

Table des matières

Résumé.....	i
Avant-Propos.....	ii
Table des matières	iii
Liste des tableaux.....	vi
Liste des figures.....	vii
Liste des abréviations.....	x
Introduction	1
1. Analyse de l'entreprise Coucou&Co.....	2
1.1. Introduction.....	2
1.2. L'entreprise.....	2
1.3. But de l'entreprise	2
1.4. Clients cibles	3
1.5. Fonctionnement	3
1.5.1. Processus de location.....	4
1.6. Plateforme et outils	7
1.6.1. BookingSync	7
1.6.2. Excel.....	7
1.6.3. Rental Ninja	8
1.6.4. Bexio	9
1.7. Architecture logicielle.....	10
1.8. Technologies.....	11
1.8.1. BookingSync	11

1.8.2.	Fichier Excel.....	11
1.9.	Limitations et améliorations futures	12
2.	Analyse des besoins	14
2.1.	Définition du projet	14
2.1.1.	Fonctionnalités importantes	14
3.	Etat de l'art.....	15
3.1.	Solutions existantes.....	15
3.1.1.	WooCommerce – Plugin WordPress.....	15
3.1.2.	Zen Cart	16
3.1.3.	Magento	16
3.1.4.	Shopify.....	17
3.1.5.	Conclusion	18
3.2.	Technologies.....	19
3.2.1.	Django – Framework Python.....	19
3.2.2.	Express – Framework JavaScript	20
3.2.3.	Laravel – Framework PHP	20
3.2.4.	Conclusion	21
3.3.	Choix final	22
4.	Implémentation.....	23
4.1.	Outils.....	23
4.1.1.	Jetbrains IDE – PhpStorm	23
4.1.2.	GitHub	23
4.1.3.	PHPUnit	24
4.1.4.	Laravel Dusk	24
4.1.5.	Travis CI	24

4.1.6.	Apache2.....	25
4.1.7.	MariaDb.....	25
4.2.	Méthodologie	25
4.2.1.	Sprint 0	26
4.2.2.	Sprint 1	27
4.2.3.	Sprint 2	27
4.2.4.	Sprint 3	28
4.3.	Fonctionnalités	28
4.3.1.	Visiteur	28
4.3.2.	Client	31
4.3.3.	Service Provider.....	33
4.4.	Difficultés.....	39
4.4.1.	Réservation pour un client	39
4.4.2.	Formulaires dynamiques.....	39
4.5.	Améliorations futures possibles	45
4.5.1.	Champs dynamiques	45
4.5.2.	Modification poussée des services	46
4.5.3.	Intégration d'un système de paiement.....	47
4.5.4.	Intégration de la synchronisation via API.....	47
	Conclusion	48
	Références.....	49
	Déclaration de l'auteur	50

Liste des tableaux

Tableau 1 : Comparaison des solutions du marché	18
Tableau 2 : Comparaison des frameworks.....	21

Rapport-Gratuit.com

Liste des figures

Figure 1 : Schéma de processus de location	6
Figure 2 : Logo BookingSync.....	7
Figure 3 : Logo Excel	7
Figure 4 : Logo Rental Ninja	8
Figure 5 : Logo Bexio	9
Figure 6 : Architecture logicielle actuelle de Coucou&Co.....	10
Figure 7 : Architecture logicielle future de Coucou&Co	12
Figure 8 : Logo WooCommerce.....	15
Figure 9 : Logo ZenCart	16
Figure 10 : Logo Magento	16
Figure 11 : Logo Shopify	17
Figure 12 : Logo Django.....	19
Figure 13 : Logo Express.....	20
Figure 14 : Logo Laravel.....	20
Figure 15 : Logo PhpStorm	23
Figure 16 : Logo GitHub.....	23
Figure 17 : Logo PHPUnit.....	24
Figure 18 : Logo Laravel Dusk.....	24
Figure 19 : Logo Travis Ci.....	24
Figure 20 : Logo Apache2	25

Figure 21 : Logo MariaDB	25
Figure 22 : Déroulement du sprint 0.....	26
Figure 23 : Burndown Chart du sprint 0.....	27
Figure 24 : Page d'accueil du site	28
Figure 25 : Détail d'un service pour un visiteur	29
Figure 26 : Formulaire d'enregistrement.....	30
Figure 27 : Champ de vérification d'email utilisateur dans la base de donnée	30
Figure 28 : Détail d'un service pour un client	31
Figure 29 : Confirmation de réservation	31
Figure 30 : Lien des réservations.....	32
Figure 31 : Annulation d'une commande.....	32
Figure 32 : Lien du panel de gestion du fournisseur de service.....	33
Figure 33 : Panel de gestion du fournisseur de service	33
Figure 34 : Menu latéral	34
Figure 35 : Liste des services du fournisseur	34
Figure 36 : Formulaire d'ajout d'un service	35
Figure 37 : Liste des champs personnalisés	35
Figure 38 : Champs personnalisés ajoutés au service.....	36
Figure 39 : Liste des réservations du fournisseur	37
Figure 40 : Choix du service à réserver	37
Figure 41 : Formulaire de réservation d'un fournisseur pour un client	38

Figure 42 : Exemple d'email de notification envoyé au fournisseur.....	38
Figure 43 : Classe générique "InputType.php"	40
Figure 44 : Classe "DateInput.php"	41
Figure 45 : Fichier de configuration "DynamicInputTypes.php"	41
Figure 46 : Méthode de génération des champs personnalisables	42
Figure 47 : Méthode de génération du formulaire de réservation.....	43
Figure 48 : Modèles Eloquent "Sellable.php", "Order.php et "SellableField.php"	44
Figure 49 : Modification d'un service.....	46

Liste des abréviations

API	Application Programming Interface
CMS	Content Management System
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
ORM	Object Relational Mapping
OTA	Online Travel Agency
PMS	Property Management System
REST	Representational State Transfer
SaaS	Software as a Service

Introduction

Ce travail de Bachelor a pris racines dans l'esprit de M. Andreas Fischer, qui eût l'idée d'une plateforme où fournisseurs de services et clients se croisent pour créer un marché. L'objectif de ce travail est d'implémenter un prototype de plateforme correspondant au maximum au besoin des clients et des entreprises.

