

Projet tutoré S3/S4 :

Médiathèque

Groupe n°9

Client : Comité Social et Économique Entreprise Thales Brest

Tuteur : Alain MILLET

Groupe : Arthur DUHOUX – S3A1, Théo GANARD – S3A1, Thibault GROSJEAN – S3A1 et Lisa DOYEN – S3A1 (chef de projet)

Année : 2020-2021

Remerciements

Tout d'abord, nous tenons à remercier notre client, Olivier DOYEN, bénévole au sein du CSE de Thales Brest, qui nous a aidé et suivi tout au long de notre projet. Il a su nous guider tout au long de ce projet. Grâce à sa confiance, l'équipe de développement a pu mener à bien un maximum de missions demandées.

Nous adressons ensuite nos remerciements à notre professeur et tuteur, Monsieur Alain MILLET. Ses conseils nous ont été d'un soutien précieux.

Enfin, nous remercions également Madame PATERLINI qui nous a beaucoup aidé concernant la rédaction du rapport et la soutenance pour le projet.

Sommaire

Remerciements.....	3
Introduction.....	5
1. Cahier des charges.....	6
1.1. Généralités.....	6
1.2. Contexte et définition du projet.....	6
1.3. Objectifs de réalisation.....	7
1.4. Démarche et organisation.....	7
2. Réalisations.....	9
2.1. Ce qui a été réalisé par rapport au S3.....	9
2.1.1 Les améliorations.....	9
2.1.2 Les ajouts.....	10
2.2. Ce qu'il reste à faire.....	17
2.2.1 Les API pour musique et jeux.....	17
2.2.2 L'ajout d'une messagerie.....	18
2.2.3 Création d'un système de sauvegarde des BDD et d'un système de restauration en cas de dysfonctionnement.....	18
2.2.4 L'ajout d'une application mobile.....	19
2.2.5 Amélioration des statistiques.....	19
2.2.6 Amélioration des performances et de l'optimisation du site.....	19
3. Bilan.....	20
3.1. Bilan humain.....	20
3.2. Bilan technique.....	20
Conclusion.....	21
Table des illustrations.....	22
Sommaire des annexes.....	23

Introduction

Le Comité Social Économique (CSE) de Thales comprend une médiathèque. Cette dernière a pour but de réunir 4 entités différentes qui sont : la vidéothèque, la musicothèque, la ludothèque et la bibliothèque. Le besoin d'un site ou d'une application pour gérer la médiathèque est donc présent.

Nous avons donc constitué un groupe de 4 personnes qui rassemble les compétences et connaissances exigées pour ce projet. Nous sommes des personnes organisées et impliquées.

Notre état d'esprit pour ce projet était de relever un challenge. En effet, ce projet est ambitieux et notre but était de nous dépasser afin de pouvoir obtenir un site internet à la hauteur de notre travail et de nos attentes.

Ainsi, ce rapport se déroule en 3 grandes parties : tout d'abord, le cahier des charges sera présenté, puis, nous vous présenterons tout ce qui a été réalisé et nous finirons par un bilan.

1. Cahier des charges

1.1. Généralités

Pour la réalisation de ce troisième projet tuteuré, nous avons formé un groupe constitué de 4 étudiants : DUHOUX Arthur, GANARD Théo, GROSJEAN Thibault et DOYEN Lisa. Cette dernière est élue chef de projet. Nous sommes le groupe numéro 9 et nous avons décidé de réaliser un site web permettant la gestion d'une médiathèque.

1.2. Contexte et définition du projet

La médiathèque en question fait partie des activités du comité social et économique (CSE) de l'entreprise Thales située à Brest. Actuellement, la médiathèque est organisée au travers de 4 commissions distinctes et autonomes qui sont la bibliothèque, la vidéothèque, la musicothèque et la ludothèque. Un projet de réorganisation est en cours afin de regrouper toutes ces activités en une seule commission et de réagencer également les locaux en conséquence dans l'objectif de création d'une médiathèque. Dans ce contexte, le besoin d'un site ou d'une application pour gérer la médiathèque est clairement exprimé et nécessaire.

Actuellement, il existe un logiciel datant d'environ 20/25 ans, développé pour une exploitation en réseau fermé (environ 7/8 machines) avec un partage d'une base de données basiques recensant tous les articles. Les fonctionnalités du logiciel sont très limitées avec une ergonomie simpliste et surtout sans aucune visibilité pour tous les emprunteurs. Suite à ce constat, la volonté du CSE est de se doter d'une application en phase avec les technologies actuelles et de proposer des services importants aux emprunteurs via Internet (consultation des médias, réservation en ligne, etc.).

Les parties prenantes pour ce projet sont le CSE représenté par Olivier DOYEN qui est bénévole au sein de la commission Bibliothèque, le tuteur académique Alain MILLET et notre groupe de 4 étudiants.

Ce projet a retenu notre attention pour plusieurs raisons :

- la diversité des langages de programmation à utiliser, (PHP, SQL) ainsi que l'utilisation du framework *Symfony* et du moteur de *template* Twig, est une source de motivation ;
- le contexte et la finalité du sujet sont très concrets avec comme objectif final une exploitation à grande échelle de l'application à terminaison ;
- l'hébergement du site sur Heroku ou sur un serveur distant tel que PlanetHoster.

Enfin, l'opportunité de dialoguer avec un client nous met en situation réelle et nous devons démontrer nos capacités d'écoute, d'analyse, de compréhension du besoin et des attentes tout en communiquant au fur et à mesure des avancées techniques réalisées.

1.3. Objectifs de réalisation

Selon les exigences du client, les principales fonctionnalités du site web sont regroupées en 6 grandes parties distinctes. En effet, le site doit pouvoir autoriser, surveiller/maintenir, administrer, enregistrer les mouvements, visiter et réserver. Ces fonctionnalités sont détaillées sous la forme de schéma en annexe 1.

Les difficultés techniques du projet sont :

- Mise en place d'une base de données conséquentes : il faut pouvoir importer l'existant (classe 10 000 articles) et ensuite répondre à une gestion complexe. En effet, le besoin de différents types d'utilisateurs amène à devoir adapter les niveaux d'autorisations et donc la mise en place d'une gestion de beaucoup de cas de figure.
- Mise en place Heroku liée à *Symfony* peut s'avérer complexe en raison de l'intégration de PostgreSQL comme système de gestion de base de données et le manque d'information et de documentation sur le Web.

Un danger important à éviter absolument est de s'éloigner du besoin et donc de réaliser un produit qui ne répondrait pas aux attentes de notre client.

