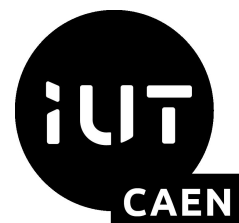




UNIVERSITÉ
CAEN
NORMANDIE



Université de Caen Normandie
Institut Universitaire de Technologie de Caen
Département Informatique

Diplôme Universitaire de Technologie
Informatique

Réalisation d'un site web de prêt d'instruments de musique

Mémoire de conduite de projet de 2^e année

Valentin BOSSARD
Alexis HESLOUIN
Quentin LEMOIGNE
Alexis MELO DA SILVA

Commanditaire : Marc LEVILLY
Tuteurs du projet : Fabienne JORT et Marc LEVILLY
Jury : Stéphane SECOUARD et Albrecht ZIMMERMANN

Année universitaire 2018 / 2019

Ceci est un mémoire de conduite de projet retraçant la réalisation d'un site web permettant la gestion du prêt et de l'emprunt d'instruments de musique entre musiciens, initialement prévu pour ceux en déplacement à l'étranger.

Le site doit être en anglais car son objectif est de mettre en relation les musiciens du monde entier. Il fonctionne sur un système qui permet aux musiciens d'ajouter à leur profil les instruments qu'il possède et de leur attribuer des plannings de disponibilités pour les emprunts.

Le site doit avoir un aspect communautaire fort pour susciter de la confiance, en effet les prêts sont gratuits et le public cible est réduit.

Mots-clés : site, prêt, emprunt, instrument, musicien

This is a dissertation about the conception of a website allowing musicians to handle the lending and borrowing of their instruments, initially conceived for the ones going abroad.

The website has to be in english because it aims to put in contact musicians throughout the world. It works with a system allowing musicians to add instruments to their profile and define a planning on which they are willing to share each of their instrument.

The website has to be strongly community oriented in order to arouse confidence between the users, indeed all the loans are free of charge and the number of users is expected to be very few.

Key-words : website, borrowing, instrument, musician, loan

Réalisation d'un site web de prêt d'instruments de musique

Remerciements

Nous tenons à remercier nos tuteurs Marc LEVILLY et Fabienne JORT de nous avoir accompagné et conseillé pendant le projet.

Nous souhaitons témoigner notre gratitude à Eric PORCQ pour avoir corrigé notre modèle de base de données, à Philippe BRUTUS pour ses conseils sur les diagrammes d'UML ainsi qu'à Jean-Christophe HERON pour ses cours sur les framework, et Jean-François ANNE pour avoir répondu à nos questions sur les CMS.

Remerciements également à Axel MASSERON pour sa proposition d'un modèle économique permettant au site de gérer les transactions d'argent, et ses réponses à nos questions concernant les cautions.

Merci aux étudiants de première année Quentin CARRE, Théa LINOT, Mathilde GORE et Nathan BONNEMAINS ayant réalisé le poster de notre projet. Cela a non seulement permis d'exposer le fruit de notre travail mais ils ont aussi eu la gentillesse de proposer un logo pour notre site. Merci aux autres étudiants de première année Jeremy L'ECUYER, Loïc LE GOAS, Luc AUDABRAM et Matthis RIVAT qui ont repris la conduite de notre projet pour le terminer.

Sans oublier Christelle PASSONI-CHEVALIER pour le document qui nous a permis de rédiger ce mémoire et Jean-Luc LAMBERT pour ses cours de gestion de projet agile.

Nous tenions aussi à remercier Marc LEVILLY pour nous avoir proposé un projet aussi intéressant.

Table des matières

Table des sigles et des abréviations	7
SQL : Structured Query Language	7
Glossaire	8
Introduction	9
Présentation du projet	10
Un site de gestion de prêt d'instruments de musique	10
Charte du projet	10
Contexte et identification des besoins	10
Enjeux	11
Objectif	11
Carnet de commande	11
Jalons du projet	13
Comment répondre aux exigences du projet	13
L'aspect communautaire du site	13
L'aspect technique du site et de ses fonctionnalités	14
Le modèle économique du site	15
Les outils utilisés	16
Outils envisagés durant la première année	16
Frameworks et langages finalement choisis pour le projet	18
Logiciels et matériels utilisés	19
Réalisation du projet	20
Les réalisations en conception	20
Diagramme de cas d'utilisation	20
Modèle physique et logique	24
Réalisations concrètes pour répondre aux exigences	25
Réponse au problème de gestion des instruments	25
Comment gérer les instruments que l'on veut prêter	25
Comment leur définir des plannings de disponibilité	27
Réponse au problème des réservations	29
Comment réserver un instrument	29
Comment répondre aux réservations	32
Réponse au problème du panel d'administration	35
Première solution envisagée en dur	35
Premier essai concret avec Easy Admin	35
Deuxième essai concret avec Sonata Admin	36

Réponse au problème du moteur de recherche	38
Réponse au problème de gestion des cautions	39
Avancée du projet aux différents lots	39
Lot 1	39
Lot 2	40
Lot 3	40
Rendu final	40
Avancée de notre application au dernier lot	40
Rétrospective et analyse des problèmes rencontrés	41
Problèmes techniques durant notre développement	41
Problèmes à long terme et réutilisabilité	41
Choses à développer pour le futur du site	42
Aspect communautaire à développer	42
Fonctionnalités à implémenter	42
Reprise du projet par les étudiants de première année	42
Conclusion	44

Table des sigles et des abréviations

1A : étudiant de première année

2A : étudiant de deuxième année

API : Application Programming Interface

CMS : Content Management System

CSS : Cascading Style Sheets

HTML : HyperText Markup Language

IHM : Interaction Homme Machine

IUT : Institut Universitaire de Technologie

MCD : Modèle Conceptuel des Données

MLD : Modèle Logique des Données

MPI : Modèle Physique d'Interaction

MVC : Model View Controller

ORM : Object Relational Mapping

PHP : Hypertext Preprocessor

POO : Programmation orientée objet

SEF/SEP : Schéma d'Enchaînement des Fenêtres / Schéma d'Enchaînement des Pages

SGBD : Système de Gestion de Base de Données

SNI : Schéma Navigationnel d'Interaction

SQL : Structured Query Language

Glossaire

Framework : Un framework est un ensemble cohérent de composants logiciels

Bundle : Un bundle est similaire à un plugin dans un logiciel mais pour le framework Symfony

Entité : Une entité correspond à une table au niveau de la base de données

Repository : Classe permettant d'effectuer des requêtes sur la base de données

Contrôleur : Classe contenant la logique du programme

Introduction

Le commanditaire de notre projet est Marc Levilly, enseignant à l'IUT de l'académie de Caen au département informatique. Dans le cadre de notre projet, il ne représente aucune organisation en particulier et ce n'est pas au professeur que nous devons ce projet, mais bien à monsieur Levilly en tant que passionné de musique et représentant de la communauté musicale.

L'intérêt du projet est de créer un service existant dans de nombreux domaines mais pas celui de la musique : un site permettant aux musiciens souhaitant emprunter et prêter des instruments de musique de se mettre en relation et de gérer leurs emprunts. Le commanditaire nous a fortement conseillé de nous inspirer de l'esprit du site ¹*couchsurfing* tant pour les fonctionnalités, les valeurs transmises par le site mais aussi pour son design.

La problématique est la suivante : comment réaliser un site web pour mettre en relation des musiciens afin de leur faciliter la gestion du prêt de leurs instruments de musique ? Sachant qu'il doit être simple, intuitif et interculturel car les utilisateurs viennent du monde entier, ont des habitudes et des connaissances différentes, mais que la communauté sera réduite car centrée autour d'une passion commune qu'est la musique.

Nous allons donc vous présenter, à travers ce mémoire, comment nous avons pu tenter d'apporter une solution au projet demandé en présentant d'abord le projet dans sa globalité au niveau conceptuel, puis au niveau physique en détaillant nos réalisations et finalement la reprise du projet et les extensions futures.

¹*couchsurfing* (www.couchsurfing.com) : *couchsurfing* est un site web permettant aux gens en voyage à l'étranger de demander à être logé chez l'habitant. Le site a un aspect communautaire car il encourage les rencontres en utilisateurs, et le design du site est épuré et intuitif.

1. Présentation du projet

Dans cette première, partie nous allons identifier les différents aspects conceptuels du projet afin de mieux identifier les besoins du client et des potentiels utilisateurs du site. Nous allons également présenter ce que nous avons imaginé pour répondre aux différentes exigences pour la réalisation du site, notamment en parlant des choix technologiques précis et des choix conceptuels pour réaliser notre site.

1.1. Un site de gestion de prêt d'instruments de musique

1.1.1. Charte du projet

1.1.1.1. Contexte et identification des besoins

Notre site doit répondre à plusieurs problématiques posées régulièrement dans la vie des musiciens passionnés et néophytes. L'idée originale est de répondre à un besoin bien précis qui est de permettre aux musiciens en voyage à l'étranger de trouver des instruments directement sur place grâce à un système de prêt entre particuliers. Le problème aussi précis viens de l'idée que les instruments sont compliqués à transporter dans les transports en commun (bus, avions, ...) et qu'il y a le risque de les dégrader lors du voyage ou de leur stockage dans les soutes.

Cependant, le site une fois mis en ligne pourra sûrement être utilisé par des gens avec des besoins différents. En effet, les instruments de musique ont trois grandes caractéristiques qui peuvent faire varier les besoins. D'abord le type d'instrument : un musicien peut vouloir découvrir de nouveaux instruments, jouer d'un instrument légèrement différent de celui qu'il possède déjà ou qu'il a l'habitude de jouer (exemple : marque différente, version plus petite, plus grande...) sans pour autant racheter un instrument neuf. Le second facteur est le prix des instruments : les musiciens n'ont pas forcément les moyens d'acheter ce qu'ils veulent utiliser. Finalement il y a aussi la place qu'occupe un instrument, en dehors des moyens financiers, certaines personnes n'ont tout simplement pas la place d'avoir certains instruments volumineux chez eux, ou bien ne peuvent pas se déplacer avec.

Le commanditaire a insisté sur un site avec un aspect communautaire assez présent car même si les scénarios pour lesquels une personne peut être amenée à utiliser ce site, la communauté globale sera assez réduite car spécialisée. Le site doit encourager l'utilisateur à rester et le fidéliser car non seulement les prêts seront gratuits, mais il y aura au final un contact réel entre les utilisateurs.

1.1.1.2. Enjeux

Le site doit répondre à plusieurs enjeux. Nous devons produire un site web novateur et viable, pouvant être utilisé sur le plus de supports possibles. Il doit permettre la rencontre entre musiciens ainsi que leur faciliter la gestion du prêt de leurs instruments. Le site doit avoir un aspect communautaire, il doit alors être intuitif, simple et interculturel pour être utilisable à l'international.

1.1.1.3. Objectif

L'objectif du projet est donc de réaliser un site web pour la gestion du prêt d'instruments de musique entre musiciens. Il doit être assez complet et performant pour être utilisé par des milliers d'utilisateurs venus de partout dans le monde. Le site final doit comporter plusieurs pages pour que les utilisateurs puissent gérer leurs instruments, rechercher des instruments à emprunter, gérer leurs réservations et communiquer entre eux. L'objectif est donc de produire un site web non seulement simple d'utilisation pour les utilisateurs finaux qui sont les musiciens, mais aussi réutilisable pour les futurs développeurs et administrateurs qui devront développer des fonctionnalités sur le site et le maintenir en vie.