L'entreprise Coucou&Co étant mandataire, par le biais de M. Fischer, de deux travaux de Bachelor 2019, les professeurs respectivement responsables de ces travaux ont demandé à M. Sylvain Roh, auteur de l'autre travail, et nous-même de collaborer pour la partie d'analyse de l'entreprise.

L'analyse ayant relevé les besoins de l'entreprise, nous avons décidé conjointement avec cette dernière de rendre la plateforme indépendante avec une possibilité de lier le PMS (Property Management System) de Coucou&Co au moyen d'une API (Application Programming Interface) par la suite. Nous avons également défini les tâches et les priorités du développement à venir.

Une fois ces informations récoltées, nous avons commencé la recherche et l'analyse des solutions et des technologies existantes pour une solution développée de zéro. À la suite de cette analyse, nous avons choisi le framework PHP Laravel pour le développement de la plateforme.

Le plus gros défi de ce projet réside en la gestion des formulaires de réservation. Chaque service étant différent, le fournisseur doit être en mesure de pouvoir définir des champs personnalisés en fonction des informations devant être livrées par le client.

1. Analyse de l'entreprise Coucou&Co

1.1. Introduction

Cette analyse a été rédigée conjointement avec M. Sylvain Roh. qui a effectué un travail de Bachelor parallèle pour Coucou&Co. C'est pour cette raison que de grandes similitudes se retrouveront dans le travail de M. Roh au même ce chapitre.

1.2. L'entreprise

Coucou&Co est une entreprise de gestion locative à courte durée de biens immobiliers.

L'équipe est composée de Nicolas Montagner et d'Andreas Fischer. Nicolas, le directeur général, détient un Bachelor en management du tourisme et Andreas est le directeur technique et possède un Bachelor en Informatique.

Coucou&Co est initialement un projet étudiant créé en septembre 2016 durant le cours de « Business Expérience » à la HES-SO de Sierre. La startup a ensuite été officiellement fondée en octobre 2017 par Nicolas et Andreas.

L'entreprise offre à ses clients la possibilité de louer leurs biens immobiliers sans se soucier de rien. Elle met en place tout ce qui est nécessaire au bon déroulement du séjour pour les locataires ainsi qu'à la restitution du bien au propriétaire

Coucou&Co s'occupe de poster les annonces des biens immobiliers sur différents sites de location (OTA) et adapte les prix en fonction de l'offre et de la demande afin d'arriver au meilleur équilibre possible en matière de rendement.

La startup gère actuellement plus de 50 propriétés et a pu accueillir plus de 1000 clients depuis le début de ses activités.

1.3. But de l'entreprise

Le but de l'entreprise est d'offrir à ses clients un service para-hôtelier de qualité par le biais d'une solution « clé en main » en Suisse.

1.4. Clients cibles

Coucou&Co cible deux types de clientèle.

Le premier client type de Coucou&Co est un particulier souhaitant louer un bien qu'il possède sans avoir à gérer tout ce que cette location implique notamment ménage ou encore remise des clés.

Le second est une personne, souvent touriste, souhaitant louer un logement pour une courte durée lors de ses vacances.

Bien que la plupart des réservations soient gérées par des tiers, l'entreprise possède également une plateforme de réservation en ligne booking.coucounco.ch.

1.5. Fonctionnement

L'entreprise et le propriétaire organisent un premier contact afin de discuter et de conclure un contrat satisfaisant les deux parties.

Coucou&Co met le bien en valeur à l'aide de photos et récupère toutes les informations le concernant afin de publier une annonce de qualité en vue de la location. Les demandes de réservation sont gérées de manière réactive.

Coucou&Co offre des conseils personnalisés aux locataires quant à la situation de l'objet (lieux à visiter, restaurants, activités, etc.), qui disposent d'une transparence parfaite de la part de l'entreprise.

Cette dernière s'assure que le bien soit propre au moment de la restitution, à savoir dans le même état qu'à l'arrivée du locataire. Elle s'occupe de la gestion des clés, du linge, etc., et veille à garder le logement en bon état.

Trois taxes s'appliquent en vue d'une différenciation des tarifs. Lors de l'inscription, une première taxe, pour la mise en place, varie selon les besoins du propriétaire. Puis un forfait annuel de 250 francs est demandé pour la gestion du bien. Finalement une commission de 18% est pratiquée sur chaque réservation. Ces montants fluctuent en fonction du contrat établi avec le client.

1.5.1. Processus de location

Un schéma de processus métier (cf. figure 1 ci-après) a été mis en place en collaboration avec M. Fischer et M. Roh, afin de comprendre et de visualiser le fonctionnement de l'entreprise lorsqu'un client effectue une réservation.

Les parties prenantes concernées par le processus en question sont les suivantes.

- Le client (Guest) : la personne voulant louer un bien, le locataire.
- BookingSync : la plateforme de gestion des biens et réservations utilisée par Coucou&Co.
- L'entreprise Coucou&Co : les employés de l'entreprise, les outils (p. ex. le document Excel gérant les décomptes ou Stripe).
- Le propriétaire du logement (Owner) : la ou les personne(s) propriétaire(s) du bien immobilier.
- Les fournisseurs de services (Provider) : l'entreprise fournissant le service de ménage dans les logements.

Pour commencer, le client amorce le processus en créant une nouvelle réservation au travers d'un des OTA utilisés par Coucou&Co (AirBnB, Booking.com, booking.coucounco.com, etc.) ou en contactant directement l'entreprise par téléphone ou par email.

Si la réservation a été effectuée via un OTA, elle sera automatiquement insérée dans la plateforme BookingSync. A contrario, si elle a été faite par voie directe, elle devra être entrée manuellement dans BookingSync.

Une fois celle-ci créée, BookingSync envoie plusieurs notifications à différentes parties. Un email concernant la caution est automatiquement envoyé au client. De plus, Coucou&Co reçoit un email avec toutes les informations utiles au traitement de la réservation. Le propriétaire, quant à lui, reçoit uniquement les informations le concernant.

Le client doit, au moyen des informations reçues dans l'email, s'acquitter du montant de la caution. Une fois le paiement validé, il reçoit toutes les données lui permettant de profiter au mieux de son séjour.

Un jour avant son départ, il reçoit une demande d'évaluation du service proposé par l'entreprise.

Lorsque le séjour prend fin la caution est remboursée, à condition que Coucou&Co ait confirmé que tout soit en ordre.