1.4. Démarche et organisation

Le type de méthode que nous allons utiliser correspond au cycle en V. Nous allons descendre le V, tout d'abord, bien définir le besoin pour être convaincu de l'architecture et, par la suite, développer l'application avec à la fin une seule livraison. En phase de développement, des versions partiellement fonctionnelles seront soumises au client afin d'évaluer l'ergonomie par exemple. Nous suivrons

donc un cycle en V global avec des livraisons partielles dans la remontée du V pour soumission au client.

Pour la répartition des tâches à réaliser au sein du groupe, nous avons réalisé un suivi des étudiants visible sur Github en annexe n°2.

Nous allons utiliser GitHub pour partager le contenu et l'avancement du projet, Google Drive pour échanger des documents et pour préparer les présentations et les rapports. La plateforme Discord sera utilisée pour pouvoir communiquer régulièrement au sein du groupe. Nous échangeons aussi, en appel vidéo, avec le client grâce à Discord.

Des échanges de mails fréquents avec le client et avec le tuteur du groupe sont également mis en place.

Nous sommes partis sur un rythme d'une réunion hebdomadaire avec le groupe afin de nous synchroniser sur nos différentes activités. Une réunion de groupe avec le tuteur est également prévue une semaine sur deux.

Concernant les rendez-vous avec les clients, ils sont planifiés quand c'est nécessaire, pour faire le point sur l'avancée du projet ou pour répondre aux éventuelles questions des développeurs.

2. Réalisations

2.1. Ce qui a été réalisé par rapport au S3

2.1.1 Les améliorations

Durant le 4e semestre, nous avons réalisé un certain nombre d'améliorations dans notre site internet.

2.1.1.1 La charte graphique

Concernant la charte graphique, nous avons décidé d'améliorer les couleurs du site web et également réalisé le logo de la médiathèque.

Durant le 3e semestre, les couleurs de notre site reposaient essentiellement sur de l'orange, du noir et du rouge. À l'origine, ces couleurs avaient été choisies plus ou moins aléatoirement. De plus, notre client nous avait fait part de son désaccord avec la couleur orange qui selon lui dominait trop le site. Cependant, par manque de temps, nous avons laissé ces couleurs sur le site. De plus, cet aspect ne représentait pas notre priorité à ce moment-là.

Au début du quatrième semestre, l'équipe de développeurs a décidé de résoudre ce problème. Pour remédier à ce désaccord, nous avons donc étudié les différentes possibilités de palettes de couleurs. Il se trouve alors que nous avons décidé de garder l'orange et le rouge mais qu'elles apparaissent seulement sur des petits éléments afin de garder un peu de chaleur à notre site. Le réel changement est l'ajout du bleu calanque qui est la couleur dominante du site à présent. Nous avons décidé d'ajouter cette couleur, car elle est plus moderne et plus professionnelle que l'orange en couleur dominante. La palette de couleurs est visible en annexe 3.

2.1.1.2 La responsivité du site

Durant le 3e semestre, la responsivité du site ne faisait pas partie de notre priorité, car l'implémentation des fonctionnalités du site était plus importante à ce

stade. Nous nous sommes donc penchés dessus à partir du 4e semestre et nous l'avons considérablement amélioré.

Nous avons adopté pour la responsivité du site sur 3 tailles d'écrans : l'écran d'un ordinateur, l'écran d'une tablette et l'écran d'un téléphone, car ces 3 écrans représentent la taille des appareils les plus utilisés. Ainsi, chaque page du site s'adapte à la taille d'écran. La responsivité du site a été réalisée à l'aide du langage de programmation CSS.

À l'intérieur de notre dossier de code, les fichiers CSS sont bien organisés. En effet, chaque page du site possède un fichier CSS ainsi qu'un deuxième fichier pour la responsivité de cette page. Cependant, il est possible qu'une page ne possède pas de fichier pour sa responsivité, car elle est rattachée au fichier de responsivité du site global. Ce fichier contient des propriétés liées à des classes qui sont utilisables dans plusieurs pages.

Pour cet aspect, nous n'avons pas rencontré de difficultés particulières. De plus, cette étape s'est faite assez rapidement.

2.1.2 Les ajouts

2.1.2.1 Hébergements

Afin que notre client puisse visualiser notre avancée sur le projet et également utiliser et tester le site, nous avons décidé d'héberger le site internet.

Après plusieurs recherches concernant les différents hébergements, nous avons soumis nos propositions d'hébergeurs à notre client. Parmi ceux-ci, on peut citer les plus populaires tels que OVH Cloud et Hostinger.

Notre client a ainsi opté pour Hostinger, car les offres étaient intéressantes. De plus, en comparaison avec OVH Cloud, Hostinger convenait mieux à l'échelle de notre type de projet étant donné que nous ne souhaitions pas avoir le meilleur hébergeur possible.

Par la suite, notre client s'est créé un compte et il a acheté le nom de domaine qu'il souhaitait.

À partir de ce stade, notre groupe de développeurs était chargé de mettre à jour régulièrement le site. À chaque mise à jour, nous indiquons un numéro de version du site afin que notre client et notre groupe soit en accord avec les fonctionnalités réalisées ou améliorées pour chaque version.

L'hébergement de notre site nous permet donc d'avancer efficacement sur le développement du site en tenant compte des remarques apportées par notre client.

2.1.2.2 Système de réservation, d'emprunts, d'achats

Le développement du système de réservation, d'emprunts et d'achats n'était pas implanté à la fin du semestre 3 cependant sa conception était finalisée. Tous les principes de fonctionnement et les interfaces ont été définis dans un logigramme validé par le client. Ainsi ce système se découpe en plusieurs parties :

- le panier (côté utilisateur) ;
- la validation de la commande (côté utilisateur) ;
- changement du statut (utilisateur et admin/bénévole).

Le panier permet à l'utilisateur de constituer ses achats ou ses emprunts. Les fonctionnalités de bases sont établies : suppression d'un article du panier, vidage complet du panier, consulter le détail de l'article et la validation du panier. À noter que lors de la suppression d'un article l'utilisateur a la possibilité de l'ajouter dans ses favoris. Le stockage des données est généré par une table « panier » dans la base de données.