1.1.2. Carnet de commande

Voici le backlog prévu pour le projet :

Numéro	Description	Lot
1	En tant que visiteur je peux m'inscrire sur le site	1
2	En tant qu'utilisateur je peux me connecter	1
3	En tant qu'utilisateur je peux enregistrer un instrument sur mon profil et lui ajouter une description	2
4	En tant qu'administrateur je peux définir des catégories d'instruments	2
5	En tant qu'utilisateur je peux rechercher des annonces sans contraintes	2
6	En tant que visiteur je peux visionner les annonces des instruments avec une visibilité réduite	2
7	En tant que visiteur je peux rechercher des instruments avec des droits réduits	2
8	En tant qu'utilisateur je peux définir le montant d'une caution	2
9	En tant que modérateur je peux vérifier les villes ajoutées par les utilisateurs	2
10	En tant que modérateur je peux gérer les villes ajoutées par les utilisateurs	2
11	En tant qu'utilisateur, je peux signaler une annonce	3
12	En tant que prêteur, je peux définir une localisation pour un instrument	3
13	En tant que prêteur, je peux éditer un instrument	3

14	En tant qu'utilisateur, je peux rechercher des annonces par catégorie/marque/nom/date	3
15	En tant qu'utilisateur, je peux rechercher des annonces par Localisation	3
16	En tant qu'utilisateur, je peux trier ma recherche d'annonces par localisation	3
17	En tant que prêteur, je peux définir un planning de disponibilités pour un instrument	3
18	En tant qu'utilisateur, je peux voir les disponibilités d'un instrument en fonction du jour et de l'horaire	3
19	En tant qu'utilisateur, je peux montrer/cacher un instrument pour autoriser ou non l'emprunt	3
20	En tant qu'utilisateur je peux emprunter un instrument	3
21	En tant que modérateur je peux bannir un utilisateur	3
22	En tant qu'administrateur je peux gérer les rangs	3
23	En tant que prêteur, je peux voir les demandes de d'emprunt	3
24	En tant que prêteur, je peux accepter ou refuser une demande de réservation	3
25	En tant qu'utilisateur je peux visionner le profil d'un autre utilisateur	3
26	En tant qu'utilisateur je peux afficher le numéro de téléphone de l'annonce	3
27	En tant qu'utilisateur je peux utiliser le tchat	3
28	En tant que modérateur je peux gérer les signalements	3
29	En tant qu'utilisateur je peux envoyer un message privé à un autre utilisateur	3
30	En tant qu'utilisateur, je peux noter le prêteur/emprunteur	3
31	En tant que modérateur je peux gérer les notes/commentaires d'utilisateurs	4
32	En tant qu'utilisateur je peux visionner l'historique des emprunts/locations	4
33	Ajouter un système de crédit	4
34	En tant que visiteur je peux entrer sur le site par carte	4
35	En tant qu'utilisateur je peux choisir mon moyen de paiement de caution	4
36	En tant que modérateur je peux surveiller les prix (cautions)	4
37	Rédiger une charte de dédommagement en cas de vol/conditions d'utilisation	4
38	En tant qu'utilisateur je peux poster une annonce de DEMANDE d'instrument	4

1.1.3. Jalons du projet

- ❖ Fin Janvier : Attribution du projet
- ❖ 09/02/2018 : Première rencontre avec le commanditaire
- ❖ 23/02/2018 : livraison de la première version du MCD + livraison de la 1ère version du diagramme de cas
- ❖ 26/02/2018 : Rencontre avec M.Porcq pour valider le MCD
- ❖ 15/02/2018 : Rédaction du cahier des charges
- ❖ 01/03/2018 : Rédaction de la charte de projet
- ❖ 28/03/2018 : Première réalisation d'une page sur Marvel App + modification du diagramme de cas
- ❖ 12/04/2018 : remplissage et test de la BDD + backlog + 2e version du diagramme de cas d'utilisation
- ❖ 30/04/2018 : livraison de la page principale (d'accueil)
- ❖ 12/05/2018 : étude comparative cms
- ❖ 24/05/2018 : réalisation du SNI
- ❖ 03/06/2018 : nouvelle réalisation d'une page sur Marvelapp
- ❖ 01/06/2018 : réalisation du SEF et du SEP
- ❖ 11/06/2018 : Rendu du lot 1 et soutenance
- ❖ 11/10/2018 : Rendez-vous avec les tuteurs pour montrer l'avancée réalisée pendant les vacances
- ❖ 14/11/2018 : Rendu du lot 2 et planification des objectifs pour le lot 3
- ❖ 14/01/2019 : rendu du lot 3 et soutenances individuelles sur le projet
- ❖ 17/01/2019 : rendez-vous avec les tuteurs pour déterminer ce qu'il y aura à rendre au dernier lot
- ❖ 01/04/2019 : rendu du lot 4 et soutenance du projet

1.2. Comment répondre aux exigences du projet

1.2.1. L'aspect communautaire du site

L'un des aspects du site qui tenait à coeur au commanditaire est d'avoir un côté social et communautaire qui se développe grâce aux rencontres faites entre les musiciens. Comme la communauté est réduite, la réputation du site et son cercle d'utilisateurs se développera surtout grâce au bouche à oreille.

Pour insister sur cela nous avons dû faire plusieurs choix au niveau graphique pour rendre le site attirant et inciter les gens à rester. Au niveau de la charte graphique rien ne nous a été indiqué par le commanditaire donc nous avons fait nos propres choix, en essayant de faire le mieux possible pour avoir un rendu visuel acceptable pour une première version du site. Les utilisateurs seront incités

à rester grâce aux couleurs (rouge sur la page d'accueil par exemple, une couleur chaude), visuels et aussi les différentes fonctionnalités propres au site.

Au niveau des visuels nous avons eu la chance d'avoir un groupe d'étudiants en première année qui a réalisé un logo pour notre site, nous l'avons accepté pour l'instant.

Comme les utilisateurs peuvent potentiellement venir du monde entier, il faut réussir à rendre le site multiculturel pour qu'il soit au maximum compréhensible par tous. Il est donc logique de le faire entièrement en anglais qui est la langue la plus couramment enseignée et parlée dans le monde. Pour faciliter la vie des utilisateurs, nous réduisons au maximum leur saisie au clavier, car chaque saisie sera faite dans une langue différente et au niveau de la recherche d'instruments cela rendra la chose très compliquée.

Finalement comme tous les prêts seront gratuits, il faut réussir à susciter la confiance le plus possible entre les utilisateurs à travers divers moyens. Pour réussir à faire cela nous avons proposé d'implémenter un système de messagerie entre les utilisateurs, ce qui constitue la base évidente. Il y a aussi le système de notation, de commentaires, de vision des profils. Nous avons aussi choisi d'ajouter un système permettant de définir une caution pour un instrument donné. Ce concept a évolué le long du projet et est détaillé dans la partie 1.2.3.

1.2.2. L'aspect technique du site et de ses fonctionnalités

Pour répondre aux exigences nous avons dû réfléchir à l'aspect technique de ce que nous allons mettre en place. Chacun de nos choix est guidé par le contexte de notre projet qui rend obligatoire d'avoir un site multiculturel au maximum et qui soit simple d'utilisation.

D'abord au niveau du système d'annonces, il a été présenté par le commanditaire à la manière du système d'annonces du site [Leboncoin.fr](https://www.leboncoin.fr), mais après réflexion il est trop complexe et ne permet pas d'avoir un site agréable à utiliser dans le cas où les musiciens vont devoir poster leurs annonces à chaque fois. Nous avons alors proposé la solution suivante : chaque musicien utilisateur du site possède un espace personnel où il peut enregistrer les instruments qu'il possède et qu'il souhaite prêter. Au lieu de recréer une annonce à chaque fois, il lui suffit d'ajouter un planning correspondant à une période où l'instrument sera disponible à la réservation.

Au niveau de ces plannings de disponibilités, il fallait faire un choix au niveau de l'affichage des jours et des horaires sachant que tout est en anglais. Nous avons

décidé que ce qui permettrait au plus de gens d'utiliser le site serait de mettre les dates au format 24h. Sur recommandation du commanditaire, les horaires sont limités à des heures entières car un découpage à la minute ou à la demi-heure serait trop précis sachant que chaque prêt requiert un déplacement, il faut une marge d'erreur. Pour mettre en place l'ajout de planning nous avons dû les différencier en deux types : date ponctuelle et récurrence sur plusieurs jours, nous reviendrons plus tard sur cet aspect.

Le moteur de recherche devait être rapide et comporter plusieurs éléments essentiels à une recherche. Nous en avons identifiés plusieurs comme essentiels : la localisation, la date et le type d'instrument sont les trois piliers. Cependant ils doivent être accompagnés d'autres critères, pour affiner nous avons ajouté la marque de l'instrument, un champ où l'utilisateur peut entrer ses propres mots-clés, et un choix à faire au niveau de la localisation : soit rechercher à un emplacement précis et dans ce cas-là les résultats seront triés par éloignement croissant du point cible, soit définir son propre rayon de recherche en fonction du point d'arrivée.

Au niveau des réservations, la question de l'acceptation automatique s'est posée. Comme le site doit mettre en confiance les utilisateurs, nous avons choisi de laisser le choix aux utilisateurs d'accepter ou non la réservation. Pour cela leur gestion se fait en trois pages distinctes : les demandes de réservation pour vos instruments, les réservations acceptées pour vos instruments, et les demandes que vous avez fait pour emprunter d'autres instruments.

1.2.3. Le modèle économique du site

Lors de nos premières réunions M. Levilly a évoqué la possibilité d'un système de caution à intégrer sur le site. La caution servira à dédommager le propriétaire de l'instrument en cas de dégâts ou de vol. Comme cela n'était pas une priorité, nous avons décidé d'implémenter ce système à la fin du projet s'il reste encore du temps.

À titre d'information, nous nous sommes entretenus avec le professeur de droit M. Axel Masseron pour en savoir un peu plus sur les mentions légales à adopter en cas d'implémentation de ce système sur notre site. Pour avoir quelque chose de sécurisé, il faudrait que l'argent soit stocké par nos soins et que les virements ne soient pas directement versés aux utilisateurs. On obtient alors un système avec trois intervenants : le prêteur, l'emprunteur et nous. Il nous a expliqué que si notre site veut être autonome il doit générer des revenus au moins suffisants pour payer l'hébergeur.

Pour cela il faudrait utiliser l'argent mis en "attente" pendant les cautions pour en tirer un bénéfice grâce aux intérêts bancaires. Il faudrait alors créer une entreprise et lui déléguer le site, pour qu'elle puisse aller auprès des banques et essayer de négocier un compte bancaire avec un fort intérêt et sur le court terme car les cautions ne restent pas longtemps. Il a insisté sur le fait qu'il faut garder les cautions au maximum sur notre compte, quitte à les rendre plusieurs jours ou semaines après que le prêt soit fini pour qu'on ait le temps de capitaliser dessus. Il faudrait également mettre en place un système de sécurité énorme au niveau de notre site pour gérer la "possession" de l'argent et les paiements en ligne, chose à laquelle nous ne sommes pas formés à l'IUT. Finalement, il faudrait payer un comptable pour qu'il garde la trace d'absolument toutes les cautions, celles en attente, celles déjà rendues, et qu'il fasse la comptabilité derrière le site.

1.3. Les outils utilisés

1.3.1. Outils envisagés durant la première année

Dans un premier temps, nous avons réfléchi à créer le site web directement en HTML, CSS, PHP et MySQL mais l'inconvénient était que pour un projet tel que celui-là, le temps de développement aurait été bien plus important qu'une autre méthode, de plus, il aurait été assez difficile de modifier le code une fois créé. De plus en codant de cette manière, le code aurait très certainement été peu structuré et plutôt difficile pour un autre membre du groupe de réutiliser ou de poursuivre ce qui aurait déjà été fait. Enfin, notre commanditaire M.Levilly nous a demandé un site potentiellement réutilisable et modifiable au niveau du code pour une possible reprise de projet. Cette contrainte a confirmé le choix de ne pas choisir cette méthode.

Sur conseils de nos professeurs et de notre client, nous nous sommes orientés vers l'utilisation des CMS. Un CMS est un logiciel permettant de créer du contenu web. Il permet de créer un site web sans avoir forcément de connaissances en programmation, grâce à des modules déjà existants que l'on peut intégrer.

Malgré cela, pour avoir un site complet et ayant un certain nombre de fonctionnalités, nous sommes dans l'obligation de créer des modules ou des extensions qui seront spécifiques au CMS. Ces outils utilisent pour beaucoup soit leur propre langage, soit du code qui ne sera tout simplement pas réutilisable dans d'autres choses. Voici une liste des CMS envisagés et leurs avantages et inconvénients d'après ce que nous avons testé.