En ce qui concerne l'entreprise, dix jours avant le début de la réservation, elle prend contact avec le client par téléphone afin de vérifier que toutes les informations lui ont été transmises. En outre, elle réserve un service de nettoyage au près d'une entreprise externe, au minimum trente jours avant la fin de la réservation.

Une fois le séjour terminé, l'entreprise de ménage s'occupe de nettoyer le logement et annonce le travail effectué à Coucou&Co. Une fois la confirmation reçue, Coucou&Co peut clore la réservation et ainsi déclencher le remboursement de la caution à l'invité.

Enfin, une fois par mois, l'entreprise récupère les données des réservations se trouvant sur BookingSync, puis les importe dans un fichier Excel afin de générer les décomptes destinés aux propriétaires, qui les recevront par email.

1.6. Plateforme et outils

1.6.1. BookingSync



Figure 2 : Logo BookingSync

Plateforme de gestion de location de vacances (Channel Manager), BookingSync est une startup française ayant pour objectif de faciliter la vie des propriétaires, des gestionnaires et des agences de locations. Que ce soit pour gérer les réservations, générer une facture, accepter un paiement ou établir un contrat, BookingSync dispose de la fonctionnalité adéquate. Axée sur la simplicité et la facilité d'utilisation, cette entreprise a développé un logiciel intuitif et convivial à l'intention des propriétaires et des gestionnaires, afin de les accompagner lors de chaque étape du processus.

Coucou&Co utilise cet outil pour synchroniser les réservations provenant des plateformes suivantes.

- Airbnb
- Booking.com
- Tripadvisor
- Site de l'entreprise elle-même (<https://booking.coucounco.ch/>)

1.6.2. Excel



Figure 3 : Logo Excel

Microsoft Excel est un logiciel tableur appartenant à la suite bureautique MS Office, développée et distribuée par Microsoft.

Coucou&Co utilise ce programme pour :

- stocker les informations de chaque logement (info, textes, équipements, etc.) ;
- répertorier les réservations ;
- et calculer les montants à reverser au propriétaire des biens immobiliers en fin de mois.

1.6.3. Rental Ninja



Figure 4 : Logo Rental Ninja

Rental Ninja est une plateforme permettant d'automatiser le check-in ; pour les locataires, au moyen d'une application mobile. Elle permet aussi d'automatiser la gestion des équipes, des clients et autres. Rental Ninja est synchronisé avec BookingSync.

Coucou&Co utilise cette plateforme pour :

- effectuer des analyses grâce au tableau de bord Rental Ninja
- et mettre une application à la disposition des clients

1.6.4. Bexio



Figure 5 : Logo Bexio

Bexio est une plateforme de gestion pour les entreprises, qui permet d'automatiser la comptabilité.

Coucou&Co utilise cette plateforme pour générer :

- factures ;
- devis ;
- offres ;
- et encaissements.

1.7. Architecture logicielle

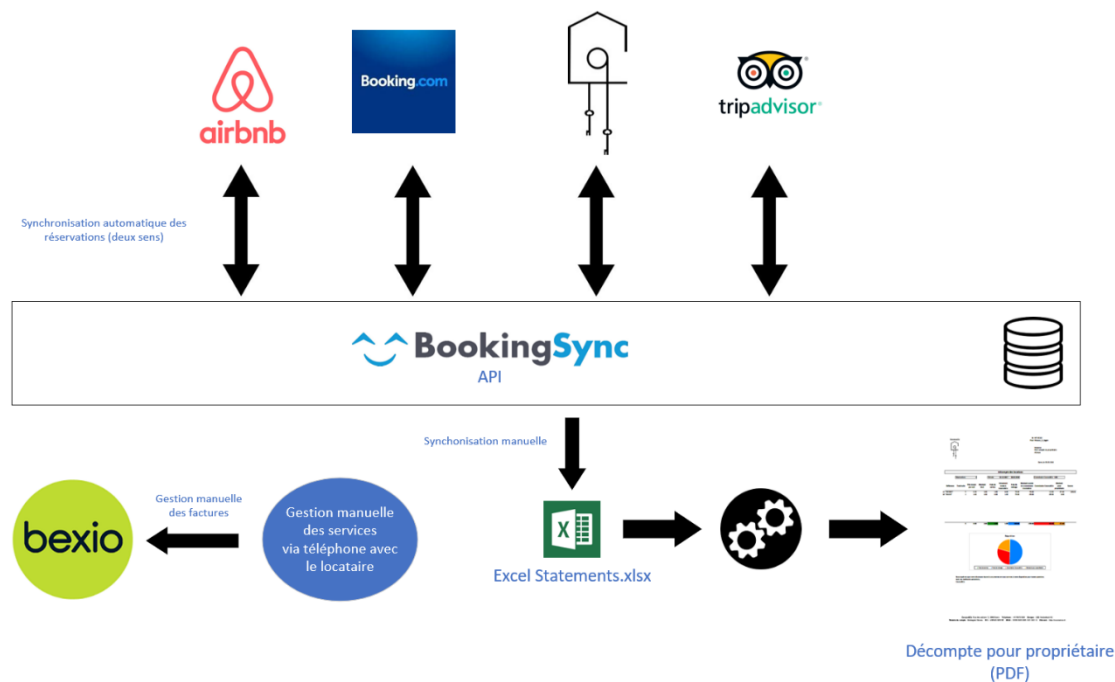


Figure 6 : Architecture logicielle actuelle de Coucou&Co

Source : Données de l'auteur

Coucou&Co poste les annonces des logements sur BookingSync, qui se chargera de les publier automatiquement sur Airbnb, la plateforme de réservation de Coucou&Co et sur TripAdvisor. En ce qui concerne Booking.com, la gestion des annonces n'est pas prise en compte et doit être faite manuellement.

La synchronisation des réservations est gérée automatiquement sur toutes les plateformes.

Les réservations sont copiées à la main dans le fichier Excel (Statements.xlsx) sur une base mensuelle. Ce même fichier contient toutes les locations gérées par l'entreprise et permet de générer un rapport pour chaque propriétaire.

Lorsqu'un client souhaite un service supplémentaire tel que le nettoyage, il téléphone directement à Coucou&Co afin de commander ledit service qui lui sera facturé séparément. La saisie des prestations additionnelles se fait manuellement dans Bexio.