Si une personne place un article dans son panier, il devient indisponible pour les autres utilisateurs. Il y a donc un changement de statut de l'article. La demande du client est de ne pas bloquer l'accès à l'article pour les autres utilisateurs principalement si le panier n'est pas validé au bout d'un certain temps (choix de limiter la durée de vie du panier à 30 minutes). Le problème réside dans le fait que si l'utilisateur ne gère pas proprement son panier (validation de la réservation ou vidage du panier), l'article reste dans un état indisponible alors que ce n'est plus le cas. Il faut donc mettre en place un mécanisme qui réinitialise l'état de l'article à disponible. Pour y répondre, la solution technique mise en place est d'avoir un chronomètre qui signale à l'utilisateur combien de temps il lui restait avant que le panier se vide. En effet, le but du chronomètre était alors de réinitialiser le panier

pour éviter la monopolisation de l'article. Cependant, cette fonctionnalité n'a pas été implantée par manque de temps et de technique. À noter qu'une solution temporaire a été mise en place : une déconnexion automatique (cf. 2.1.2.5). Une meilleure solution serait de faire une session locale (sur la machine de l'utilisateur) avec une possibilité de sauvegarder le panier avant la déconnexion. Par exemple en mettant tous les articles dans les favoris ou en sauvegardant le panier.

Lorsque la commande est validée, les articles passent dans le statut « emprunté ». Ainsi un enregistrement dans la table du même nom est créé. S'ensuit le vidage du panier qui correspond à la suppression des lignes dans la table « panier ».

Une interface permet à l'utilisateur de consulter l'historique et ses commandes en cours. De la même façon, les bénévoles peuvent consulter l'arrivée de nouvelles commandes via une interface propre aux bénévoles et ainsi changer le statut de la commande. Au changement de statut de l'article, l'utilisateur reçoit une notification par mail et ce à chaque phase de la préparation de la commande. Une amélioration à faire serait de regrouper les articles issus d'un même panier pour permettre aux bénévoles de valider l'entièreté d'une commande d'une traite et non article par article comme c'est le cas actuellement.

2.1.2.3 CRUD Article complet et fonctionnel

Les opérations CRUD (*Create Read Update Delete*) sur les articles étaient des formulaires plus complexes que pour la plupart des autres tables.

En effet, selon le cahier des charges, il ne faut pas pouvoir supprimer un article de la BDD (Base De Données), quand un article est « supprimé » par un administrateur ou un bénévole, il est en réalité déclassé ce qui le rend inaccessible à l'emprunt et accessible à la vente.

De plus, la création et la modification d'un article nécessite de créer ou modifier des champs dans d'autres tables de la BDD. Par exemple, lorsque l'on crée un article, une « action » lui est associée automatiquement pour indiquer quel utilisateur à créer l'article et à quelle date. Ce système d'action doit être géré automatiquement par le formulaire pour différentes actions :

- achat d'un article ;
- création d'un article ;
- modification d'un article ;
- déclassement d'un article ;
- éventuel reclassement d'un article.

On peut ainsi obtenir un historique complet des actions effectuées sur l'article de l'achat jusqu'à son éventuelle sortie de la médiathèque, en passant par toutes les modifications.

Le champ « date d'achat » des formulaires de création et de modification d'un article ne fait donc pas référence à un champ de la table article en BDD, ce champ n'est donc pas irrigué automatiquement par *Symfony* dans le cas d'une modification d'article et il faut l'injecter manuellement.

De plus, chaque article est inclus dans un système d'entités qui le décrivent, par une relation *many to many* l'article est relié à aucune ou plusieurs de ces entités. Chaque entité à un type (auteur, éditeur, groupe, collection, acteur...) choisi parmi les types d'entités présents dans la BDD ainsi qu'un nom et éventuellement un prénom. Dans les formulaires de création et de modification d'article il y a donc une section qui permet d'ajouter, modifier ou supprimer une entité associée à l'article. C'est ce point qui a été le plus difficile à implémenter et qui a nécessité le plus de recherches. En effet, il faut inclure dans le formulaire concernant l'article autant de sous-formulaires concernant les entités que le souhaite l'utilisateur qui crée ou modifie l'article et il a donc fallu mêler les formulaires *Symfony*, des fonctions JavaScript qui actualisent dynamiquement ces sous-formulaires et macro *Twig* pour uniformiser le style de ces sous-formulaires.

2.1.2.4 Système de notification des récapitulatifs de commandes par mail

Ce système a été implémenté durant le 4^e semestre, et utilise le *bundle Mailer* de *Symfony* avec le *bundle Google-Mailer*. Un mail est envoyé à l'utilisateur et aux bénévoles lors de la validation de la commande et lors d'un changement de statut d'un article. Ainsi lors de la validation de la commande l'utilisateur reçoit un récapitulatif de sa commande avec les articles, les prix (si ce sont des achats) et le

total. Pour les améliorations, il faudrait créer un système de notification directement dans le site pour éviter d'être dépendant des mails, ou encore mettre en place un système de SMS. L'utilisateur a également la possibilité de consulter sa dernière commande depuis le site. On utilise le *bundle DomPDF* afin de générer un récapitulatif de la dernière commande. Une amélioration serait de pouvoir récupérer tous les récapitulatifs des commandes depuis un espace dédié sur le site.

2.1.2.5 Système de déconnexion automatique

Le fonctionnement d'emprunt choisi par le client est le suivant : lorsqu'un adhérent ajoute un article à son panier, le statut de l'article est immédiatement mis à jour ce qui le rend inaccessible à l'emprunt pour les autres adhérents.

Cela ne pose pas de problème si l'adhérent valide ensuite son panier, emprunte l'article et le rend à la fin de la période d'emprunt, mais si l'adhérent ajoute l'article à son panier et abandonne ensuite le site, l'article est inaccessible pour les autres adhérents alors que le premier n'est pas réellement en train de l'emprunter.

Nous avons donc pensé à plusieurs solutions notamment :

- un système de compte à rebours qui vide le panier si le panier n'est pas validé au bout d'un certain temps ;
- vider le panier à la déconnexion de l'utilisateur et implémenter un système de déconnexion automatique en cas d'inactivité prolongée.

Nous avons donc fait des recherches sur ce genre de système sans vraiment trouver de solution pour le système de compte à rebours.

La solution de déconnexion automatique a donc été la solution choisie même si l'implémentation reste à améliorer. En effet, la déconnexion s'effectue au moment d'une requête si le serveur n'a pas reçu d'autres requêtes de l'utilisateur depuis un temps défini. Ce qui veut dire que l'utilisateur est déconnecté à son retour et que son panier reste inactif pendant son absence, le problème n'est donc pas résolu avec cette implémentation.

2.1.2.6 Alimentation de la base données par les APIs

Afin de rendre le site le plus intéressant possible pour les utilisateurs, notre client nous a demandé d'assister la création d'articles avec la recherche

d'informations issues du web. En effet, toutes les données sont présentes dans des bases de type métadonnées très riches. Le fait de disposer d'un maximum d'informations (visuel, titre, auteur, résumé...) est primordial pour exploiter les fonctionnalités de recherche ou de filtrage du site. Ce point devient donc crucial pour la suite du développement.

