projet\_CSI/View/html/choix\_mag.php