1.8. Technologies

1.8.1. BookingSync

BookingSync met à disposition une API REST en JSON. Ainsi, il est possible d'implémenter facilement de nouvelles solutions à la plateforme. Et c'est aussi grâce à cette API que les différentes plateformes de réservation comme Airbnb et Tripadvisor peuvent être connectées et synchronisées.

1.8.2. Fichier Excel

Le fichier Excel est rempli selon les entités de données suivantes.

- User : utilisateur, administrateur, propriétaire, locataire, etc.
- Rental : bien en location, logement.
- Cleaning fee : frais de ménage
- Booking : réservation
- Statement : décompte des montants dus aux propriétaires par Coucou&Co

1.9. Limitations et améliorations futures

La partie gestion des réservations (booking) est automatisée à l'aide de technologies récentes et fonctionne à merveille. Il n'y a pas d'amélioration à apporter de ce côté-là.

En ce qui concerne le fichier Excel (Statements.xlsx), il peut être grandement amélioré. Le fait qu'il faille entrer manuellement les réservations dans ce fichier rend la tâche laborieuse, risquée (erreur de saisie) et coûteuse en temps.

Quant aux services, bien que primaires pour l'instant, ils seront élargis à l'avenir, car Coucou&Co souhaite développer son offre et proposer une plateforme pour la réservation de ces services. La plateforme consisterait en un espace d'échange entre entreprises souhaitant proposer un service et locataires désireux d'en profiter. À même de récupérer les données de BookingSync, elle pourrait automatiquement les entreprises des informations nécessaires concernant les logements.

Cette plateforme pourrait permettre à Coucou&Co d'améliorer son offre tout en automatisant le processus déjà en place.

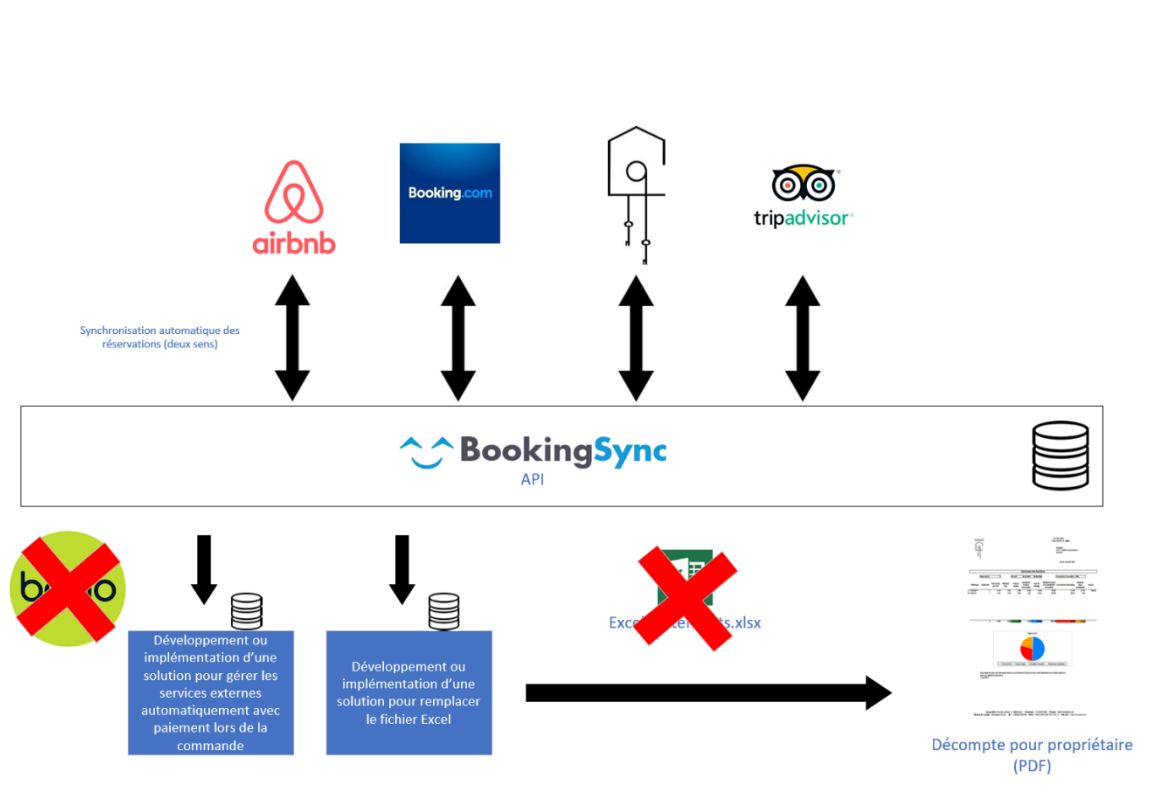


Figure 7 : Architecture logicielle future de Coucou&Co

Source : Données de l'auteur

Il serait donc primordial pour l'entreprise d'automatiser les processus en développant ou en implémentant ces deux solutions, qui remplaceraient les tâches manuelles de synchronisation.

2. Analyse des besoins

Une fois l'analyse sur la situation de l'entreprise terminée, nous avons rencontré le client pour mieux définir les besoins du projet avant de commencer l'implémentation. Après discussion et réflexion approfondies, le client souhaite une plateforme indépendante qui correspondrait à un besoin de l'entreprise ne se limitant pas à une utilisation exclusive à l'entreprise. Nous avons donc défini les principales fonctionnalités, certaines avancées, du projet.

2.1. Définition du projet

Notre objectif principal consiste en la réalisation d'une plateforme indépendante du système existant de Coucou&Co. Il a été décidé qu'un prototype basique devait être disponible à la fin du travail, prototype dotant les fournisseurs de la possibilité d'administrer services et réservations client et les clients de celle de gérer leurs propres réservations.

Les fonctionnalités touchant l'API, le système de paiement, les rôles d'administration et de gérant de PMS ainsi que le système de communication client-fournisseur ne rentrant pas dans le cadre de ce travail en raison du temps à disposition pour sa réalisation.

2.1.1. Fonctionnalités importantes

Les fonctionnalités jugées importantes par le client sont les suivantes.

- Chaque utilisateur dispose d'un login avec validation d'adresse mail.
- Les fournisseurs peuvent ajouter, modifier et supprimer des services.
- Les clients peuvent créer une commande et l'annuler.
- Les fournisseurs peuvent gérer les commandes pour le compte de leurs clients.
- Les fournisseurs peuvent gérer les champs du formulaire de réservation.

