

Alyacoubi Reda

Brault Clément

Champion Antoine

Orain Alexandre

**Rapport de projet**

**Refonte du logiciel du foyer des élèves**

État de l’art

Le foyer des élèves est au centre de notre vie aux Arts et Métiers.

L’objectif de ce projet est d'améliorer sa gestion en effectuant une refonte du logiciel actuellement utilisé.

Actuellement, ce logiciel permet seulement de décompter les consommations des personnes sur leurs comptes, il permet également d’avoir un historique très sommaire des consommations et l’argent restant sur les comptes.

En l'état, il s'agit d'une interface front-end liée à quelques macros Access, le logiciel est très basique et ne permet pas d’avoir une gestion des stocks précise, ce qui est pénalisant pour la comptabilité.

L’idée serait donc de créer un logiciel qui permettrait une gestion des stocks et des approvisionnements automatisée et poussée, mais aussi avec une interface web adaptée aux ordinateurs et aux smartphones avec la possibilité pour un utilisateur de recharger son compte et de consulter son historique, et pour les administrateurs, avoir accès à un panneau de contrôle complet.

I – Organisation et gestion de projet

Afin de cadrer le projet et de ne pas partir en aveugle, nous avons utilisé plusieurs séances afin de définir la portée du projet à l’aide de différents outils.

Notre premier objectif était de définir la portée du projet, les évolutions par rapport au logiciel précédent et de mettre en lumière des liens entre les idées des différents membres. Nous avons donc réalisé une Mind Map (carte des idées).

Dans la continuité de cette Mind Map, nous avons réalisé une liste exhaustive des fonctions que notre plateforme allait offrir. En parallèle, nous avons soumis nos idées à la promotion et réalisé des sondages afin d’avoir le plus de retours possibles.

Nous leur avons soumis deux questions, et voici les réponses qui revenaient le plus souvent :

**Q :** Que faudrait-il changer dans le système actuel ?

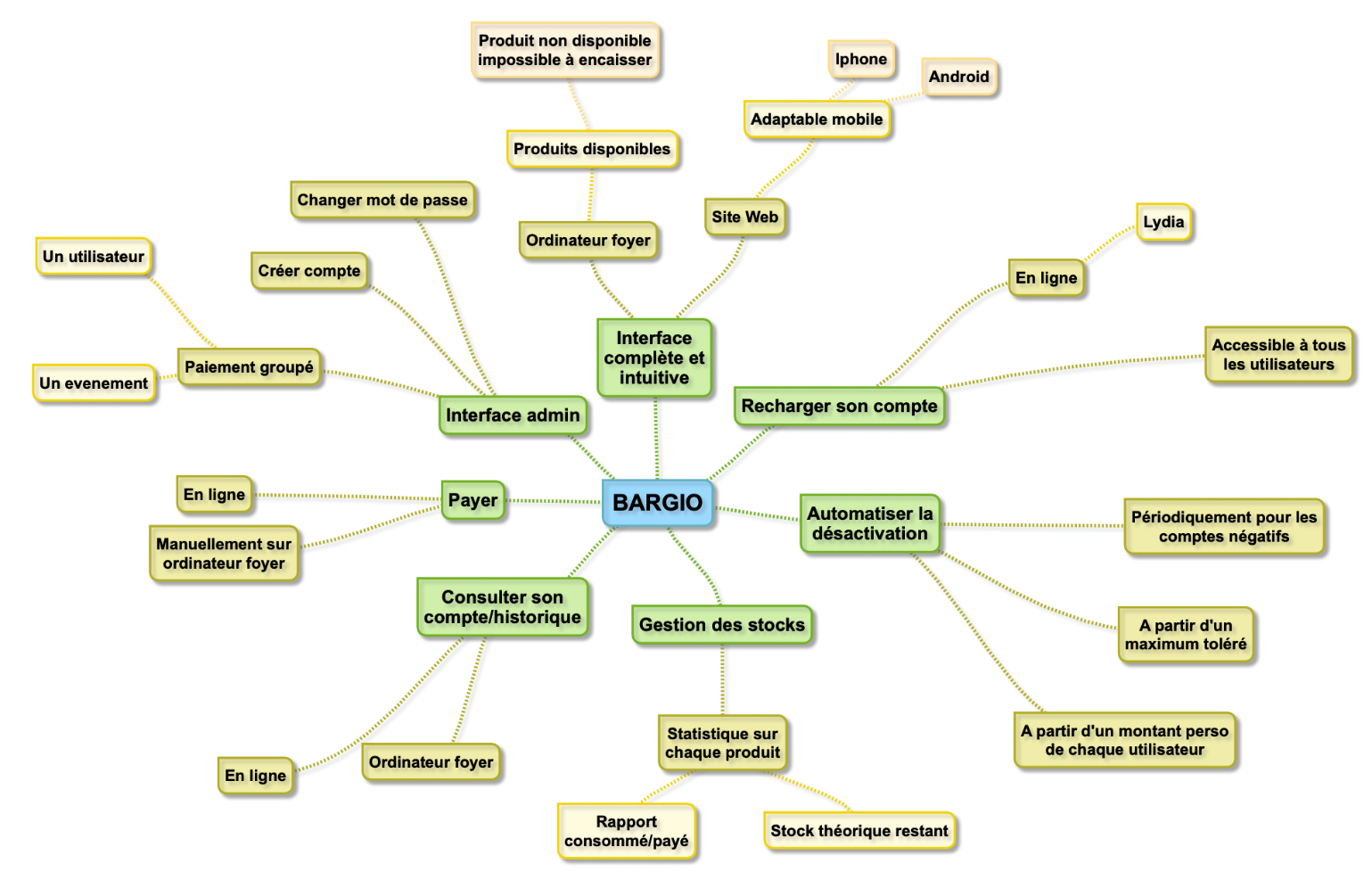
**R :** Davantage de précisions sur l’historique personnel de consommations (type de produit, quantité, date)