Nous avons donc choisi d'utiliser des APIs qui nous permettent, grâce à l'ISBN (*International Standard Book Number*) ou au code EAN¹ d'un article, de récupérer des données concernant l'article en question.

L'API qui nous semblait la plus évidente et la plus puissante était celle d'Amazon, mais Amazon a changé la façon dont on accède à leur API et il est maintenant obligatoire d'être vendeur chez eux.

Nous nous sommes donc tournées vers des APIs moins connus, gratuite mais tout de même efficaces, pour les livres *Google Books* est la plus intéressante, elle nous permet de récupérer différentes informations sur l'article :

- titre ;
- auteurs / éditeurs ;
- date de publication ;
- image ;
- description.

Pour les vidéos nous avons utilisé principalement DVDFr qui nous permet de récupérer :

- titre ;
- auteur ;
- date de publication ;
- image ;
- catégorie.

En plus de ces APIs, nous avons implémenté d'autres APIs pour compléter les informations comme :

- Ebay ;
- Bibliosurf ;

1 EAN : code barre disponible sur de nombreux types de produits pour les identifier.

- GoodRead ;
- OpenLibrary.

Toutes les APIs que nous avons utilisées sont gratuites et illimitées en matière de requête, certaines ont seulement besoin d'un lien avec l'ISBN au bout comme *Google Books*, par exemple : `'https://www.googleapis.com/books/v1/volumes?q=isbn:' . $this->isbn`

avec `$this->isbn` correspondant à l'ISBN que l'utilisateur rentre.

Par contre, certaines ont besoin d'une clé pour se connecter, nous pouvons l'obtenir sur le site qui fournit l'API mais cela prend déjà un peu plus de temps.

Pour le choix des APIs nous nous sommes limités à 5 par catégorie car sinon le temps de chargement pour faire toutes les recherches devenait excessif et nous avons choisi les APIs qui fournissent le plus de données possibles.

L'utilisation d'API comme ceci est tout nouveau pour nous donc au début nous avons eu beaucoup de mal et nous avons dû investir énormément de temps de recherches sans forcément avoir de résultat au départ puis petit à petit nous avons commencé à obtenir des résultats satisfaisants.

Une de ces difficultés concerne l'API DVDFr, elle a deux options, la première option qui est illimitée permet d'obtenir environ 8 données différentes et l'autre option permet d'obtenir un maximum 200 requêtes par jour mais avec plus de 50 données. Nous voulions donc prendre la deuxième option, car elle contenait plus de données que la première. Mais, il a fallu contacter les gérants de l'API pour obtenir cette option malheureusement ils ne nous ont jamais répondu.

Les étapes d'enrichissement des données se font comme ceci :

1. On récupère le *gencode*² de l'article
2. On récupère les données via les différentes APIs
 - a. si c'est du JSON c'est simple il n'y a presque rien à faire
 - b. si c'est du XML c'est un peu plus compliqué il faut créer un élément XML et le transformer en JSON
3. On transforme ces données au format que l'on souhaite
 - a. dates en chaînes de caractères → objets *DateTime*

2 Gencode : Numéro d'identification de l'article (ISBN ou EAN)

4. On regroupe les différentes données récupérées des APIs dans un tableau à double entrée :
 - a. `donnees['auteurs']][0] = "JK Rowling", donnees['image']][3] = "image.png"`
5. On envoie ensuite ces données à un formulaire de création d'article, elles constitueront des options et on laisse ensuite le choix à l'utilisateur.

Pour la transformation d'une chaîne de caractères en format de date, nous avons eu quelques difficultés. Tout d'abord, certaines APIs ne renvoyaient que l'année ("2016" par exemple) ou encore que l'année et le mois ("2016/11" par exemple), or pour que l'objet *DateTime* comprenne que c'est une date il a besoin d'être sous le format jj/mm/aaaa. Nous avons donc dû ajouter automatiquement le premier mois et le premier jour, par exemple "2016" est transformé en "2016/01/01". D'autres APIs renvoient une date sous la forme "16, janvier 2015", faute de temps nous avons décidé de ne pas implémenter ce format de date et nous l'avons tout simplement supprimé des données récupérées.

2.2. Ce qu'il reste à faire

Le site web est à présent fonctionnel, mais il va de soi qu'il peut être amélioré, des fonctionnalités peuvent être également ajoutées.

2.2.1 Les API pour musique et jeux

Étant donné la difficulté d'accès des APIs et la priorité des catégories des articles, les articles de catégorie jeu et musique passent après les livres et les vidéos. Ainsi, l'ajout d'informations à partir des APIs pour les musiques et jeux n'a pas été fait. Cependant, la recherche des APIs a quand même été effectuée, mais nous avons rencontré beaucoup de difficultés à trouver des APIs, encore plus que pour les APIs concernant les livres et vidéos.

Concernant la musique, nous avons trouvé une piste vers une API mais le nombre de requêtes était beaucoup trop limité, nous étions limités à 5 requêtes par jour sachant que pour cette API, une donnée était égale à une requête. Nous avons donc pris la décision d'abandonner cette API.

2.2.2 L'ajout d'une messagerie

Au début de nos échanges avec le client, nous nous étions fixés un certain nombre d'objectifs, certains étaient prioritaires et d'autres optionnels. Ainsi, parmi les fonctionnalités optionnelles, certaines ne sont pas réalisées souvent par manque de temps, c'est le cas, pour la mise en place d'une messagerie entre les différents utilisateurs.

Les utilisateurs auront la possibilité de discuter avec un autre utilisateur directement sur le site internet. Aussi, un affichage des utilisateurs qui sont connectés en direct sera visible.

Cette fonctionnalité n'entrave pas le service principal du site qui est de pouvoir emprunter ou acheter un article. Ainsi, nous avons décidé dans un commun accord avec le client que cette fonctionnalité fera partie des dernières à implémenter. L'équipe de développement estime que ce fût un bon choix car si nous avions jugé cette fonctionnalité essentielle, le risque serait que le service principal de la médiathèque serait aujourd'hui dysfonctionnel voire non réalisé.

2.2.3 Création d'un système de sauvegarde des BDD et d'un système de restauration en cas de dysfonctionnement

Parmi les objectifs du site, la création d'un système de sauvegarde des BDD et la création d'un système de restauration en cas de dysfonctionnement du site étaient présentes. Malheureusement, ces aspects n'ont pu être mis en place malgré le fait que ces derniers soient importants.

Nous nous sommes penchés sur d'autres aspects du site que nous avons jugés plus essentiels que celui-là, comme le système d'emprunts ou encore l'alimentation d'informations sur les articles à partir des données récupérées par les APIs.