	Avantages	Inconvénients
Wordpress	<ul style="list-style-type: none"> - Grande communauté - Installation simple - Une grande quantité de thèmes et d'extensions - Ergonomique - Mise à jour régulière 	<ul style="list-style-type: none"> - Trop simple d'utilisation et de mise en service pour un projet 1A/2A
Spip	<ul style="list-style-type: none"> - Installation simple 	<ul style="list-style-type: none"> - Besoin d'apprendre un nouveau langage "spip", celui-ci n'est utilisé que par spip - Petite communauté
Drupal	<ul style="list-style-type: none"> - Grande communauté - Sécurité renforcée - Flexible - Une grande quantité de thème et de module 	<ul style="list-style-type: none"> - Adaptation au CMS plus longue que pour les autres
Joomla	<ul style="list-style-type: none"> - Grande communauté - Grand nombre de thèmes et de modules 	<ul style="list-style-type: none"> - Temps d'apprentissage du fonctionnement - Assez gourmand en ressources - Nombreux modules et thèmes payants

Au final, après des heures d'essais de différents CMS, nous avons décidé d'arrêter d'utiliser un CMS car ceux-ci ne nous permettaient pas de développer les fonctionnalités tel que nous le souhaitions et nous ne pouvions pas non plus modifier la base de données à notre manière. Nous avons donc décidé de nous tourner vers un Framework.

1.3.2. Frameworks et langages finalement choisis pour le projet

Nous avons choisi le Framework Symfony, tout d'abord car sa documentation est très complète, il dispose d'une très grande communauté et il est très utilisé dans les entreprises du monde entier, ce qui nous permettait de développer des compétences pour notre future insertion professionnelle.

Symfony est un framework PHP utilisant le modèle MVC. Ce modèle permet de séparer l'affichage, les données à afficher de la base de données et la logique du programme. Un framework est un ensemble d'outils incluant une multitude de bibliothèques. Ces dernières permettent une POO, ce type de programmation rendra le code plus compréhensible pour tous les membres du groupe et pour les personnes extérieures au projet. De plus, en utilisant ce framework, nous pouvons réaliser le site d'annonces sans contraintes (hormis le cadre imposé par le Framework) contrairement au CMS qui limite fortement les possibilités de développement de nouvelles fonctionnalités.

L'utilisation d'un framework, d'après ce que l'on nous a enseigné et ce que l'on a pu constater, possède un inconvénient au début de sa mise en oeuvre. Ce dernier est son temps d'apprentissage (qui peut varier entre des semaines, voir des années de formation pour bien maîtriser un framework), mais une fois appris, c'est un investissement très rentable car on peut développer des fonctionnalités assez rapidement et beaucoup de framework possèdent des similarités.

De plus, le commanditaire ayant particulièrement insisté sur le fait d'avoir un site qui puisse être réutilisable au niveau du code pour une possible reprise de projet, le choix du framework qui encadre la structure de notre code et de la programmation orientée objet nous a semblé être un très bon choix.

Au niveau du design, nous avons choisi d'utiliser Bootstrap qui est un framework mais cette fois pour le HTML et le CSS. Notre choix a été guidé par le fait qu'il est utilisé par de nombreux bundle sous Symfony et parce que la documentation sur ce framework est très complète, il est d'ailleurs utilisé sur énormément de sites connus notamment Twitter qui en est le créateur.

Nous utilisons des langages s'exécutant au niveau du serveur, notamment le langage JavaScript, la bibliothèque JQuery développée elle-même en JavaScript ainsi que du Ajax pour envoyer des requêtes du client vers le serveur sans avoir à changer ou recharger la page courante.

Enfin, nous avons choisi d'utiliser le SGBD MySQL car il est libre, gratuit mais aussi car il est assez proche d'oracle au niveau du langage SQL.

1.3.3. Logiciels et matériels utilisés

Pour réaliser le MCD et le MLD, nous avons utilisé le logiciel Win'design, c'est un logiciel que nous avons utilisé avec Mme Mandin et qui répondait parfaitement à nos besoins pour la création du model de notre base de données.

Nous avons choisi d'utiliser l'IDE PHPStorm de l'entreprise [Jetbrains](https://www.jetbrains.com/phpstorm/) qui est un outil de développement pour le PHP. Il possède une multitude de fonctionnalités qui permettent d'améliorer notre productivité, dont : gérer tout le fonctionnement du git, accéder à notre base de données et exécuter des requêtes SQL directement depuis l'interface, il possède une console linux intégrée et une autocomplétion du code très avancée et disponible pour de nombreux langages et frameworks (autocomplétion du code Symfony et Bootstrap notamment). De plus ce logiciel est gratuit pour les étudiants de l'université. L'inconvénient de cet IDE est qu'il demande un très grand nombre de ressources : pour ceux dont le PC est un peu moins performant, nous avons utilisé l'IDE Atom.

Nous sommes restés comme lors de la première année avec le gestionnaire de versions git. En effet, il nous permet d'accéder au projet depuis n'importe quel PC, de pouvoir donner des versions aux fichiers et en cas de problème, revenir sur une version précédente, il nous permet également d'avoir une trace du travail fourni par chaque étudiant du projet.

Pour héberger la base de données, nous avons donc utilisé le site alwaysdata.net et nous hébergeons le site en local sur nos PC.

Dans cette première partie, nous avons présenté les différents aspects conceptuels du projet : il fallait alors créer un site web permettant la gestion du prêt d'instruments de musique et la mise en relation de musiciens, avec la principale contrainte qu'il doit être utilisable à l'international. Nous allons maintenant présenter notre démarche et les différentes étapes de réalisation du projet qui ont conduit à répondre à cette problématique.

2. Réalisation du projet

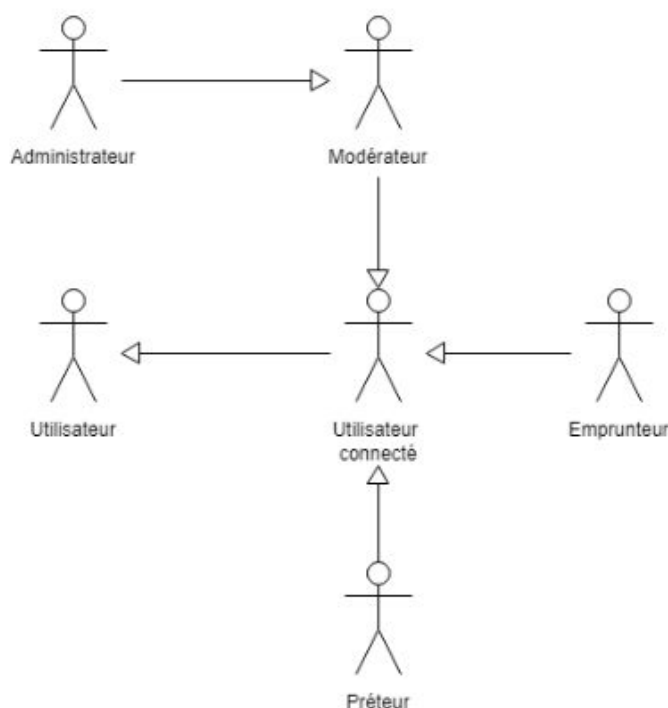
Dans cette deuxième partie nous allons montrer l'ensemble de nos réalisations en présentant les réalisations en conception puis nous allons expliciter les solutions concrètes que nous avons proposé afin de répondre aux exigences, pour finalement présenter l'avancement du projet aux différents lots.

2.1. Les réalisations en conception

2.1.1. Diagramme de cas d'utilisation

Nous avons commencé par la conception d'un diagramme de cas d'utilisation qui au début était très simplifié et qui s'est développé, en suivant les précisions de notre client, en un diagramme qui couvre la quasi-totalité des actions que peut réaliser un utilisateur du site en fonction de son rôle.

Diagramme de cas simplifié

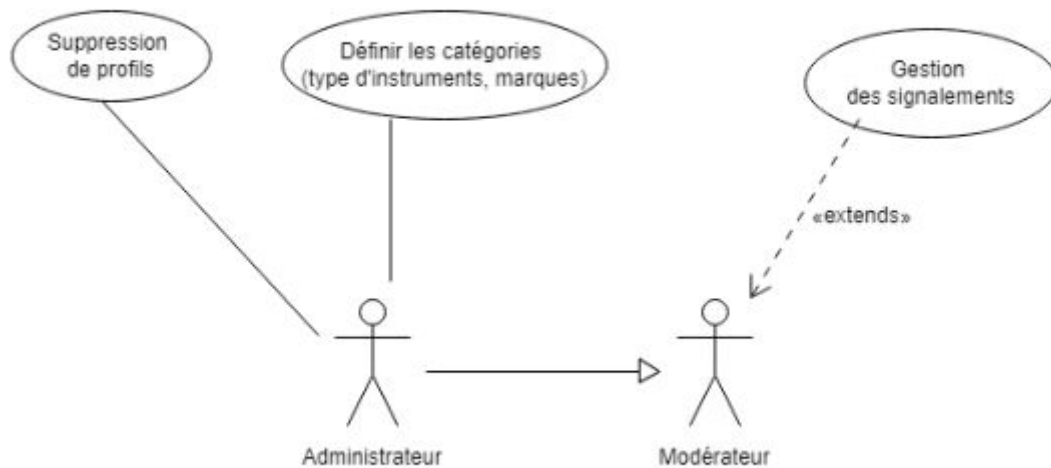


Ci-dessus le diagramme de cas d'utilisation avec seulement les différents rôles

Voici l'arborescence des utilisateurs de notre site. L'utilisateur "Utilisateur" est le rôle par défaut lors de l'arrivée sur le site. Les autres rôles sont des dérivés de l'utilisateur et possède les droits/rôles de tous les utilisateurs "au-dessus" d'eux par exemple l'administrateur possède les permissions du modérateur même si dans notre cas, le bundle d'administration n'étant pas terminé, le modérateur et l'administrateur ont le même rôle. Pour le moment il définit les catégories

(instruments, marques...) et il gère la suppression des profils, la gestion des signalements (voir image ci-dessous).

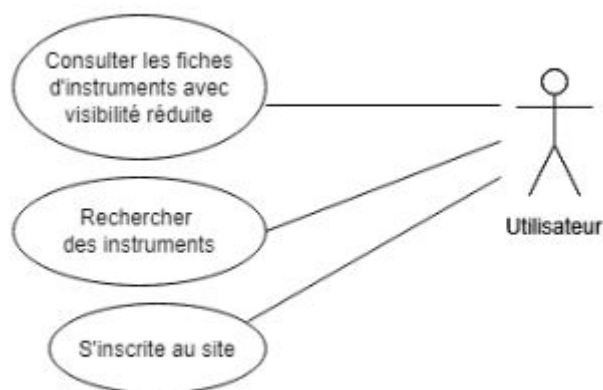
Diagramme de cas axé sur les administrateurs et modérateurs



Ci-dessus les différentes actions possibles pour un administrateur et un modérateur

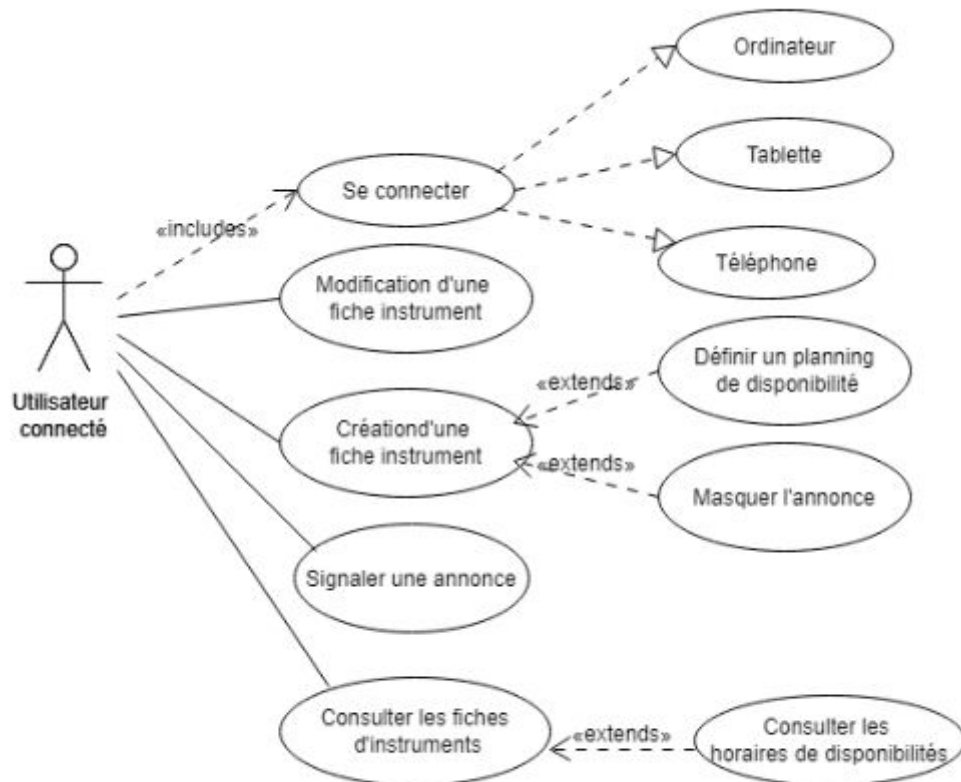
Comme vous pouvez le voir sur l'image ci-dessous, le visiteur dispose de permissions très limitées. L'inscription au site lui permet d'accéder à beaucoup de fonctionnalités.