3. Etat de l'art

3.1. Solutions existantes

L'open source étant un aspect important pour le client mais également pour le développement d'une telle plateforme du fait de sa particularité et de son potentiel d'expansion, toutes les solutions retenues ici sont des solutions de ce type.

3.1.1. WooCommerce – Plugin WordPress



Figure 8 : Logo WooCommerce

WooCommerce est une extension gratuite du CMS WordPress. Elle se trouve être la solution la plus utilisée en matière de vente de produits et services. L'extension rajoute les éléments suivants qui correspondent aux besoins décrits ci-dessus :

- Pages de produits
- Panier
- Validation de commande

Cette extension promet une personnalisation du design, un développement basé sur une API REST et un accès au code de l'extension pour pouvoir l'étendre, le modifier et l'adapter à différents besoins. La documentation technique est complète et bien détaillée. Un dernier point positif réside en l'application mobile proposée, dédiée à la gestion de sa plateforme.

Malheureusement, bien que gratuite pour une utilisation très basique, l'extension comporte beaucoup de fonctionnalités payantes, telles que :

- l'acceptation d'une réservation,
- les champs de formulaire dynamiques
- et les règles tarifaires dynamiques.

Il serait possible de lier le PMS BookingSync a un site WooCommerce au travers d'une extension à développer, mais ces dernières sont soumises à validation.

3.1.2. Zen Cart



Figure 9 : Logo ZenCart

Zen Cart est un logiciel libre et gratuit sous licence GPL2. Il requiert uniquement PHP, MySQL et Apache pour fonctionner et promet une personnalisation unique, des fonctionnalités et un support de bonne qualité. Il est également complètement modifiable en fonction des besoins.

Bien que légère, la documentation technique existe, et le support, accessible par le biais de leur forum, semble correct. En ce qui concerne la vente de services, un plugin permet la vente de bons et pourrait correspondre à nos besoins, or ce système peut être très limitant.

La liaison au PMS devra être créée de toute pièce et peut s'avérer compliquée au vu des fonctionnalités présentées dans la documentation.

3.1.3. Magento



Figure 10 : Logo Magento

Magento est une plateforme open source développée par Magento Inc., société fille de Adobe, qui permet de créer des sites marchands. Le code source est disponible sur GitHub mais il n'y a pas de développement libre pour les modules spécifiques aux besoins de notre projet, la communauté étant principalement active dans le développement du programme.

Magento Inc., développeur du programme, décide des améliorations qui seront intégrées aux prochaines versions. Propulsée en PHP, la plateforme utilise une base de données MySQL.

Lorsque l'on désire des fonctionnalités adaptées au développement d'un projet de l'ampleur du nôtre, les extensions et thèmes proposés se chiffrent en centaines de dollars, ce qui n'est pas négligeable.

3.1.4. Shopify



Figure 11 : Logo Shopify

Shopify est une plateforme qui permet de créer une boutique en ligne simplement, sans connaissances techniques. Elle permet également de promouvoir les produits sur les réseaux sociaux. Le Shopify app store propose des extensions de personnalisation pouvant être payantes ainsi qu'une API. Il serait donc possible d'utiliser les extensions présentes et de développer les fonctionnalités manquantes via l'API Shopify

Le prix de ce service est de 29\$ par mois, prix justifié par une solution SaaS (Software as a Service). En d'autres termes, nul besoin pour nous d'héberger le site. Cependant, cette configuration ne nous permet pas de modifier la plateforme selon nos besoins et une application indépendante, qui utilise l'API Shopify, doit être créée à cette fin.

3.1.5. Conclusion

Dans le tableau ci-dessous, nous comparons les différentes plateformes analysées au paragraphes précédents selon quatre critères essentiels pour mener à bien notre projet. Il s'agit de la possibilité de vendre un service, de la disponibilité d'une documentation technique pour le développement de modules ou de fonctionnalités personnalisés, du prix du service et de ses extensions, et de la connexion avec BookingSync, PMS de Coucou&Co. Chaque plateforme se verra attribuer une note de 1 à 4 pour chaque critère, 4 étant la note la plus élevée.

	WooCommerce	ZenCart	Magento	Shopify
Vente de services	4	2	2	2
Documentation technique	4	3	3	3
Prix	1	4	1	2
Liaison au PMS	2	1	1	1
Total	11	10	7	8

Tableau 1 : Comparaison des solutions du marché

Des quatre solutions présélectionnées WooCommerce et ZenCart sont celles qui se rapprochent le plus des besoins du projet. C'est donc celles-là mêmes qui seront retenues pour que nous puissions effectuer une comparaison avec une solution « maison ».

3.2. Technologies

Les différentes solutions disponibles n'étant pas suffisamment satisfaisantes pour la plateforme désirée, nous décidons d'analyser les technologies existantes pour éviter un développement complet. Utiliser un framework facilitant le développement de fonctionnalités basiques, nous permettra de gagner du temps. Un framework est un ensemble d'outils et de bibliothèques qui simplifient le développement des fonctionnalités fréquemment utilisées.

Aussi, nous analyserons trois frameworks open source. Actuellement, il en existe une multitude mêlant performance et modularité et apportant ainsi un gain de temps et d'effort non négligeable, il ne serait donc pas judicieux de s'en passer alors qu'une solution « from scratch » n'est pas justifiée.

3.2.1. Django – Framework Python



Figure 12 : Logo Django

Django est un framework Python développé il y a plus de vingt ans par une agence de presse. Conçu pour créer des sites internet de qualité en un temps record, il possède une communauté importante et active qui partage beaucoup de code personnalisé.

Le framework nécessite Python 3.4 ou supérieur, et s'installe globalement sur la machine. Il se base sur le modèle « Model-View-Controller » et possède son propre ORM (système de gestion des relations objets ou Object-Relational Mapping).

3.2.2. Express – Framework JavaScript



Figure 13 : Logo Express

Express est un framework orienté web et basé sur Node.js, une plateforme développée en JavaScript qui permet l'exécution de ce dernier du côté serveur. Il contient une multitude de fonctionnalités de base destinée aux applications web. Offrant une installation très simple, il se trouve dans les dépôts de paquets de Npm, le gestionnaire de paquets officiel de Node.js.

Le framework en se veut minimaliste et est donc complètement personnalisable grâce aux modules développés par des tiers. La communauté de Node.js est très active et propose de nombreuses ressources. La documentation d'Express est fournie et claire, ce qui n'est malheureusement pas le cas pour tous les modules Node.js que nous allons potentiellement être amenés à utiliser.