De plus les hébergeurs proposent généralement de faire des sauvegardes de la BDD et/ou du site, cette fonctionnalité est donc partiellement disponible via l'hébergement du site.

2.2.4 L'ajout d'une application mobile

Le développement de cette application représentait la finalité de notre projet. En effet, l'ajout d'une application mobile était d'actualité uniquement lorsque le site était totalement fini. Or, certaines fonctionnalités sont manquantes sur le site actuellement. Ainsi, nous n'avons pu réaliser cette application.

Le site est aussi responsive depuis ce semestre, il reste donc accessible par mobile même sans avoir une application dédiée.

2.2.5 Amélioration des statistiques

Pour connaître et évaluer la capacité d'emprunt sur une certaine période, il était nécessaire d'avoir une page dévoilant des chiffres comme le nombre d'emprunts par article, par catégorie, par genre... Cette page est accessible aux administrateurs uniquement.

Actuellement sur la page, nous pouvons visualiser un tableau accompagné d'un graphique qui représente les 10 premiers articles les plus empruntés. Cependant, cette page peut être améliorée. En effet, parmi ces améliorations, il serait possible de pouvoir choisir la période sur laquelle ces chiffres seraient calculés. De plus, ces données pourraient être téléchargeables au format pdf afin de garder un historique de ces chiffres.

2.2.6 Amélioration des performances et de l'optimisation du site

Étant donné la quantité de données que nous possédons dans notre site internet, il est évident que les performances et l'optimisation sont perfectibles. En effet, le site internet propose un service, il est donc indispensable que le site soit rapide et optimisé pour satisfaire les besoins du client. Des améliorations en termes de rapidité et d'optimisation du site sont donc possibles.

3. Bilan

3.1. Bilan humain

Concernant le bilan humain, nous sommes un groupe qui coopère efficacement. Nous sommes organisés et engagés dans le projet. Nous avons travaillé régulièrement tout au long de l'année. Notre projet reflète la quantité et la qualité du travail que nous avons fourni. Les compétences complémentaires des différents membres du groupe ont été bénéfiques pour le projet.

Concernant le client, il est impliqué dans le projet. Il communique beaucoup avec notre groupe de développeurs.

Afin de communiquer efficacement, nous avons instauré un système de questions/réponses via un tableau *excel* partagé sur *Google Drive* visible en annexe n°4.

3.2. Bilan technique

Concernant le bilan technique, nous avons décidé de résumer les objectifs atteints et non atteints par rapport aux objectifs initiaux définis au début du projet. Ce résumé apparaît sous forme de graphique et est visible en annexe n°5.

Ce comparatif nous permet d'établir un bilan technique de nos objectifs. Il nous permet d'avoir une vision globale des fonctionnalités du site et de voir que les plupart des fonctionnalités essentiels pour la médiathèque ont été réalisés même s'il reste possible de les améliorer.

Conclusion

Ce projet nous permet de concevoir un produit qui se réfère au monde de l'entreprise. L'enjeu crucial est de satisfaire les besoins du client et de répondre à ses exigences tout en respectant les délais qui nous ont été imposés.

Notre groupe de développement a su se dépasser et concevoir un site internet fonctionnel dans les délais impartis. C'était un projet ambitieux qui reste perfectible, mais nous pensons que nous avons réalisé un bon travail tout au long de cette année.

Ce projet est maintenant dans les mains du client. Le site pourra donc être proposé en tant que premier prototype pour le CSE de Thales.

Table des illustrations

Figure 1: CDC : Autoriser, Surveiller/Maintenir.....	I
Figure 2: CDC : Administrer.....	I
Figure 3: CDC : Enregistrer les mouvements, Réserver.....	II
Figure 4: CDC : Visiter.....	II
Figure 5: CDC : Acheter.....	III
Figure 6: CDC : Télécharger.....	III
Figure 7: Suivi des étudiants sur Github.....	IV
Figure 8: Palette de couleurs.....	V
Figure 9: Exemple de question.....	VI
Figure 10: Exemple de réponse.....	VII
Figure 11: Schéma présentant les objectifs réalisé, à améliorer et peu réalisé.....	VIII

Sommaire des annexes

Fonctionnalités principales.....	I
Suivi des étudiants.....	IV
Charte graphique.....	V
Tableau de questions réponses.....	VI
Travail réalisé.....	VIII