Diagramme de cas d'un utilisateur (½)



Ci-dessus le diagramme de cas d'un utilisateur classique avec des responsabilités restreintes

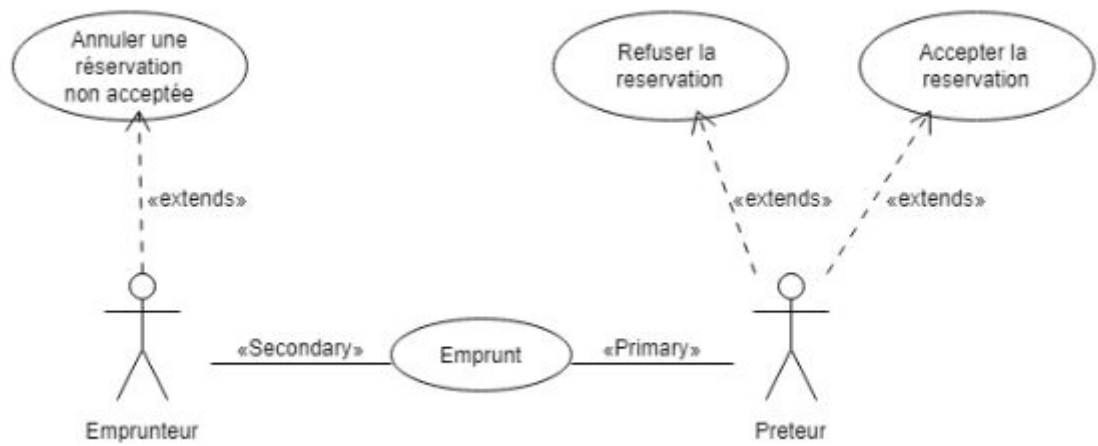
Diagramme de cas d'un utilisateur (½)



Ci-dessus le diagramme de cas d'un utilisateur connecté avec plus de possibilités qu'un utilisateur non-connecté

Accéder à l'ensemble des fonctionnalités du site implique de se connecter sur un ordinateur, une tablette ou un téléphone. L'utilisateur connecté peut accéder à un ensemble de fonctionnalité qui seront détaillé dans la deuxième partie du mémoire. Notamment à la gestion de réservation ci-dessous.

Diagramme de cas d'utilisateurs connectés



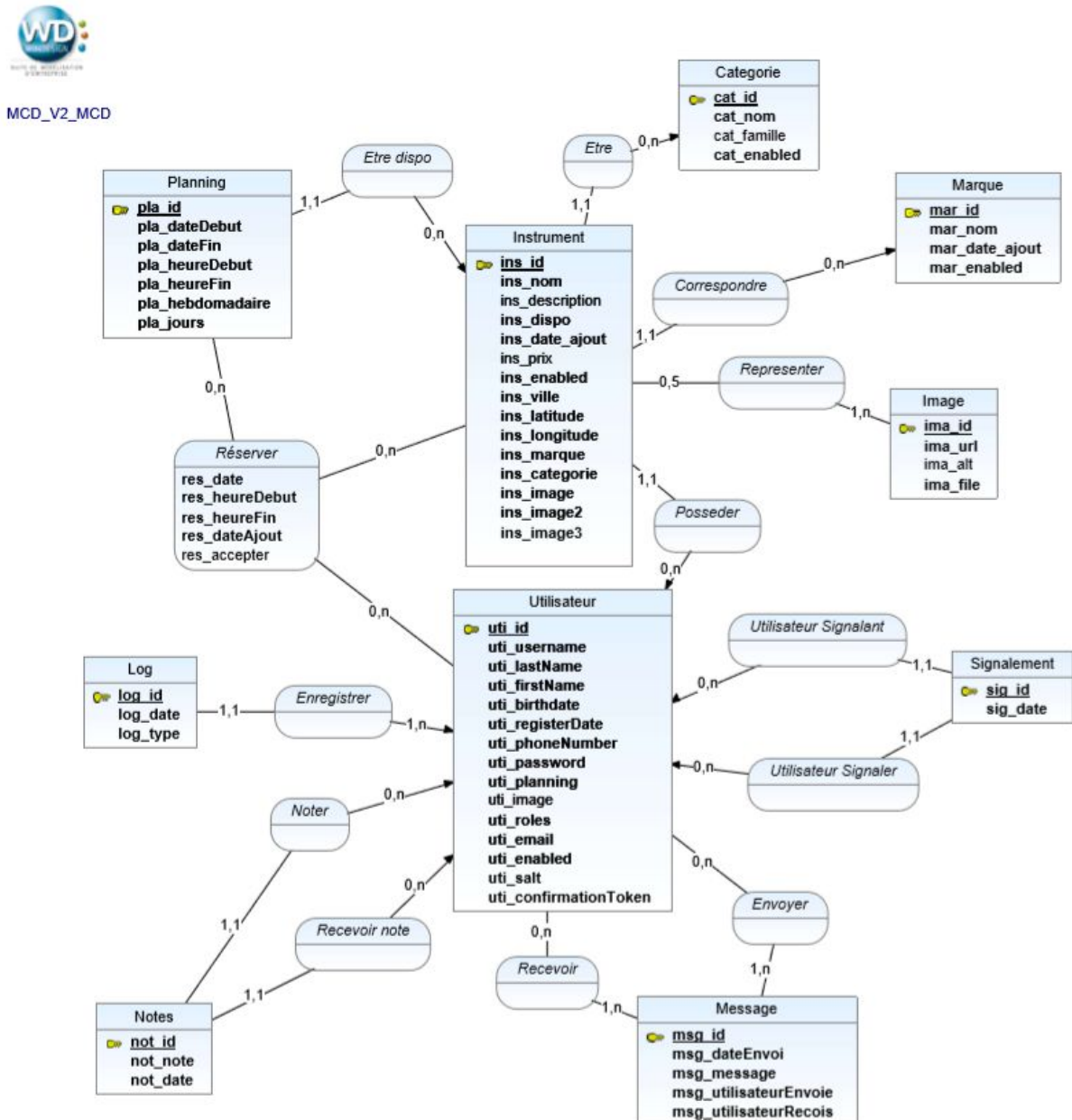
Ci-dessus le diagramme de cas entre un prêteur et un emprunteur

Une explication des différents symboles utilisés est disponible en annexe 1.

2.1.2. Modèle physique et logique

Une des premières étapes de notre projet a été de nous informer sur le projet en posant des questions à notre commanditaire sur le fonctionnement du site ainsi que sur les différentes fonctionnalités. Grâce à toutes ces informations, nous avons pu constituer le MCD de la base de données.

MCD De la base de données du site



Le MCD ci-dessus correspond à ce que représente la base de données à ce jour.

Le premier MCD (annexe II) à été vérifié par M.Porcq, depuis celui-ci a quelque peu évolué en fonction des fonctionnalités ajoutées par notre commanditaire au cours du projet et de nos choix de conception, mais dans l'ensemble il est resté la même par rapport à ce que nous avons créé au début du projet. Nous avons notamment ajouté l'entité Instrument qui correspond à la création d'une annonce pour un instrument et nous avons supprimé les entités Pays et Ville car celles-ci ne sont plus nécessaires car nous utilisons une API qui nous fournit toutes les villes du monde ainsi que leur localisation précise. L'entité rang a également été supprimée au profit d'un bundle appelé Sonata admin (cf sous-partie 2.2.2).

Ayant choisi le framework Symfony qui inclut un ORM, nous avons décidé de l'utiliser. Le nom de l'ORM est Doctrine, qui est celui géré par défaut par Symfony. Un ORM est un outil qui permet d'effectuer des accès à la base de données sans écrire de requêtes SQL. Les accès sont réalisés directement dans le langage utilisé, pour nous en PHP. Pour cela, on a donc dû convertir chaque entité du MCD en classe PHP avec pour attribut, chaque propriété de l'entité.

2.2. Réalisations concrètes pour répondre aux exigences

2.2.1. Réponse au problème de gestion des instruments

2.2.1.1. Comment gérer les instruments que l'on veut prêter






Comme dit précédemment, le concept d'annonce étant obsolète nous avons mis en place un système permettant d'ajouter des instruments à son profil pour ensuite leur attribuer des plannings de disponibilité. L'ajout se fait via un formulaire classique avec les champs essentiels pour identifier un instrument (nom, marque, description, type d'instrument, caution, ...). L'instrument est ensuite ajouté à la liste des instruments que l'on possède mais aucun planning de disponibilité ne lui est encore attribué.

La page répertoriant tous les instruments que l'on possède ressemble à une sorte de "jeu de cartes à collectionner" pour inciter les gens à ajouter leurs instruments et les prêter.

Capture d'écran de la page des instruments possédés

My 10 owned instruments

Reservations + Add instrument +

<p>Set de Cymba...</p>  <p>Sabian 100.00 €</p>	<p>synthétiseur ...</p>  <p>ACME 20000.00 €</p>	<p>Guitar</p> <p>Le piano est un instrument de musique polyphonique, à clavier, de la famille des cordes frappées. Il se présente sous deux formes : le piano droit, les cordes sont verticales, et le piano à queue, les cordes sont horizontales.</p> <p>13/01/2019</p>
<p>guitare électr...</p> 	<p>Guitare ACM...</p> 	<p>super Guitare...</p> 

Les instruments apparaissent sous trois aspects, de face normale avec une photo, de face quand l'annonce est désactivée en grisé, et de dos avec la description

Chaque instrument a ensuite sa propre page où le propriétaire peut visionner les informations et y ajouter des plannings de disponibilité.

Capture d'écran de la page d'un instrument



La couleur verte indique que l'annonce est actuellement disponible, il est possible de cacher l'annonce en appuyant sur "Hide" ("cacher" en français)

2.2.1.2. Comment leur définir des plannings de disponibilité

Quand il n'est plus possible de prêter l'instrument ou quand l'utilisateur ne veut plus le prêter temporairement, il n'est pas nécessaire de supprimer sa fiche. Il y a une fonction qui permet de désactiver l'annonce, pour qu'elle n'apparaisse pas dans les résultats de recherche. Néanmoins les réservations déjà acceptées ne sont pas supprimées.

Pour répondre à la problématique de comment gérer les emplois du temps de chaque instrument, nous avons décidé de laisser le choix aux utilisateurs entre deux types de planning. Le premier est de définir un jour unique où l'instrument sera disponible, entre une heure de début et une heure de fin. Pour éviter de redéfinir le même planning sur une longue période, l'autre option est de définir une récurrence de disponibilité. Par exemple, le propriétaire de l'instrument peut indiquer qu'il souhaite louer son instrument tous les mardis entre le premier janvier 2018 et le premier janvier 2020 entre huit heures et neuf heures, plutôt que de faire 52 plannings d'une journée.

Le formulaire d'ajout du planning utilise plusieurs bibliothèques jquery notamment pour le choix de l'horaire avec un "slider" et pour le choix des dates avec le plugin [Date Range Picker](#) qui nous permet d'avoir des calendriers autres que ceux proposés par le HTML de base. Date Range Picker permet surtout d'avoir un double calendrier, c'est à dire de choisir une date de début et une date de fin en un nombre de clic réduits et dans un unique champs HTML.

