

# LeBarD : Mise en Oeuvre

---

## Sommaire

- Acteurs
- Le choix des fonctionnalités
- Les Problématiques
  - Brainstorming avec Trello
  - Transformation en tâches
  - Les fonctionnalités
- L'implémentation
  - Démarrage
    - Affectation des tâches depuis Trello
    - Découpage fichier du projet
    - Découpage du site
  - Problèmes rencontrés
- Propositions d'améliorations

## Acteurs

- Rémi Mollandin ~~aka Lineal aka ibdw.fr à vendre 5\$~~
- Théo Peuckert ~~aka Toast aka la salière~~
- Antoine Chapusot ~~aka Chap aka Fantôme~~
- Xavier Grimaldi ~~aka Xerx aka Application God~~
- Benoit Scholl ~~aka Gefelic aka DB Man~~

## Le choix des fonctionnalités

### Les Problématiques

Les problématiques auxquelles répond l'application sont les suivantes :

- "Puis-je me prendre un café ? Combien d'argent il me reste sur mon compte au bar ?"
- "Pourquoi ne me reste-t-il que X€ ? Quelles ont été mes dernières transactions ?"
- "Qu'y a-t-il de disponible au bar ? Leur reste-t-il des Twix ?"
- "Quand est le prochain événement du bar ? J'aimerais savoir sans aller consulter mes mails..."

### Brainstorming avec Trello

Pour le cahier des charges, nous nous sommes reposé sur Trello pour communiquer nos idées autour du projet. Nous avons donc pris 30min de temps pour mettre des mots sur ce que nous a évoqué le projet.

### Transformation en tâches

Suite à cela, nous avons préciser certaines idées vagues. Il nous a fallu aussi épurer les cartes afin de ne garder que les plus à propos. Puis atomiser ces dernières en tâches unitaires que l'on a finalement prioriser selon leurs degré d'importance.

## Les fonctionnalités

- User
  - Voir son solde
  - Voir ses transactions
  - Voir le catalogue
  - Voir les nouvelles
- Barman
  - Effectuer une transaction
    - Rechercher un client
    - Remplir une commande
    - Débiter client
  - Créditer compte client
  - Visualiser les transactions d'un client
  - Ajouter des produits au stock
    - Créer un nouveau produit
    - Incrémenter le stock d'un produit existant
- Admin
  - Ajouter des annonces (CRUD)
  - Modifier le prix des produits
  - lecture seule de tous les objets en base
    - infos client
    - infos produit
    - infos annonces
    - infos barman

## L'implémentation

### Démarrage

#### Affectation des tâches depuis Trello

Toujours via Trello, nous nous sommes affecté les tâches unitaires. En prenant comme modèle le template docker fourni avec le sujet, nous avons pu nous abstraire de la partie technique middleware de tout projet web.

Les rôles affectés à chaque membre de l'équipe :

- Rémi Mollandin : Backend & Infra
- Théo Peuckert : Backend & Tests
- Antoine Chapusot Management projet
- Xavier Grimaldi Frontend & Features
- Benoit Scholl Base de données

## Découpage fichier du projet

Au niveau du squelette du projet, nous avons opéré le découpage suivant :

- **Database** : tout repose dans data/, l'architecture de la base de données se veut simple.
- **Backend** : dans le dossier module, avec pour chaque objet (Product /news/Client...) un dossier : entity, repository, hydrator contenant réciproquement : les caractéristiques de l'objet, les requêtes bases de données, les méthodes pour passer de l'un à l'autre.
- L'**autoloader** permet de charger les dépendances sur les modules du backend (namespaces)
- **Frontend** :
  - public/ correspond à la gestion du code php, à l'initialisation du code et des variables, et à l'appel des vues
  - view/ contient les vues html
  - public/assets/ contient toutes les ressources (js/images/etc...)
  - public/CSS/ contient les fichier CSS
  - data/ possède 3 copies de la DB
    - un sample avec des valeurs exemples
    - un export vide sans données avec seulement la structure de la DB
    - la DB actuelle, celle utilisée en prod