Recharge du compte autonome et en ligne

Changement du design de l’interface utilisateur

Fusion avec compte Borgia

Protection du compte par mot de passe personnel



*Fig 1. Mind Map*

**Q :** Quelle(s) fonctionnalité(s) supplémentaire(s) voudriez-vous ajouter dans ce programme ?

**R :** Déblocage du compte automatique suite au rechargement

Afficher des statistiques (grands consommateurs, records journaliers...)

Recevoir une notification sur téléphone portable lors d’une consommation

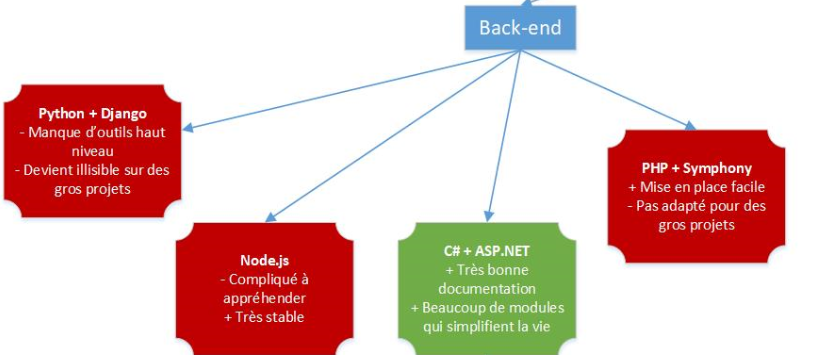
Seuil de blocage de compte personnalisable

Pouvoir annuler une consommation dûe à une faute de frappe

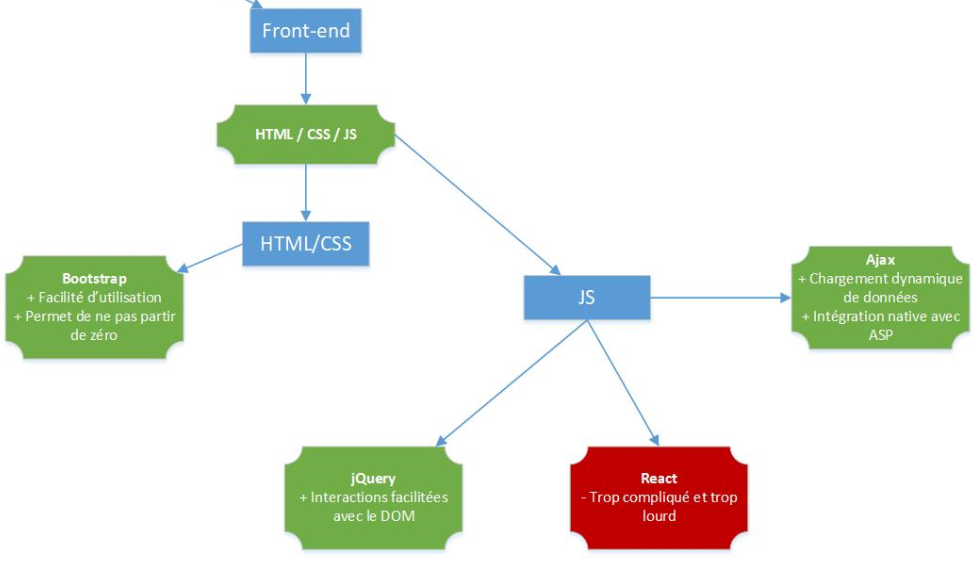
Une fois terminée, cette liste des fonctionnalités a pu être exploitée afin d’entamer une réflexion sur quelles technologies utiliser.