Capture d'écran du formulaire d'ajout de planning

Add planning

Many days One day

09/04/2019 - 17/04/2019

13:00 20:00

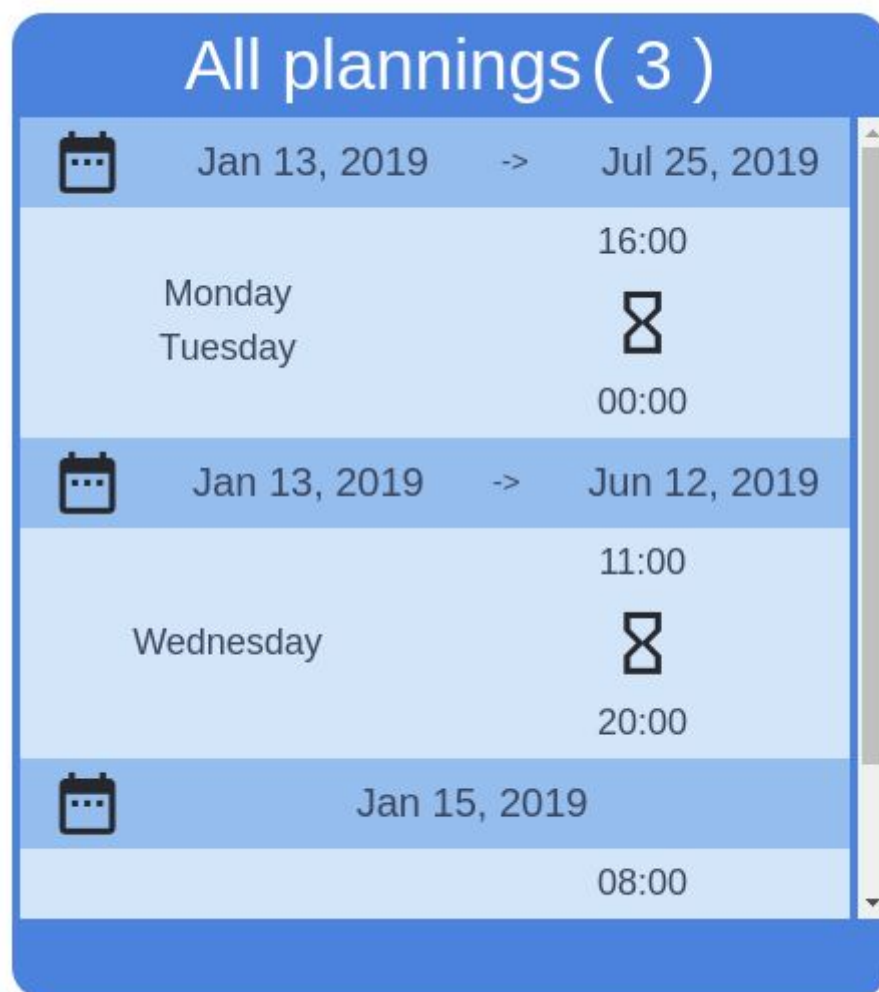
Sun Mon✓ Tue Wed✓ Thu Fri Sat✓ +

Add this planning

Many days correspond à l'ajout d'un planning avec récurrence tandis que One day correspond à l'ajout d'un jour unique

Dans un soucis de clarté il est également possible de visionner tous les plannings que l'on a déjà défini pour un instrument.

Capture d'écran de la liste des plannings définis d'un instrument



Les plannings sont affichés différemment en fonction de leur type, les jours de la récurrence sont à gauche et les horaires en dessous et au dessus du sablier

2.2.2. Réponse au problème des réservations

2.2.2.1. Comment réserver un instrument

Pour réserver les instruments il fallait une interface simple et qui permette de se rendre compte des créneaux disponibles pour l'instrument voulu. Lors de la sélection d'un instrument l'utilisateur arrive sur la page descriptive de l'instrument (aperçu à l'annexe V) où il peut contacter directement le propriétaire ou bien lui faire une demande de réservation directe.

Il peut ensuite réserver l'instrument pour la période qu'il souhaite en remplissant la partie " Reservation Request " composée uniquement de quatre champs. Le premier est un calendrier répertoriant toutes les dates auxquelles

l'instrument est disponible, il suffit ensuite de choisir une heure de début et de fin pour l'emprunt. Les heures disponibles sont évidemment calculées en fonction des réservations déjà existantes et leur affichage est dynamique pour se mettre à jour à chaque fois qu'un des champs est changé (date différente, heures de fin possibles calculées en fonction de l'heure de départ choisie, ...).

La vue récupère toutes les dates possibles lors du clique sur le calendrier grâce à une requête Ajax au contrôleur. Toutes les dates disponibles sont stockées dans un tableau sur la vue qui va exécuter du javascript utilisant le plugin Date Range Picker.

Extrait d'une méthode de la page réservations

```
$('#oc_platformbundle_reservation_date').daterangepicker({  
  [...]  
  isInvalidDate: function (arg) {  
  
    ladata = new Date(arg);  
  
    if (($.inArray(arg.format('DD/MM/YYYY'), datesValides) > -1)) {  
      return false;  
    }  
    else {  
      return true;  
    }  
  }  
}
```

Méthode servant à griser ou non les cases du calendrier

La méthode isInvalidDate s'exécute à chaque affichage du calendrier pour vérifier qu'une case (égale à un jour) doit être sélectionnable ou non. On récupère la date en argument et on utilise la méthode JQuery \$.inArray pour vérifier que la date que l'on veut tester est présente ou non dans le tableau de toutes les dates valides pour l'instrument. Si elle est présente la fonction retourne faux et la date passe en indisponible dans le calendrier.

Pour les réservations d'un instrument, les données sont enregistrées dans une table planning qui stock principalement les horaires ainsi que les dates de début et de fin de disponibilité d'un instrument. Il y a ensuite une seconde table réservation qui stock la date et l'horaire de début de réservation ainsi que la date et l'horaire de fin de réservation.

Pour que les dates de disponibilité d'un instrument soient affichées dynamiquement, la page du client envoie une requête Ajax au contrôleur, qui va effectuer une requête à la base de données et récupérer toute les dates pour

lesquelles l'instrument dispose d'un planning. Une fois les dates disponibles récupérées, le contrôleur va renvoyer à la page toutes les dates au format JSON, qui seront ensuite traitées en JavaScript par la page.

Pour récupérer ces dates et les enregistrer, on utilise le code ci-après. Dans un premier temps, on récupère tous les plannings pour un instrument et on passe dans une boucle pour chacun de ces plannings. Ensuite, on crée une seconde boucle qui va enregistrer dans un tableau toutes les dates disponibles pour chacun des plannings.

Extrait d'une méthode du contrôleur `platformController`

```
$listPlanning = $em  
->getRepository('OCPlatformBundle:Planning')  
->findByInstrument($id);  
  
foreach($listPlanning as $planning)  
{  
    $date = new \DateTime("now");  
    $date->modify('-1 day');  
  
    if($planning->getDateDebut() > $date)  
    {  
        $date = $planning->getDateDebut();  
    }  
  
    for($i = $date; $i <= $planning->getDateFin(); $i->modify('+1 days'))  
    {  
        if(in_array($i->format("N"), $planning->getJours()))  
        {  
            $listDispo[] = $i->format("m-d-Y");  
        }  
    }  
}
```

Méthode permettant de récupérer la liste des plannings d'un instrument

Ensuite, une fois la date choisie par l'utilisateur, le contrôleur va récupérer la date et vérifier les horaires en fonction des horaires de réservation définies par le prêteur ainsi que des horaires de réservation déjà effectués. Enfin, on procède de la même manière pour récupérer les horaires de fin de réservation quand celle du début est sélectionnée. Pour finir, une fois que l'utilisateur valide sa réservation, sa demande est enregistrée dans la table des réservations et passe en attente d'acceptation.

2.2.2.2. Comment répondre aux réservations

L'image ci-dessous correspond à la séparation des différentes pages de gestion des réservations. L'utilisateur peut se rendre sur la page des réservations qui lui sont demandées (2e colonne sur l'image) depuis la page Instrument et peut ensuite naviguer entre les différentes pages de réservation

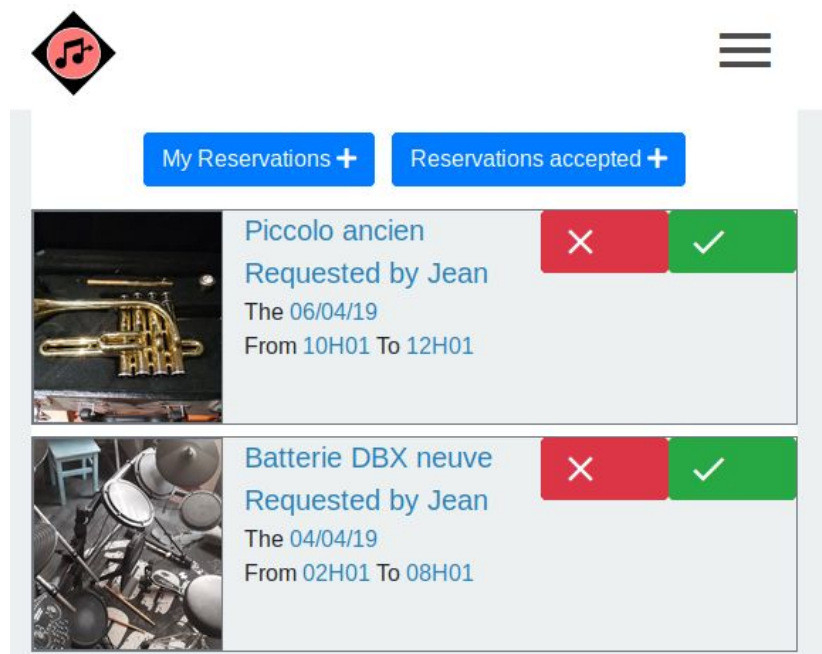
Schéma de séparation des pages pour la gestion des réservations

Réservations que l'utilisateur a demandé	Réservations demandées à l'utilisateur à accepter	Réservations acceptées par l'utilisateur
Réservation 1 	Réservation 4  	Réservation 7
Réservation 2 	Réservation 5  	Réservation 8
Réservation 3	Réservation 6  	Réservation 9

Le schéma montre que le système de réservation est réparti en trois sections distinctes

Les demandes de réservation reçues par le propriétaire d'un instrument s'affichent à l'écran sous forme d'une liste (image ci-dessous). Chaque élément de la liste correspond à une demande de réservation et récapitule toutes les informations à savoir le nom de l'instrument, le nom du demandeur, le jour, les heures de début et de fin ainsi que deux boutons permettant de refuser ou d'accepter la demande.

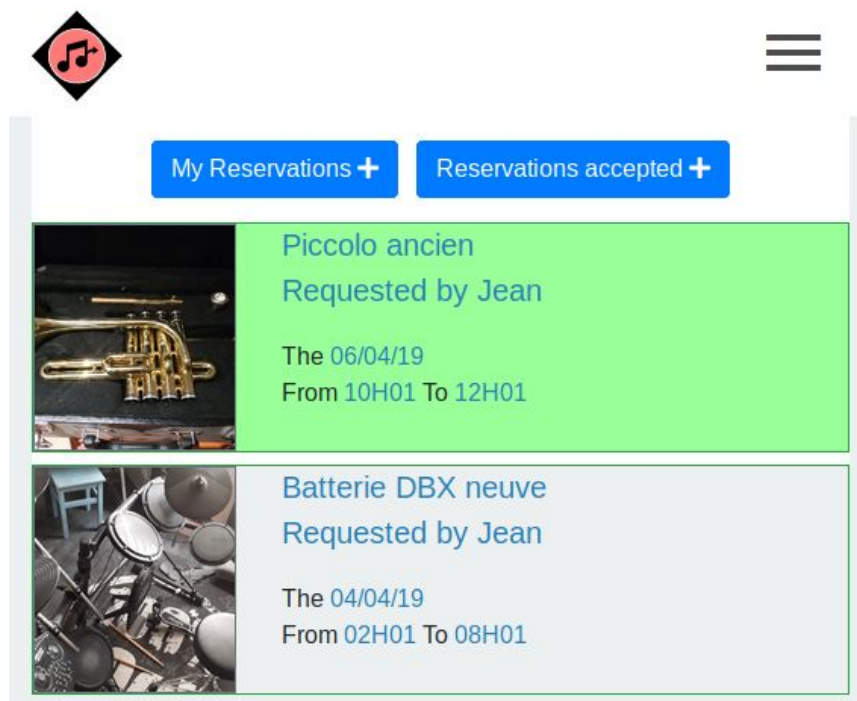
Capture d'écran de la page de gestion des réservations



Il est possible d'accepter ou de refuser les demandes de réservation

Lorsqu'une réservation est acceptée, elle s'affiche sur la page ci-dessous dédiée aux réservations acceptées en récapitulant les informations de la même façon, cependant les boutons ne sont plus présents car la demande est "bloquée" et ne pourra plus être modifiée, ni refusée. Si la réservation est refusée, elle est simplement supprimée de la liste de demande de réservations mais pas de la base de données, pour garder la possibilité de faire un historique.