Nous avons ajouté un router initialisé via le fichier ".htaccess", et qui pointe sur index.php. C'est index.php qui s'occupe de charger les pages nécessaires en fonction de l'URL donné. Si l'URL est mauvaise ou inconnue, redirection sur la page 404.php, avec "mademoiselle oups"

- **Tests**

Dans le dossier public/test/ nous avons disposé des tests qui peuvent être lancés avec php.unit

## Découpage du site

Le site tiens compte de la gestion du responsive qui pourra permettre d'adapter le contenu à une tablette et rendre l'application utilisable avec une tablette directement au bar : plus simple d'utilisation.

Deux parties au site :

- Partie utilisateur avec connexion obligatoire par Arise Id pour s'authentifier et accéder à son espace client pour y consulter son solde, son compte, ses informations, ses statistiques de consommation, les transactions effectuées, les news du bar, le catalogue du bar et autres
- Partie administrateur /console , pour effectuer des commandes, créer et gérer des fonctionnalités du site: Gestion utilisateurs, news, produits, visualisation des transactions globales, etc...
  - Gestion des news: CRUD news avec enrichissement de la gestion avec un éditeur de page HTML intégré, permettant l'upload d'images sur serveur (page de couverture + image dans le contenu HTML)

- Gestion des produits: CRUD produit + shortcut disponible pour commander rapidement un produit sur la page de console partie commande.

La sécurité des pages est gérée au moyen de variable de session objet utilisateur/admin.

## Problèmes et difficultés rencontrés

- Peu de problème de communication entre les membres de l'équipe

Chacun a trouvé sa place plutôt naturellement dans l'équipe et le partage des tâches a été respecté dans les grandes lignes.

- Sous-estimation de la charge de travail

La plupart des projets sont victimes de retards et/ou de dépassements de budget. Puisque nous n'avons pas de budget et une deadline fixe, nous avons subi un autre type de risque : la charge de travail.

- Typage de la monnaie

Le typage de la monnaie sur le site s'est initialement fait en décimale, donc en **double float**. Mais ce format apporte des soucis d'affichage lors d'opérations sur ces nombres. Une solution proposée aurait été de changer pour des **integer**. Trop tard pour changer et trop d'impacts, nous avons laisser tel quel.

- Le stress de la deadline

Pour diverses raisons qui peuvent être : une idée de feature au dernier moment, une feature oubliée, le soucis du détail, un bug récalcitrant, l'équipe subit le stress de la deadline. Cependant, d'un commun accord, nous avons éviter toute modification de dernière trop impactante.

## Propositions d'améliorations

- Responsive design

En effet, la version actuelle du site contient un bon nombre de pages qui s'affichent correctement sur tablette, un peu moins qui s'affichent sur smartphone. Sachant que cette appli est destiner à obtenir rapidement une info de manière mobile, le *design responsive* est une partie importante de l'ergonomie de l'application

- Remplacer dans le débit la saisie du mot de passe barman par une authent NFC (ID office de clé unique)

Aujourd'hui, les barmans ont tous un mot de passe unique qu'ils utilisent pour débiter les clients. Il se trouve que chaque étudiant possède une carte NFC pour accéder à l'entrée de l'école ou au garage à vélo par exemple. Puisque les ID des cartes sont uniques, il serait intéressant d'attacher cette identifiant aux barmans qui pourront ainsi débiter en quelques secondes. Cette fonctionnalité ne peut pas être implémentée facilement aujourd'hui, les navigateurs n'étant pas tous aptes à lire le NFC sur des périphériques.

- Améliorer les shortcuts

Le site possède pas mal de raccourcis claviers. Il serait intéressant d'avoir un retour d'expérience sur ceux-ci de la part des barmans et ajouter ceux qu'ils souhaitent