Fonctionnalités principales

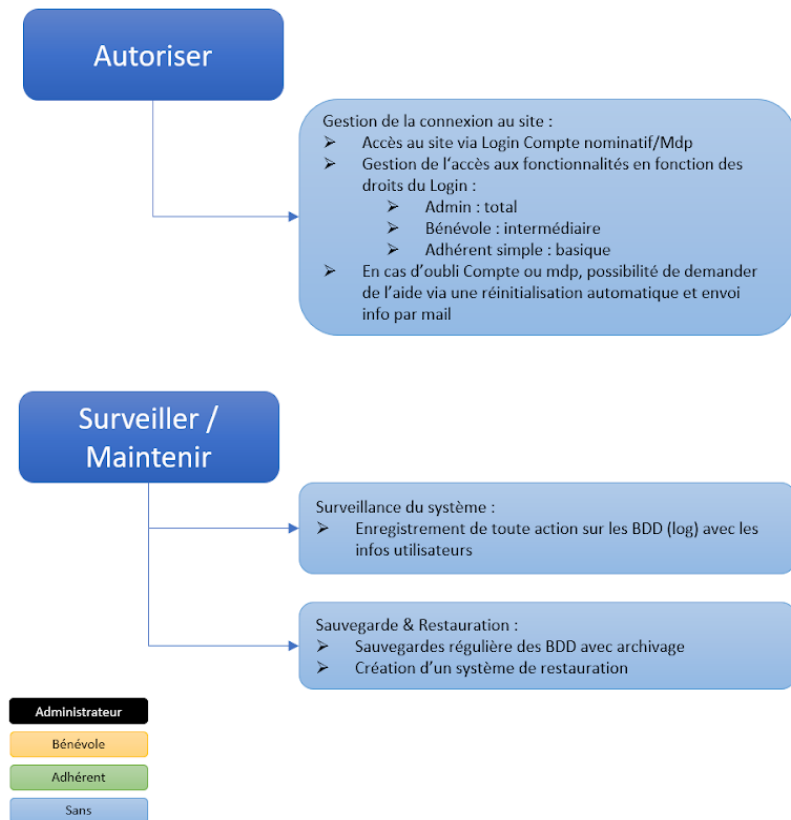


Figure 1: CDC : Autoriser, Surveiller/Maintenir

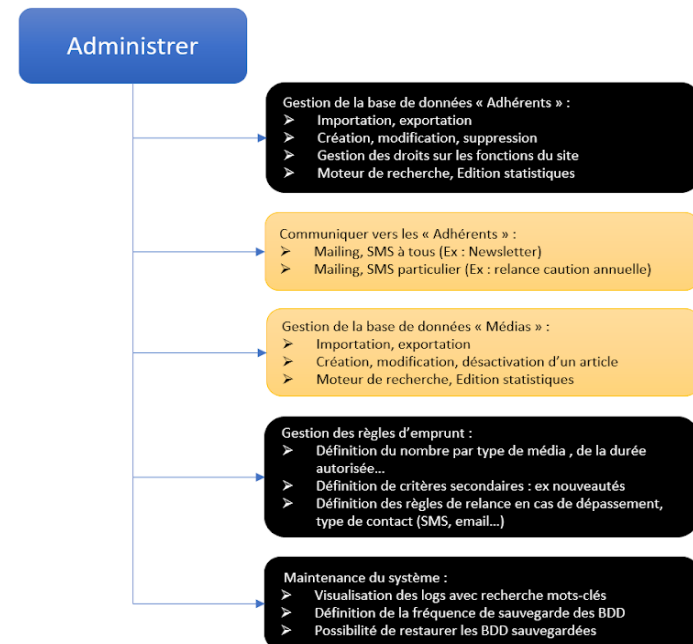


Figure 2: CDC : Administrer

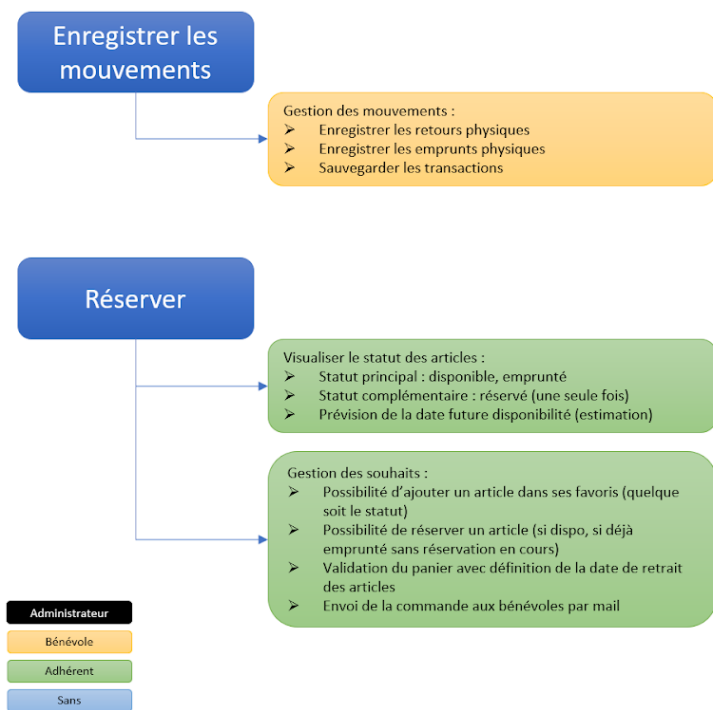


Figure 3: CDC : Enregistrer les mouvements, Réserver

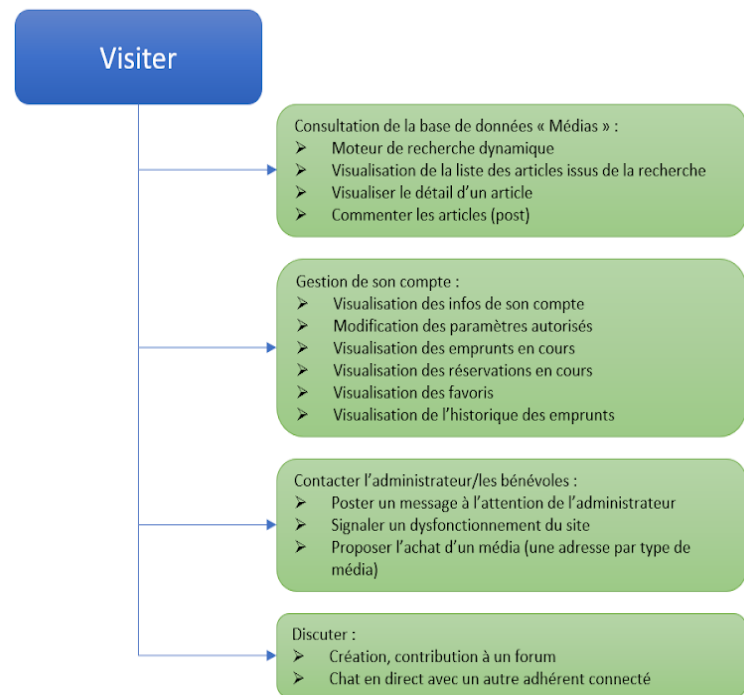


Figure 4: CDC : Visiter

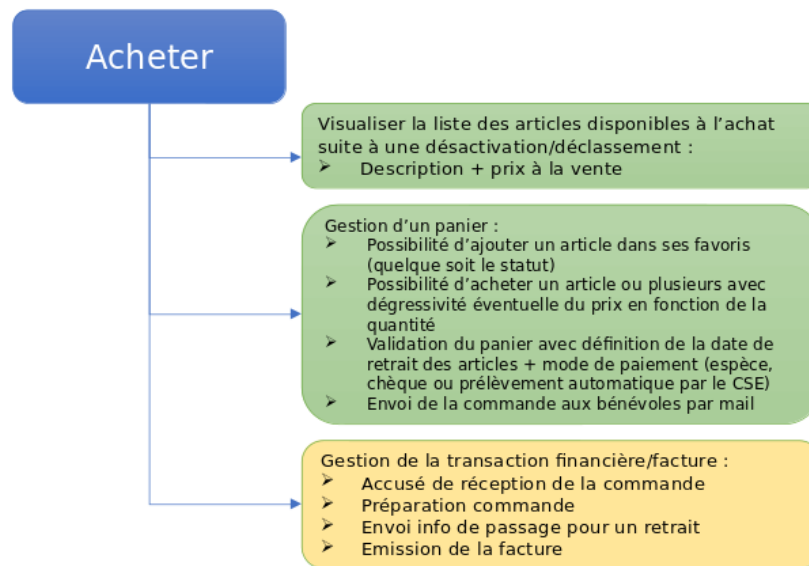


Figure 5: CDC : Acheter

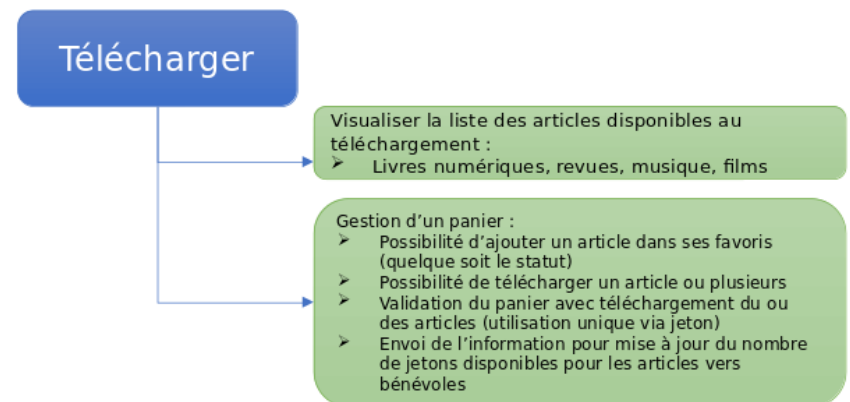


Figure 6: CDC : Télécharger

Suivi des étudiants

Semaine 15 (du 09/01 au 16/01) :

étudiant 1 : ajout des commentaires

étudiant 2 : controllerLivres => controllerArticle + ajout des livres dans la bdd

étudiant 3 : correction page profil + changement photo de profil

étudiant 4 : Annonces + crud automatique

Semaine 16 (du 16/01 au 23/01) :

étudiant 1 : page statistique

étudiant 2 : apiMusique + renouvellement apiLivres + detail ApiVideo

étudiant 3 : page panier + emprunt

étudiant 4 : crud article complet

Semaine 17 (du 23/01 au 30/01) :

étudiant 1 : page show et détail article => correction de certains bugs

étudiant 2 : recherche sur les ApiMusique , début de tri des APIs livres pour pouvoir intégrer les données dans la bdd actuelle

étudiant 3 : page panier + gestion des emprunts et début des commandes

étudiant 4 : correction de bugs du crud article et design des autres crud

Semaine 18 (du 30/01 au 06/02) :

étudiant 1 : finition du filtre dans la page des articles

étudiant 2 : récolte des images, éditeurs, auteurs, date de publications, titre des APIs Livres

étudiant 3 : gestion des commandes

étudiant 4 : début de modification du crud article

Semaine 19 (du 06/02 au 13/02) :

étudiant 1 : commencement du responsive du site

étudiant 2 : continue le tri et les transformations de date des données des APIs livres

étudiant 3 : correction bug gestion des emprunts et des commandes

étudiant 4 : continuation de la modification d'un article pour modifier toutes les tables

Figure 7: Suivi des étudiants sur Github

Charte graphique

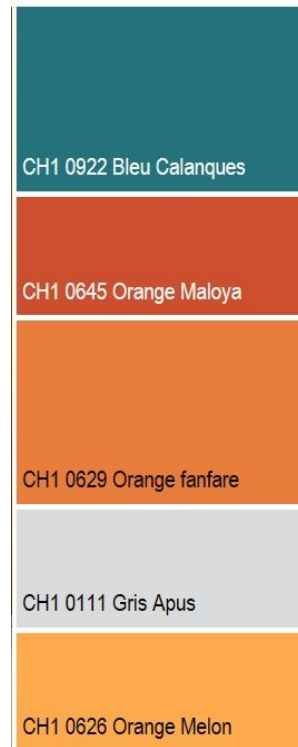


Figure 8:
Palette de couleurs

Tableau de questions réponses

A	B	C
	ERD	logigramme