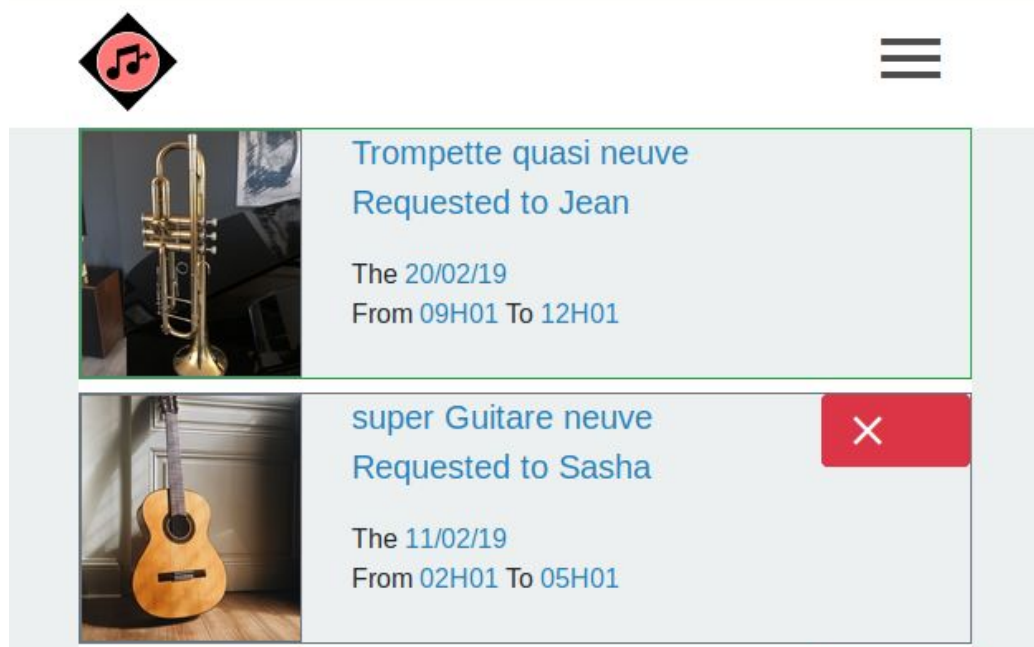
Capture d'écran de la page récapitulant les réservations acceptées



Les réservations acceptées sont en vert

L'image ci-dessous correspond à un extrait de la page "mesReservations", à l'origine de la demande peut consulter le statut de la réservation. Dans le cas d'une réservation acceptée par le propriétaire de l'instrument le cadre de la réservation est vert et la couleur de fond est verte au survol, dans l'autre cas la réservation est en attente et l'utilisateur peut choisir d'annuler la réservation avant qu'elle soit acceptée.

Capture d'écran de la page récapitulant les réservations personnelles



Les instruments que l'on souhaite emprunter

2.2.3. Réponse au problème du panel d'administration

2.2.3.1. Première solution envisagée en dur

Au début nous avons réfléchi à créer nous-même une interface pour les administrateurs et une seconde pour les modérateurs. Mais nous avons trouvé par la suite des bundles permettant déjà de gérer cela. Nous avons donc opté pour la solution du bundle car nous ne voulions pas réinventer la roue. De plus avec cette méthode, nous aurions perdu un temps précieux que nous avons préféré garder pour d'autres fonctionnalités.

2.2.3.2. Premier essai concret avec Easy Admin

Pour réaliser une interface administrateur, nous ne voulions pas réinventer la roue et avons choisi de prendre un bundle pour le configurer. Le bundle EasyAdmin se présentait très bien pour répondre à notre besoin. Nous avons configuré assez rapidement une interface permettant de gérer les utilisateurs, de définir leurs rôles, de gérer et modérer les annonces, ...

L'avantage de cet outil est qu'il est plutôt simple à configurer et qu'il a déjà une interface administrateur créée. Cependant nous nous sommes rendu compte par la suite que celui-ci ne pouvait pas afficher des fonctionnalités en fonction des rôles : c'est-à-dire que nous voulions afficher certaines données pour les

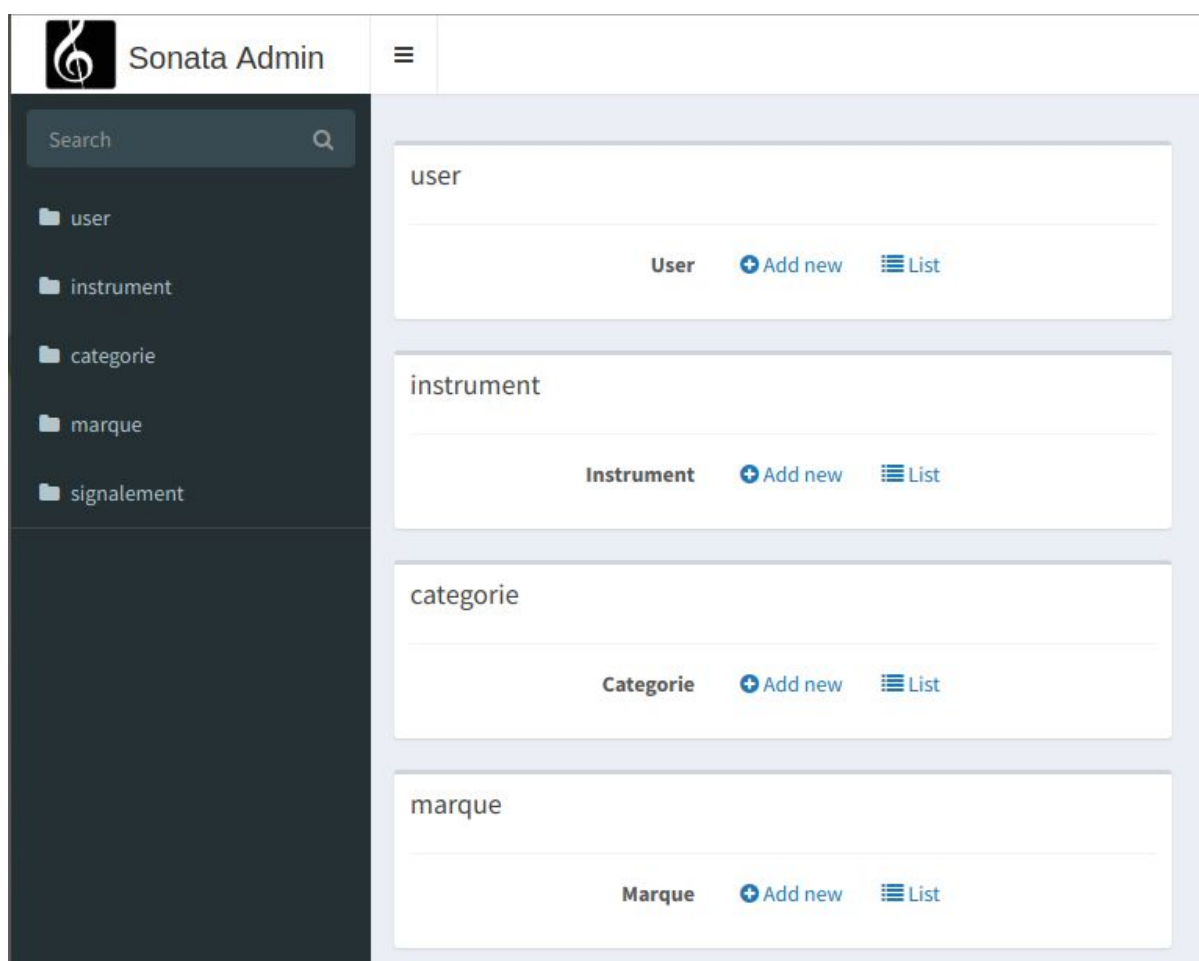
modérateurs et rajouter des informations plus sensible pour les administrateurs, mais le bundle étant trop ancien et ne permettait pas une telle séparation.

2.2.3.3. Deuxième essai concret avec Sonata Admin

Nous avons donc décidé de nous diriger vers un autre système, également géré par Symfony : Sonata Admin.

Sonata Admin est un module de gestion de toute la partie administration qui, sur le papier, nous permet de réaliser la page d'administration et de pouvoir gérer tout le site depuis cette page, c'est-à-dire que ce soit les gestions des différents rôles, les nouvelles demandes de réservations, les nouveaux signalements et d'autres fonctionnalités indispensables pour le site. Pour appréhender le fonctionnement et la création du module nous nous sommes dirigés vers le tutoriel Openclassroom sur SonataAdmin. En suivant ce tutoriel, nous avons pu réaliser les pages de gestion de différents composants du site comme on peut le voir sur l'interface graphique de Sonata Admin ci-dessous.

Affichage de la page d'administration



Capture d'écran de la page d'administration Sonata Admin de notre site

La page des utilisateurs est fonctionnelle, néanmoins elle ne permet pas pour le moment de gérer les rôles des différents utilisateurs.

La page des instruments ainsi que des catégories et des marques est fonctionnelle, on peut les supprimer mais aussi en ajouter depuis cette page.

Pour supprimer un instrument, on a juste à le sélectionner (cliquer sur la case à cocher) et appuyer sur le bouton supprimer tout en bas de la liste. Malheureusement on ne peut pas encore les modifier, une fonctionnalité qui reste à implanter par la suite ainsi que la partie signalement qui n'est pas encore disponible.

2.2.4. Réponse au problème du moteur de recherche

Pour réaliser le moteur de recherche nous avons tout d'abord pensé à utiliser un bundle pour réaliser cette tâche. Nous avons choisi Elasticsearch qui est un logiciel très utilisé pour de la recherche de données. Cependant, il demande énormément de ressources car il redéfinit sa propre base de données en fonction de la base de données du site. Sachant que nous n'avions pas besoin d'un moteur de recherche très performant, nous avons préféré ne pas l'utiliser et créer le nôtre pour gagner en ressources utilisées. Nous avons donc créé une requête dans le repository Instrument qui prend en paramètres les différents éléments que sont la marque, la catégorie, la date de prêt, les mots clés et la ville avec un rayon, à noter que tous ces paramètres sont optionnels.

Sachant que nous utilisons un ORM, nous avons effectué la requête en PHP. Nous vérifions dans un premier temps si le champs mots clés est rempli et si oui, nous ajoutons cette contrainte à la requête finale à la manière d'une concaténation de "where" en SQL. Cette vérification est effectuée de la même façon pour tous les autres champs. La distance ainsi que le rayon sont quant à eux calculés par la suite, car nous ne pouvions pas faire tous les traitements que nous voulions directement dans la requête avec l'ORM. Le calcul de la distance et le trie par localisation sont les principaux traitements exécutés après la requête.

Enfin, nous utilisons les variables de sessions pour sauvegarder les différentes contraintes définies par l'utilisateur dans le moteur de recherche pour que celui-ci ne perde pas les informations en passant d'une page à une autre du résultat du moteur de recherche.

Le code ci-dessous se situe dans le repository Instrument et permet la création d'une requête sur les entités instrument et planning pour récupérer seulement les annonces d'instruments encore disponibles à ce jour.

Extrait d'une méthode du repository InstrumentRepository

```
$query = $this->createQueryBuilder('ins')
    ->leftJoin('OCPlatformBundle:Planning', 'pla', 'WITH', 'ins.id = pla.instrument
and ins.enabled = true')
    ->andWhere('pla.dateFin >= :dateActuel and ins.dispo = 1')
    ->setParameter('dateActuel', $dateActuel);
```

Méthode permettant de construire une requête SQL

Cette condition permet de vérifier si une marque est ajoutée dans le moteur de recherche. Si tel est le cas on ajoute à la requête la condition avec la marque passée en paramètre.