3.2.3. Laravel – Framework PHP

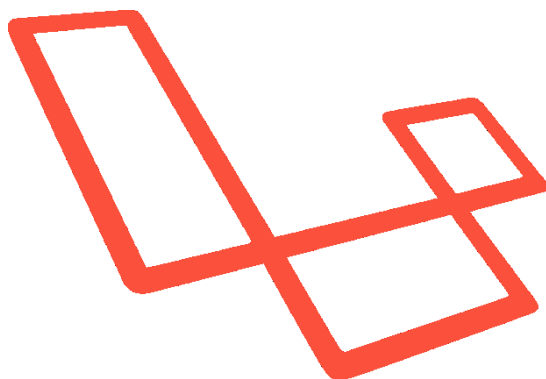


Figure 14 : Logo Laravel

Laravel est un framework PHP basé sur Symfony2. Régulièrement mis à jour, il possède une communauté conséquente. Avec plus de 50 000 « Star », il est le projet le plus suivi du

thème « PHP » sur GitHub. C'est un framework complexe qui contient une multitude de fonctionnalités en vue d'un développement d'applications web grandement facilité.

De sa complexité découle une difficulté de prise en main pour un utilisateur sans expérience avec Laravel mais sa documentation est détaillée et précise. Elle est étayée d'exemples pour chaque élément du framework. Des packages communautaires sont recensés sur différents sites et téléchargeables depuis Npm ou Composer.

3.2.4. Conclusion

Nous allons juger ces trois frameworks sur la base de leur domaine d'utilisation, de la simplicité de leur prise en main, de nos connaissances préalables, de leur testabilité, de leur facilité d'hébergement et de leur documentation. L'échelle est la même que pour l'analyse des solutions existantes (1 à 4).

	Django	Express	Laravel
Web app de grande ampleur	3	1	4
Simplicité	3	3	2
Connaissances	1	3	2
Testabilité	3	2	4
Hébergement	2	4	3
Documentation	3	3	4
Total	15	19	19

Tableau 2 : Comparaison des frameworks

Les frameworks Express et Laravel sont donc les deux frameworks qui ressortent de cette comparaison.

3.3. Choix final

Nous avons donc quatre possibilités retenues jusqu'à présent (deux plateformes et deux frameworks), qui sont WooCommerce, ZenCart, Laravel et Express. De ces quatre éléments nous éliminons les deux solutions existantes car les de fonctionnalités offertes ne contrebalancent pas la difficulté d'adaptation engendrée par les besoins du projet. Grâce aux frameworks nous évitons une perte de temps concernant les fonctionnalités basiques d'une application web et, ainsi, nous pouvons nous permettre une flexibilité complète quant aux pages et tableaux de bord utilisateurs.

Basé sur Symphonym le framework Laravel est celui qui correspond le mieux à nos besoins. Il possède une documentation fournie et claire, de nombreuses fonctionnalités permettant de simplifier notamment la gestion de la base de données, la gestion d'utilisateurs et de rôles, et l'authentification des requêtes. Comparé à Express, il apparaît comme plus performant pour des projets de grande ampleur. Doté d'une communauté active et un développement continu il est connu de l'entreprise Coucou&Co, ce qui permettra à cette dernière de reprendre le projet aisément par la suite. De plus, le travail de Bachelor parallèle, mandaté par la même entreprise, repose également sur ce choix. Par conséquent, il convient d'utiliser le même framework pour les deux projets similaires afin de faciliter leur intégration dans le système de l'entreprise.

4. Implémentation

4.1. Outils

4.1.1. JetBrains IDE – PhpStorm



Figure 15 : Logo PhpStorm

PhpStorm est un IDE (Environnement de développement) performant développé par JetBrains pour le langage PHP. Il comprend une intégration des outils Git ainsi qu'un dépôt d'extensions, dont l'extension pour Laravel. C'est l'outil que nous avons décidé d'utiliser pour le développement.

4.1.2. GitHub

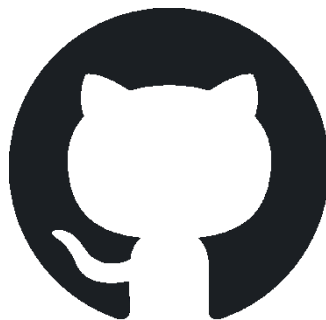


Figure 16 : Logo GitHub

GitHub est un gestionnaire de code source très répandu. Un gestionnaire est indispensable lors du développement d'un tel projet. Pour une utilisation open source il est gratuit.

4.1.3. PHPUnit

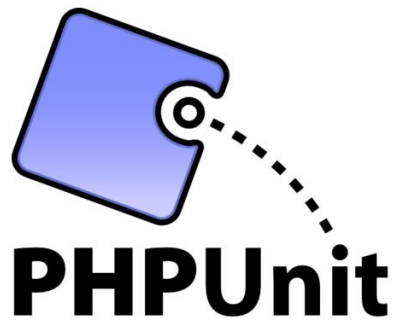


Figure 17 : Logo PHPUnit

PHPUnit est utilisé pour réaliser les tests unitaires, il s'intègre parfaitement à Laravel.

4.1.4. Laravel Dusk



Figure 18 : Logo Laravel Dusk

Laravel Dusk est un logiciel développé par Laravel pour les tests navigateur. Il utilise une version standalone de ChromeDriver, ce qui facilite son utilisation.

4.1.5. Travis CI



Figure 19 : Logo Travis CI

Travis CI est un logiciel SaaS d'intégration continue existant en version gratuite si l'on possède un compte GitHub Scolaire. Il suffit de lier le projet GitHub à Travis et de configurer l'exécution automatique des builds.

4.1.6. Apache2



Figure 20 : Logo Apache2

Apache2 est le serveur web utilisé dans l'environnement de développement. Très simple à configurer pour un environnement de développement, il a été choisi par défaut pour ce projet.

4.1.7. MariaDb



Figure 21 : Logo MariaDB

Les bases de données NoSQL n'étant pas adaptées pour des projets nécessitant une structure complexe, c'est naturellement une base de données relationnelle qui a été choisie pour cette application. MariaDB est un serveur de base de données open source populaire.