développeur	26/11 : à quoi correspond les commentaires 1 et 2 dans la tables utilisateurs?	demande de validation de la page accueil?
		26/11 : ok, j'ai changer
		26/11 : je te redemande confirmation pour la page d'accueil? Comme ca on y touche plus et cela nous servira pour notre prochain rapport. J'aimerais aussi avoir les premiers retours sur le logigramme de la page en détail d'un article et de la page de la liste des articles, profil, panier et favoris aussi (dans le weekend ce serait cool). Dans la page profil, on pense qu'il faut regrouper réservations et emprunts sur la meme page car sinon les pages seront relativement vide, il suffira de faire 2 tableaux différents dans la page (1 réservation et 1 emprunt). Pour les historiques, est-ce que tu veux qu'on ai la possibilité de supprimer des réservations, emprunts, commandes et de vider les historiques? Je pense qu'il faut également revoir la partie commentaire, avec les commentaires malveillants aussi, je ne sais plus ce qu'on aviat décider par rapport à ca. Aussi, savoir si c'est sur une page a part que l'on écrit le commentaire ou sur la meme page que l'article, en dessous, aussi, je ne sais pas si c'est utile, mais je me suis posé la question si il ne fallait pas avoir une page avec l'historique de ses commentaires pour chaque article, dis moi ce que t'en pense. Faut aussi bien définir les différents status pour les articles, les emprunts. Le logigramme s'adaptera au schéma ERD qu'on t'enverra ce weekend, donc ce qui est sur le logigramme est provisoire. Il est possible que j'ai oublier certains trucs mais dans certains pages, si tu as d'autres fonctionnalités à ajouter, n'hésite pas! Pour la page profil, il faut revoir les différentes pages de réservation, achats en cours par rapport au panier!
		4/12 : ok pour page accueil, et ok, j'ai modifié pour la page des articles, on peusent que concernant le nombre limite, on le fait drectement lorsque l'on clique sur ajouter au panier, pareil pour les droits achats, droit emprunt, en fonction du statut (dispo, vendable, empruntable...), les boutons seront visibles ou non. Coconerant l'histoire d'avoir visuellement la liste du panier et des articles que l'on a, c'est pas trop possible mais je t'ai fait une