Une plateforme web se sépare en deux parties : le front-end qui concerne tout ce qui est exécuté côté client (design, layout, scripting…), et le back-end qui est exécuté côté serveur (gestion du hardware, bases de données, sécurité…).

A la fois pour le front et le back-end, il existe des *frameworks*, qui sont des infrastructures logicielles spécialisées permettant de créer les fondations d’autres projet. Nous avons dû étudier et comparer différents frameworks afin de trouver le plus pertinent. Nous avons choisi ASP.NET pour le back-end (cf fig. 2), et l’association de bibliothèques Bootstrap/jQuery pour le front-end (cf fig. 3).



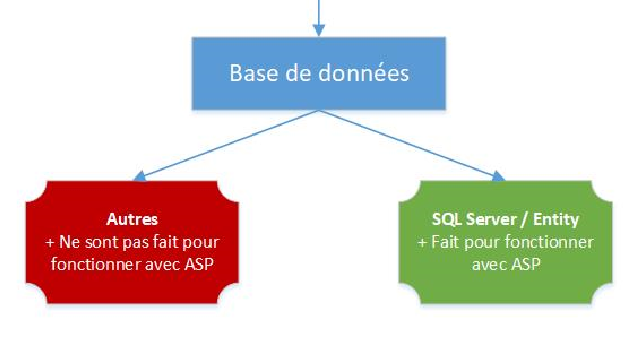
*Fig. 2 : Arbre des technologies (framework back-end)*



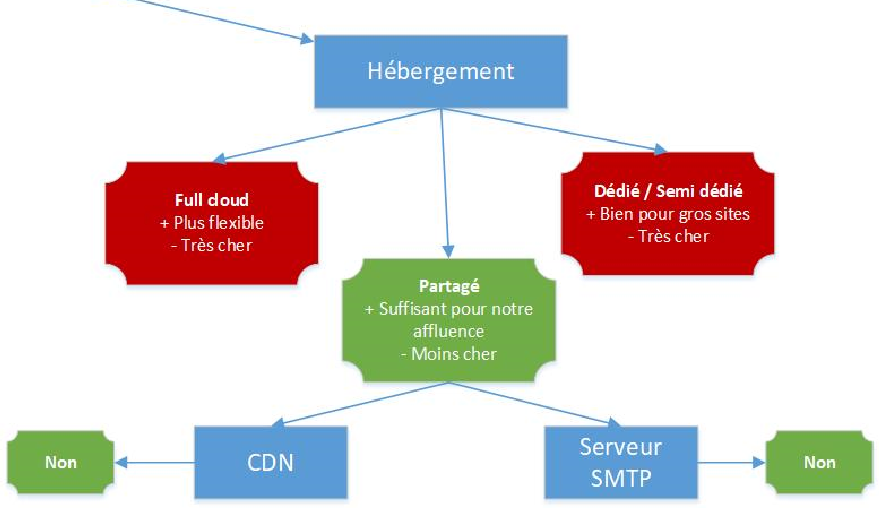
*Fig. 3 : Arbre des technologies (framework front-end)*

Une fois le framework déterminé, il a fallu choisir un *design-pattern*, qui représente une manière de programmer, d’organiser son code et de le hiérarchiser. Effectivement, notre framework en proposait deux différents, nous avons choisi Razor Pages (cf fig. 4).

Ces choix étant effectués, il ne nous restait plus qu’à choisir un système de bases de données, ainsi qu’un modèle d’hébergement. Nous sommes partis sur SQL Server couplé à son framework Entity, et un hébergement partagé (cf fig. 4,5).