4.2. Méthodologie

Ce travail a été réalisé selon une méthodologie Agile, fortement inspirée de Scrum. Le projet s'est déroulé en 4 sprints (cycles de développement) de 8 à 10 jours. Un rendez-vous avec le client était prévu entre chaque sprint pour démarrer et clore ces derniers. Lors de ces rendez-vous, les user stories (fonctionnalités à implémenter) du sprint se terminant étaient

présentées au client, afin que nous puissions décider ensemble des user stories du sprint suivant.

Plusieurs critères comme la prise en main du framework et le temps à disposition sont à l'origine des variations du nombre de story points par sprint. Le document Excel disponible dans la clé USB « TB_02_HuserGaetan_ProductBacklog.xlsx », regroupe toutes les informations quant à la gestion du projet.

4.2.1. Sprint 0

Le sprint 0 a été dédié aux différentes recherches et analyses ainsi qu'à la mise en place de l'environnement de développement.

La figure suivante indique les users stories du sprint en question et son état d'avancement au fil des semaines que nous avons à notre disposition.

ID US.task	Task Name	Initial Estimate	W 18	W 19	W 20	W 21	W 22
dev_01-1	Définir les recherches à effectuer	1	0	0	0	0	0
dev_02-1	Recherche de plateformes de vente	2	1	0	0	0	0
dev_02-2	Recherche de gestions de roles	2	1	0	0	0	0
dev_02-3	Recherche de gestionnaire de notifications	2	2	0	0	0	0
dev_03-1	Recherche des possibles technologies	1	1	0	0	0	0
dev_03-2	Recherche de frameworks web	2	2	0	0	0	0
dev_04-1	Définir les meilleures technologies à utiliser from scratch	1	1	1	0	0	0
dev_04-2	Définir les meilleures technologies à utiliser avec un framework	1	1	1	0	0	0
dev_04-3	Comparer les solutions entre elles	2	2	2	0	0	0
dev_04-4	Comparer les solutions avec le TB parallèle	1	1	1	0	0	0
dev_05-1	Installation des technologies / IDE	2	2	2	2	0	0
dev_06-1	Création de la DB locale	2	2	2	1	0	0
dev_07-1	Créer un mockup des écrans de base	5	5	5	5	0	0
dev_08-1	Création du projet de base	1	1	1	1	1	0
dev_09-1	Choix de la technologie CI	1	1	1	1	1	0
dev_09-2	Configuration du CI	2	2	2	2	2	0
dev_10-1	Déploiement du Projet sur le serveur	1	1	1	1	1	0
1,1	Créer un login	1	1	1	1	1	0

Figure 22 : Déroulement du sprint 0

Les estimations de tâches permettent de prévoir le nombre de fonctionnalités implémentables lors du sprint. Nous pouvons également suivre la vélocité du sprint grâce à un graphique comportant le nombre de jours et le nombre d'heures estimées.

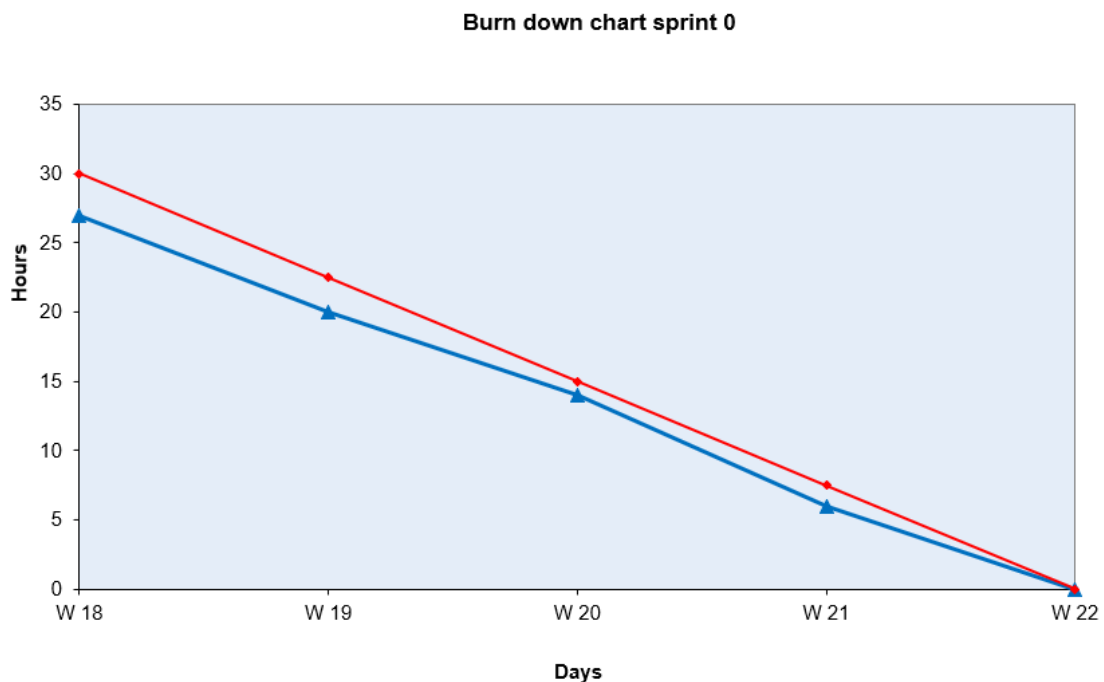


Figure 23 : Burndown Chart du sprint 0

Le suivi rouge étant l'avancement selon un partage parfait des tâches en fonction de chaque entrée et le suivi bleu l'avancement réel, nous pouvons voir ici que le travail a avancé de manière régulière tout au long du sprint. Ces figures sont des captures d'écran de la feuille « Sprint 0 » du fichier Excel « TB_02_HuserGaetan_ProductBacklog.xlsx ». La correspondance de chaque sprint peut y être retrouvée à la suite de cette dernière.

4.2.2. Sprint 1

Le sprint 1 a été consacré à la gestion des rôles utilisateur ainsi que la création de services basiques. Il s'est déroulé sur 8 jours, répartis sur 3 semaines.

4.2.3. Sprint 2

Le sprint 2 a été consacré à la gestion des réservations et le panneau de gestion du fournisseur. Il s'est déroulé sur 10 jours, répartis sur 2 semaines.

4.2.4. Sprint 3

Le sprint 3 a été consacré à la gestion dynamique des formulaires de réservation et la création de notifications par email. Il s'est déroulé sur 10 jours, répartis sur 2 semaines.