Figure 9: Exemple de question

A	B	C
client		<p>29/11 : Page profil/compte :</p> <p>Par défaut, il faut afficher quelques paramètres mais pas tous (matricule/Nom/Prénom/Identifiant/rôle/droitemprunt/droitAchat. Le reste des paramètres est accessible en allant dans Modifier les paramètres. Il faut aussi afficher par défaut ce qui est en cours dans des cadres avec scrollbar pour la détail (Commande/Réservation en cours, Commande/HA en cours, Emprunt en cours) avec surtout le statut des commandes en cours. Il faut surtout afficher en gros les dates qu'il faut se rappeler : date de retour des articles dans l'ordre plus près à plus loin. Du coup, cela me fait penser que ce serait bien au login d'afficher un popup en même temps qu'on arrive sur le site pour indiquer si il y a des articles qui sont en retard à rendre et le laisser indiquer sur la page principale de manière constante (en rouge près de la barre recherche par exemple).</p> <p>Via les cadres des commandes/emprunt en cours, on peut zoomer pour avoir la totalité de la commande/emprunt à l'écran.</p> <p>Ensuite, on peut accéder par des clics à "Modifier paramètres du compte" et "historique des activités"</p> <p>Pour "modifier param...", on affiche tous les champs de la table Utilisateur sauf id, motdepasse, commentaireadmin, datecreation & datemodification. Ceux accessibles à la modif directement sont telperso, telperso2, emailperso, telpro, telpro2, emailpro, adresserue, adresseruecomplement, ville, codepostal, commentaireuser et vignetteavatar. Pour les notifications, on peut les bouger mais il faut au minimum 1 qui soit activée. Les infos suivantes sont affichées mais non modifiables : matricule, nom, prénom, sexe, droitemprunt, droitachat, libelleentreprise, role et libellefonction. Pour la modification de l'identifiant, motdepasse et emailrecup, il faut faire une action pour précéder à de smodifs et on peut imaginer récupérer la mécanique de chgt mdp de la page identification. Pour le chgt identifiant, pas de souci particulier. Pour le chgt mdp, il faut demander l'ancien puis le nouveau 2 fois. Pour l'email recup, pas de souci particulier mais il faut vérifier que l'adresse est valide tout comme les adresses perso et pro si modif. Dans tous les cas, il faut envoyer un recap des modifs à l'ancienne adressenotifiée et la nouvelle adresse s'il y a des chgts. C'est pour se prémunir d'un éventuel piratage.</p> <p>Possibilité de les sauvegarder par un bouton et si on quitte le menu sans avoir sauvegarder, on l'indique et on précise que les modifs seront perdues si on continue.</p> <p>Pour "historique activités...", on affiche sous forme de liste avec possibilité de trier HA/Emprunt, par date, par typearticle (si plusieurs type article par commande, on valide). On pourrait ajouter une notion de statistiques à titre personnel : somme des achats en fonction des filtres et somme des articles empruntés par type et par intervalle de dates. Cela me semble déjà pas mal !</p>
		<p>"29/11 : Page Affichage Liste articles :</p> <p>Globalement, le logigramme est OK.</p> <p>Pour le filtrage, je vois le filtre TrancheAge à rajouter. Il faudra pouvoir en rajouter des plus fins : on pourrait imaginer un accès à des filtres supplémentaires et ensuite on choisit par</p>

Figure 10: Exemple de réponse

Travail réalisé

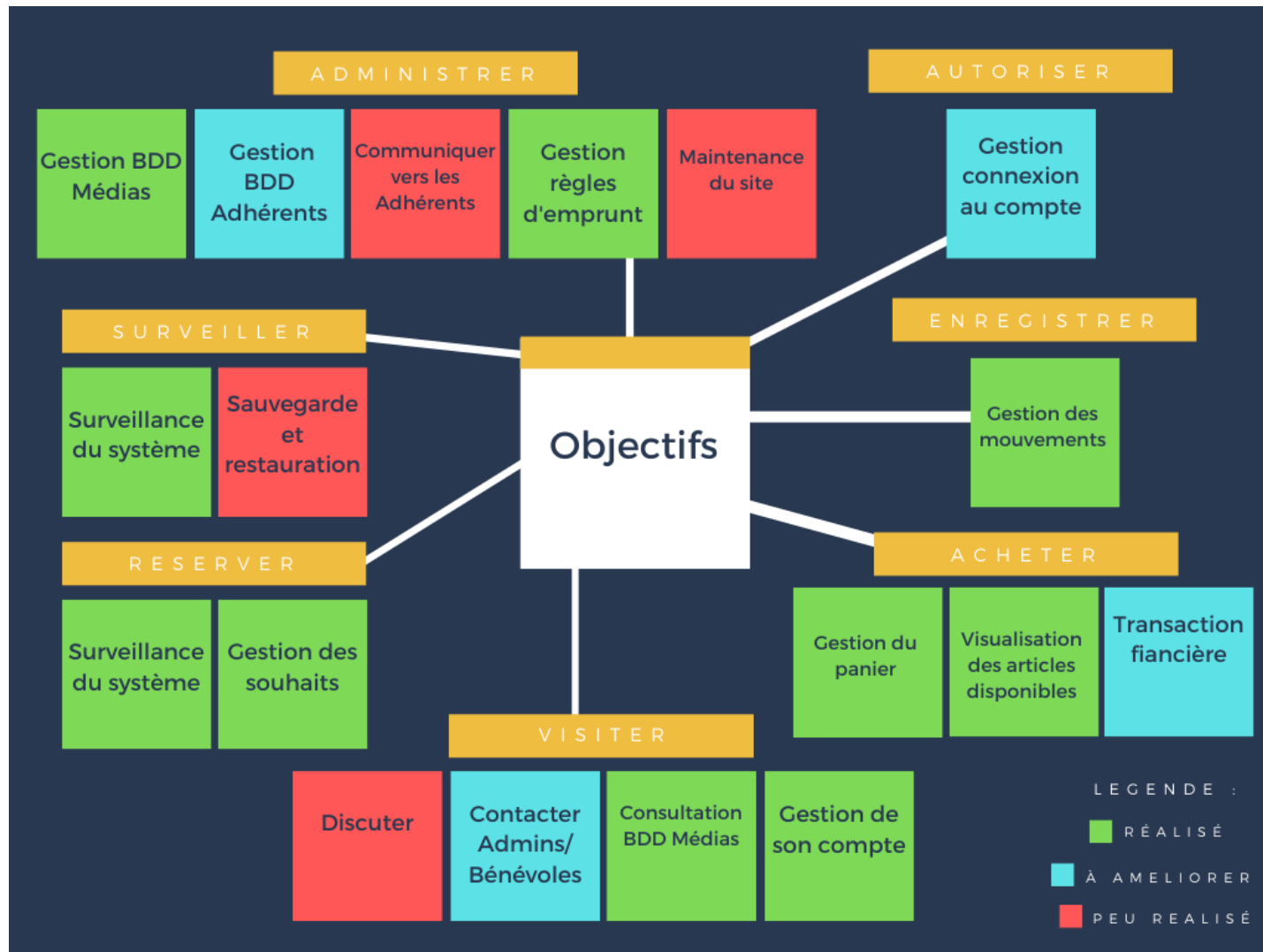


Figure 11: Schéma présentant les objectifs réalisé, à améliorer et peu réalisé