*Fig. 4 : Arbre des technologies (bases de données)*

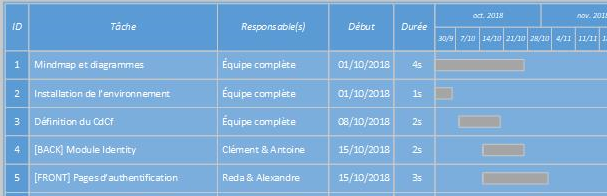


*Fig. 5 : Arbre des technologies (hébergement)*

II – Répartition des tâches et familiarisation avec les outils

A ce stade, nous avions une liste exhaustive des fonctionnalités à implémenter ainsi qu’un arbre des technologies complet.

La prochaine étape a donc été de réaliser un diagramme Gantt afin de relier les fonctionnalités aux technologies et de planifier le travail (cf fig. 6).



*Fig. 6 : Vue partielle du diagramme Gantt*

Le diagramme étant bien fourni (25 tâches sur 36 semaines), nous ne vous proposons qu’une vue partielle. Le diagramme entier est disponible en annexes.

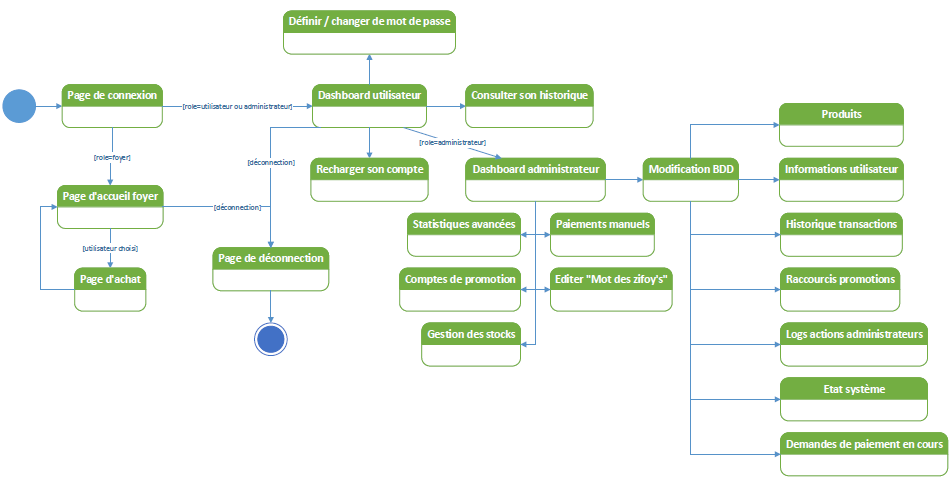
Lors de la réalisation de ce Gantt, nous avons constaté qu’il était compliqué de paralléliser différentes tâches : le back-end a besoin que le front-end soit au même avancement que lui pour pouvoir en réaliser une implémentation. Cela nous a amené à une logique de travail en 2 équipes, la première s’occupant exclusivement du front-end (MM. Alyacoubi et Orain), et la seconde exclusivement du back-end (MM. Brault et Champion). Les étapes intermédiaires telles que la tenue de projet, les phases de test et le passage en production sont cependant gérées par l’équipe entière.

Afin de réaliser ce diagramme, nous avons dû beaucoup nous documenter afin d’anticiper au maximum les différentes étapes du projet, et de savoir comment les appréhender.

Tous les membres de l’équipe n’ayant pas forcément la même expérience avec les outils de programmation, un temps d’appréhension et de familiarisation avec les outils de programmation a également été mis en place, et nous avons commencé avec les modules les plus simples à réaliser, c’est-à-dire la création de comptes et les pages de connexion/déconnection.

Pour le back-end, nous avons codé en C# avec ASP.NET Core 2 et Javascript sur Visual Studio, et pour le front-end, en HTML+CSS et Javascript sur Adobe Brackets.

Suite à cette première période, nous avons réalisé une arborescence complète de notre site afin de faire apparaître clairement la navigation au sein de celui-ci (cf fig. 7).



*Fig. 7 : Arborescence du site*

III – Module d’authentification

La première étape que nous avons réalisée concernait le module d’authentification : créer des bases de données d’utilisateurs qui peuvent se connecter à l’aide d’un mot de passe optionnel, gérer ce mot de passe et se déconnecter.