Extrait de code du repository `InstrumentRepository`

```
if($marque != null)
{
    $query->andWhere('ins.marque = :marque')
    ->setParameter('marque', $marque);
}
```

Code permettant d'ajouter une condition where à la requête si une marque est définie

Enfin, cette ligne permet de retourner la requête complète au controller :

```
return $query->getQuery();
```

La fonction complète permettant de faire une recherche en fonction du moteur de recherche se trouve en annexe VI.

2.2.5. Réponse au problème de gestion des cautions

N'ayant pas pu développer le système de caution voulu qui était trop long et trop compliqué pour un projet de DUT nous avons décidé de gérer le problème en laissant les utilisateurs fixer le prix de la caution lorsqu'ils ajoutent l'instrument. Les modérateurs peuvent vérifier les prix de ces cautions pour voir si il n'y a rien d'indécent et ensuite lors du prêt de l'instrument l'emprunteur donne la somme au prêteur qui lui rendra lors du retour de l'instrument si il est en bon état, en cas de conflit des messages avec des preuves peuvent être envoyés aux modérateurs (module à implémenter encore).

2.3. Avancée du projet aux différents lots

2.3.1. Lot 1

Le premier rendu de lot comprenait l'ensemble des diagrammes de conception (MCD, UML, ...), des maquettes du site ainsi qu'une étude qualitative des différents CMS et l'explication de notre choix d'utiliser le framework Symfony. Ce lot comprend également la possibilité de s'inscrire et de se connecter au site.

2.3.2. Lot 2

Lors du second lot, nous avons principalement créé les entités ainsi que leur repository respectif. Au niveau du design, nous avons ajouté la page d'accueil, les formulaires de connexion, inscription et d'ajout d'une annonce pour un instrument. Enfin, nous avons ajouté une page permettant de lister les différentes annonces et permettant de visualiser une annonce d'un instrument en particulier.

2.3.3. Lot 3

Pour le troisième lot, nous avons amélioré la page d'affichage d'un instrument en ajoutant un système de réservation dynamique et en refaisant la mise en page. Nous avons ajouté un formulaire d'ajout d'un planning et on a aussi rajouté un moteur de recherche permettant à l'utilisateur de cibler sa recherche en fonction de plusieurs paramètres notamment par localisation, par marque ainsi que par catégorie. Enfin, nous avons ajouté un système de signalement pour éviter tout problème sur une annonce.

Nous avons donc explicité l'ensemble des réalisations conceptuelles et concrètes ainsi que l'avancement du projet à chaque lot. Dans la dernière partie nous allons présenter notre rendu final et les perspectives d'avenir du projet.

3. Rendu final

Dans cette partie nous allons parler de notre rendu final en rappelant l'avancement du projet au dernier lot puis nous ferons une rétrospective et nous analyserons les problèmes rencontrés lors du projet et enfin nous nous intéresserons aux choses à développer et à la reprise du projet par les 1A.

3.1. Avancée de notre application au dernier lot

Au dernier lot, nous avons réalisé un système de messagerie privé entre utilisateurs pour qu'ils puissent communiquer entre eux. Nous avons aussi ajouté un système de gestion des réservations pour que l'utilisateur qui emprunte puisse voir l'état de ses demandes mais aussi pour que le prêteur puisse accepter ou refuser une demande de réservation.

Le site web n'est pas encore utilisable car il manque beaucoup de choses et de visuels pour que le site soit publiable en ligne (notamment des mentions légales), mais les principales fonctionnalités sont utilisables et fonctionnelles.

3.2. Rétrospective et analyse des problèmes rencontrés

3.2.1. Problèmes techniques durant notre développement

Nous avons eu quelques soucis avec la base de données car après trois mois de non connexion sur leur site (et non pas trois mois d'inactivité sur la base de données), le site a tout simplement supprimé la base de données. Bien heureusement, sachant que nous avons un ORM, nous avons pu reconstruire la base très vite. De plus, nous avons quelques données de la base de données sauvegardé.

Enfin, nous avons abandonné l'idée d'utiliser le serveur fourni par l'université car la version PHP de leur serveur est en PHP 7.0 et le framework Symfony fonctionne sous la version PHP 7.2. De plus nous n'avons pas pu utiliser le serveur de l'université pour héberger la base de données, car l'utiliser impliquait forcément d'héberger notre site sur le serveur universitaire, ce qui est impossible.

3.2.2. Problèmes à long terme et réutilisabilité

Au moment de la réalisation du projet nous avons constaté que plusieurs problèmes pourraient survenir, il faudra penser à louer un serveur pour pouvoir héberger le site ainsi que la base de données et avoir une capacité suffisante pour accueillir un nombre élevé de personnes.

De plus, nous utilisons une API nommée Mapbox qui nous offre un service de localisation qui propose un nombre de 50 000 requêtes gratuites par mois puis une fois cette limite dépassée, le service devient payant. Le prix est de 0.50\$ toutes les 1 000 requêtes tous les mois mais il est tout à fait possible de prendre contact avec leur service pour souscrire à un forfait.

3.3. Choses à développer pour le futur du site

3.3.1. Aspect communautaire à développer

L'aspect communautaire doit être un grand pilier du site mais nous n'avons pas eu le temps de développer ce point. En effet, Monsieur Levilly souhaite que le site puisse servir de "plateforme de rencontre" entre musiciens. Pour le moment les interactions possibles entre nos utilisateurs sont l'emprunt d'instrument et la messagerie instantanée. Nous n'avons pas eu le temps de développer les fonctionnalités qui permettent à la communauté d'interagir davantage comme la possibilité d'organiser un mini-concert entre les utilisateurs.

3.3.2. Fonctionnalités à implémenter

Comme évoqué au point précédent Monsieur Levilly souhaite favoriser les interactions sociales entre les musiciens par l'intermédiaire du site. Cela devra passer par une page de création/gestion d'évènement qui devra permettre aux utilisateurs d'organiser leurs propres rencontres que ce soit un mini-concert ou une répétition en groupe.

3.4. Reprise du projet par les étudiants de première année

Comme présenté précédemment, le site n'est pas totalement terminé. Il reste des fonctionnalités à implémenter, des documents à rédiger ainsi que des soucis à régler. Le commanditaire a alors décidé de proposer une reprise de projet pour les étudiants de première année.

Nous avons rencontré deux fois le groupe de première année qui nous succédera sur ce projet. Il sera composé de quatre étudiants appelés Luc Audabram, Jeremy L'Ecuyer, Loic Le Goas et Matthis Rivat. Pendant la première rencontre nous avons présenté le projet, son contexte, son but et ce que nous avons déjà fait. Nous les avons également conseillé sur les outils à maîtriser et sur les langages à apprendre pour une bonne reprise du projet. Comme ils ont eu des difficultés à installer les logiciels et dépendances nécessaires à Symfony nous avons convenu d'un second rendez-vous où nous leur avons montré comment tout installer.

Les 1A devront se focaliser sur l'aspect communautaire qui est un des grands piliers du site que nous avons un peu mis en retrait pour privilégier l'aspect fonctionnel du site. Ils pourront également améliorer ou optimiser notre code, ainsi que rajouter les fonctionnalités demandées par le commanditaire que nous n'avons pas été en mesure de finir. Ils devront également résoudre les problèmes

de compatibilité entre le site et les serveurs de l'université pour pouvoir le mettre en ligne, ou bien trouver un autre hébergeur fiable.

Conclusion

Le but du projet était de développer un site web permettant aux musiciens du monde entier de gérer le prêt de leurs instruments lors de leurs déplacements. Nous avons effectué de nombreux rendez-vous avec le commanditaire pour préciser ses besoins et attentes du projet, ce qui nous a permis d'avoir une idée très précise de la tâche à effectuer.

Le projet était laborieux au début mais la deuxième année et les nouveaux choix technologiques ont permis d'accélérer grandement son déploiement. Au final nous avons réussi à mettre en place les principales fonctionnalités du site mais il manque l'aspect communautaire qui est encore trop peu présent. Ce projet nous a permis d'apprendre à travailler en équipe et nous a démontré la difficulté de gérer un projet informatique de A à Z sur une période aussi longue.

À notre grand plaisir le projet va être repris par des étudiants de première année, ils pourront alors continuer ce que nous avons commencé en y ajoutant des fonctionnalités auxquelles nous n'avons pas pensé et améliorer celles existantes.

Bibliographie

Cours de PHP et de base de données de M. Porcq

Cours d'UML et de JavaFX de M. Brutus

Cours d'expression-communication de Mme Passoni-Chevalier

Document "Méthodologie des mémoires et des soutenances universitaires appliquée au DUT INFORMATIQUE"

Cours d'IHM de Mme Drafat

Cours de Merise de Mme Mandin, M. Brutus et M. Porcq

<https://marvelapp.com/>

<http://www.umletino.com/>

<https://symfony.com/>

<https://twig.symfony.com/>

<https://www.couchsurfing.com/>

<https://openclassrooms.com/>

<https://www.joomla.fr/>

<https://fr.wordpress.org/>

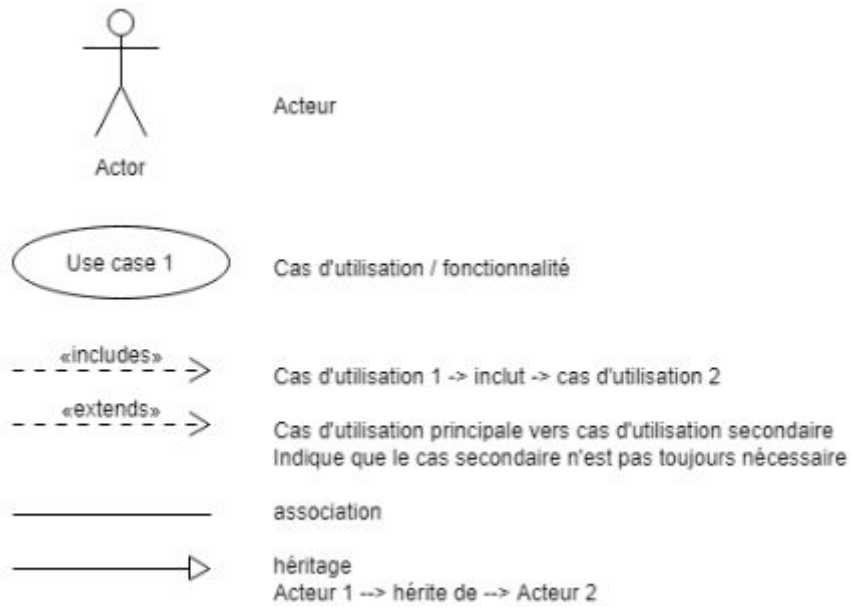
<https://www.drupal.org/>

<https://www.spip.net/>

<http://www.daterangepicker.com/>

Annexes :

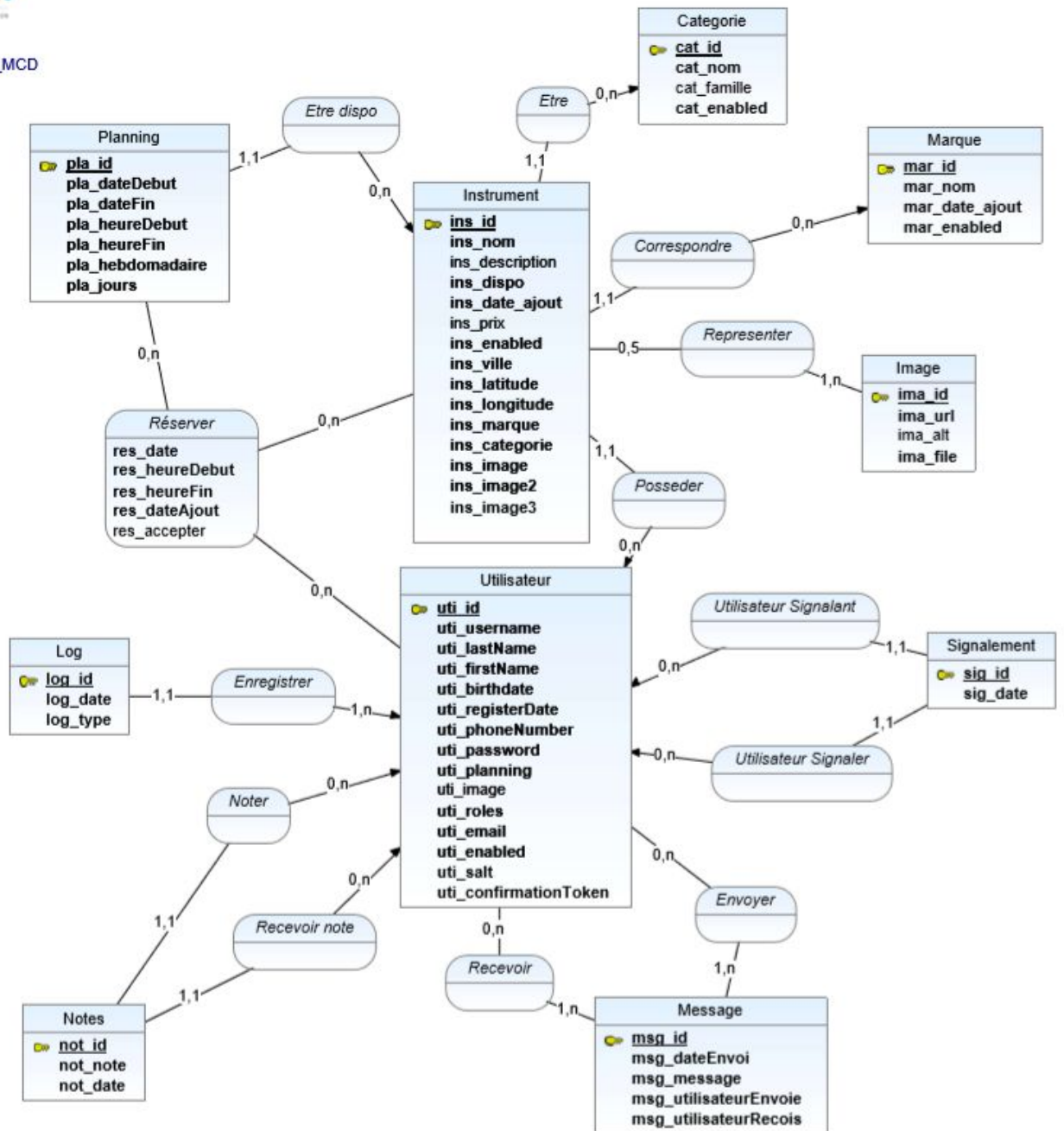
Annexe I : Explication des symboles UML




Annexe II : MCD de départ du projet



MCD_V2_MCD



Annexe III : page d'administration des instruments en ligne


Sonata Admin

Instrument List

Search

user

instrument

categorie

marque

signalement

Filters 0

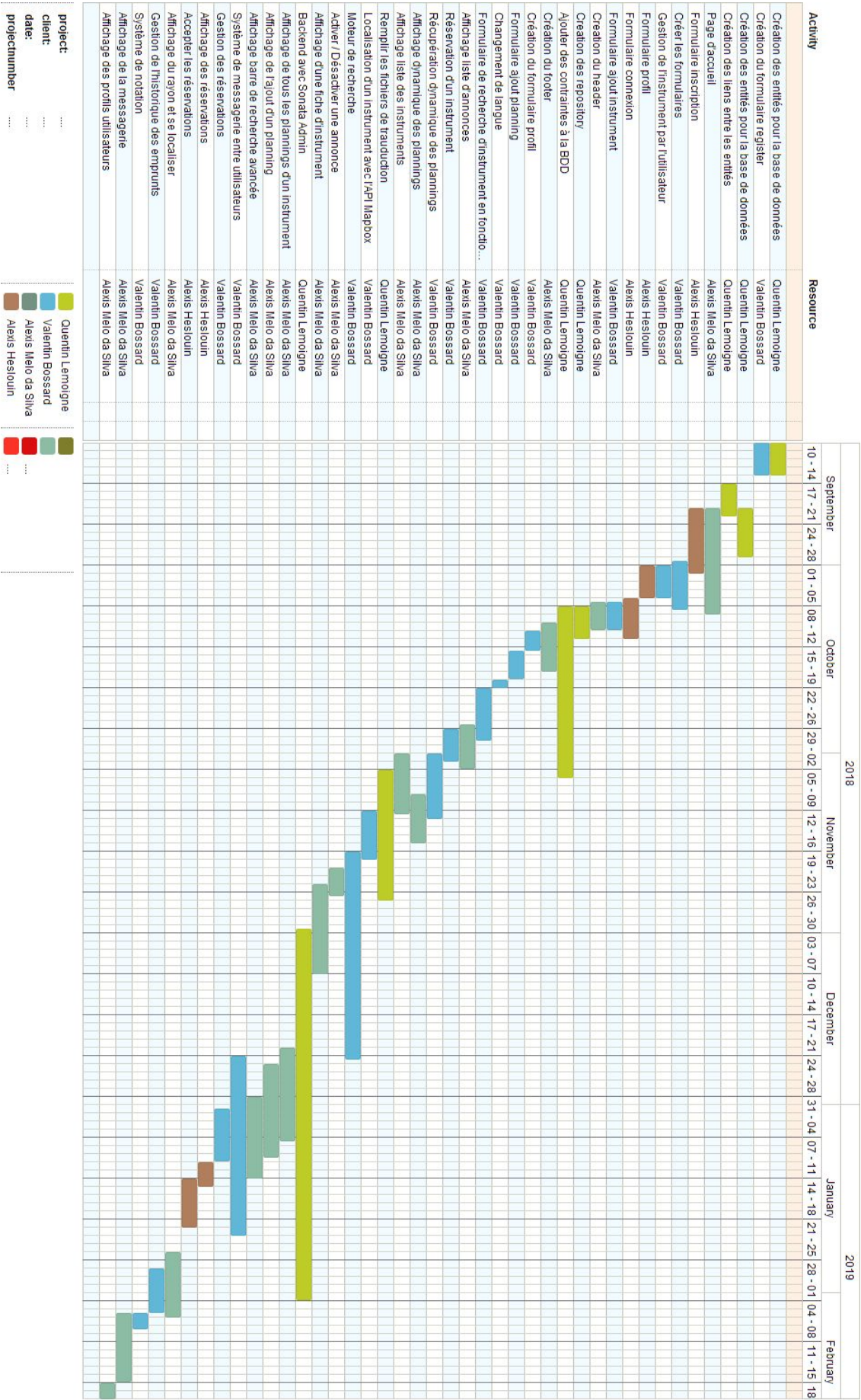
+ Add new

<input type="checkbox"/>	Nom	Description	Disponibility	Marque	Categorie
<input type="checkbox"/>	Piano	Je loue mon piano Yamaha presque neuf, et accordé récemment.	1	Yamaha	Guitar
<input type="checkbox"/>	Piano	Je loue mon piano récemment accordé et en très bon état.	1	Yamaha	Guitar
<input type="checkbox"/>	Trompette quasi neuve	La trompette est un instrument de musique à vent de la famille des cuivres clairs. Elle est fabriquée dans un tube de 1,50 m de long comme le cornet.	1	Adam Hall	Guitar
<input type="checkbox"/>	Piccolo ancien	e prete cette trompette piccolo avec tons Sib et La car je n'ai absolument pas l'occasion de m'en servir, et j'éprouve de plus des difficultés à en jouer correctement. Elle est prêtée avec sa boîte et son embouchure (type 14A4A). Les pistons et les coulisses fonctionnent parfaitement.	1	Alto	Guitar
<input type="checkbox"/>	Orgue electrique avec clavier neuf	L'orgue est un instrument à vent multiforme dont la caractéristique est de produire les sons à l'aide d'ensembles de tuyaux sonores accordés suivant une gamme définie et alimentés par une soufflerie. L'orgue est joué majoritairement à l'aide d'au moins un clavier et le plus souvent d'un pédalier.	1	BC Rich	Guitar
<input type="checkbox"/>	Batterie DBX neuve	Photo de batterie avec divers micros. Une batterie en studio d'enregistrement. Une batterie est un ensemble d'instruments de percussion (de type fûts et cymbales) disposé pour être joué par une seule personne à l'aide de baguettes et de pédales.	1	DBX	Guitar
<input type="checkbox"/>	super Guitare neuve	La guitare est un instrument à cordes pincées.	1	Ableton	Contrabasso

0

sf

Annexe IV : diagramme de Gantt



Annexe V : Exemple d'une annonce

Orgue electrique avec clavier neuf



■ BC Rich

▲ Guitar

\$ 120

Paris, France

👤 Sasha

📞 Show number

💬 Message

Description

L'orgue est un instrument à vent multiforme dont la caractéristique est de produire les sons à l'aide d'ensembles de tuyaux sonores accordés suivant une gamme définie et alimentés par une soufflerie. L'orgue est joué majoritairement à l'aide d'au moins un clavier et le plus souvent d'un pédalier.

Reservation request

01/04/2019

Annexe VI : Code de la méthode retournant la requête des instruments disponibles

```
public function getAnnonceDispoRecherche($nom, $marque, $categorie, $date,
$page, $nbPerPage)
{
    $dateActuel = new \DateTime('now');
    $dateActuel->modify('-1 day');

    $query = $this->createQueryBuilder('ins')
        ->leftJoin('OCPlatformBundle:Planning', 'pla', 'WITH', 'ins.id = pla.instrument
and ins.enabled = true')
        ->andWhere('pla.dateFin >= :dateActuel and ins.dispo = 1')
        ->setParameter('dateActuel', $dateActuel);

    if($nom != null)
    {
        $query->andWhere('upper(ins.nom) like upper(:nom)')
            ->setParameter('nom', '%' . $nom . '%');
    }

    if($marque != null)
    {
        $query->andWhere('ins.marque = :marque')
            ->setParameter('marque', $marque);
    }

    if($categorie != null)
    {
        $query->andWhere('ins.categorie = :categorie')
            ->setParameter('categorie', $categorie);
    }

    if($date != null)
    {
        $query->andWhere(':date between pla.dateDebut and pla.dateFin')
            ->setParameter('date', $date)
            ->andWhere('pla.jours like :jour')
            ->setParameter('jour', 'a:_%' . $date->format('N') . '%');
    }
    return $query->getQuery();
}
```


FICHE DE LECTURE

du mémoire de conduite de projet 2A
établie par BOURTAL-ZARAT Médhi, EMERY Romain et LECLERE
Charlotte, et transmise le 21 mars 2019

Intitulé du projet : Réalisation d'un site web de prêt d'instruments de musique

NOMS et Prénoms des membres du groupe de projet : Valentin BOSSARD; Alexis HESLOUIN; Quentin LEMOIGNE; Alexis MELO DA SILVA.

1. La forme du mémoire

→ Le respect des normes de présentation

- **Couverture** :
 - Si possible, vous pouvez ajouter une illustration sur la page de couverture pour la personnaliser.
 - Il manque les mots clés en anglais.
- **Liminaires** :
 - Les annexes et l'introduction ne sont pas dans le sommaire et pas non plus de table des annexes.
 - Les numéros des parties n'apparaissent pas sur le sommaire.
- **Corps du mémoire** :
 - Manque d'introductions/conclusions partielles et de transitions entre les parties.
 - Les parties ne commencent pas à chaque fois sur une nouvelle page.
- **Visuels** :
 - Tous les visuels ne sont pas titrés et légendés et le backlog n'est pas introduit.
- **Appareil de référence** :
 - Annexes non numérotées et titrées.
 - Aucune bibliographie.
- **Respect du volume de pages indiqué** :
 - Seulement 6 pages pour l'instant.

2. Le contenu du mémoire

→ La cohérence

- **Plan :**
 - Plan logique et bien articulé autour de la problématique mais attention à bien l'équilibrer en contenu.
 - Attention aux abréviations (2A au lieu de 2e année).
- **Logique du discours :**
 - Bonne articulation du discours.
- **Introduction :**
 - Bonne introduction du sujet, de sa problématique et du plan.
- **Conclusion :**
 - Bonne conclusion de la conduite du mémoire.

→ La pertinence

- **Caractère globalement analytique du propos :**
 - Bonne justification des outils utilisés et prise de distance par rapport à l'action menée.
- **Problématisation autour de la mission à effectuer :**
 - Bonne problématique.
- **Capacité à présenter l'existant, puis le projet ou la mission :**
 - Bonne présentation du client et de l'existant.
- **Qualité de la réflexion :**
 - Réflexion de qualité.
- **Recours pertinent à des connaissances, des compétences et des outils universitaires :**
 - Certains liens sont établis mais il ne faut pas oublier de les mentionner dans la bibliographie.