4.3. Fonctionnalités

4.3.1. Visiteur

Lorsqu'un visiteur se rend sur le site, il accède à la liste des services disponibles.

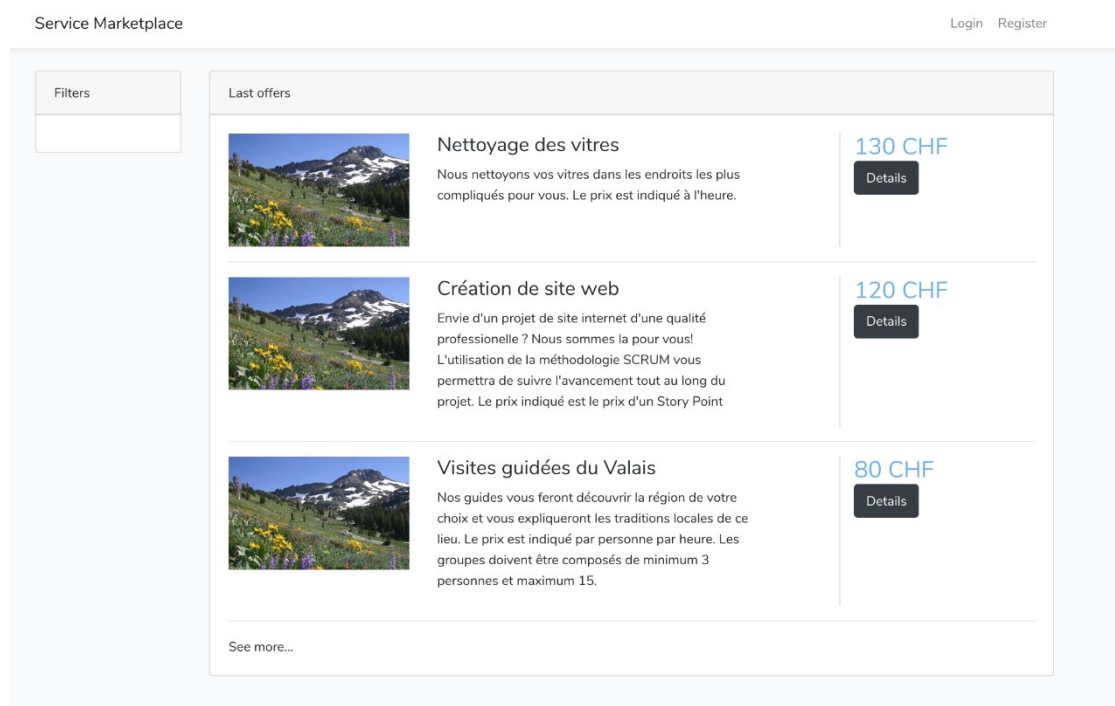


Figure 24 : Page d'accueil du site

Il peut afficher les détails de chaque service, mais n'a pas accès au formulaire de commande.



Nettoyage des vitres

130 CHF

Description

Nous nettoions vos vitres dans les endroits les plus compliqués pour vous. Le prix est indiqué à l'heure.

Please login to place an order

Figure 25 : Détail d'un service pour un visiteur

Il peut s'enregistrer ou se connecter grâce au liens situés dans la barre de navigation.

Register

Name

E-Mail Address

Password

Confirm Password

Street and Number

Zip / Town

Country

Figure 26 : Formulaire d'enregistrement

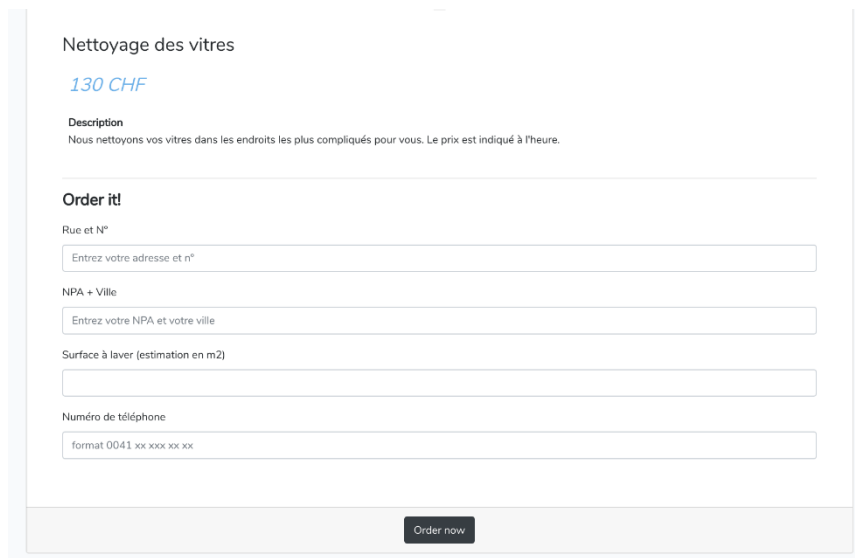
À l'enregistrement, il recevra un email pour confirmer son adresse de messagerie et valider son compte.

#	id	name	email	email_verified_at	password	street	npa	town	country	remember_token	created_at	updated_at
1	1	Mike Horn	client@example.com	---	\$2y\$10\$ZSqHoy3Vs7...	Bombay Street 77	3960	Sierre	Suisse	---	2019-07-29 20:25:04	2019-07-29 20:25:04
2	2	Carvitre SA	servprov@example.com	---	\$2y\$10\$ymrDwZNAol...	Rue de l'OURS 4	1950	Sion	Suisse	---	2019-07-29 20:25:04	2019-07-29 20:25:04
3	3	Gaëtan Huser	huser.gaetan@hotmail...	2019-07-22 10:43:01	\$2y\$10\$eg31pvQYaX...	Rue de la Censure 42	1963	Vétroz	Suisse	---	2019-07-21 20:25:04	2019-07-22 10:43:01

Figure 27 : Champ de vérification d'email utilisateur dans la base de donnée

4.3.2. Client

Une fois l'utilisateur connecté, le formulaire de commande est disponible dans le détail d'un service.



Nettoyage des vitres

130 CHF

Description
Nous nettoyons vos vitres dans les endroits les plus compliqués pour vous. Le prix est indiqué à l'heure.

Order it!

Rue et N°

NPA + Ville

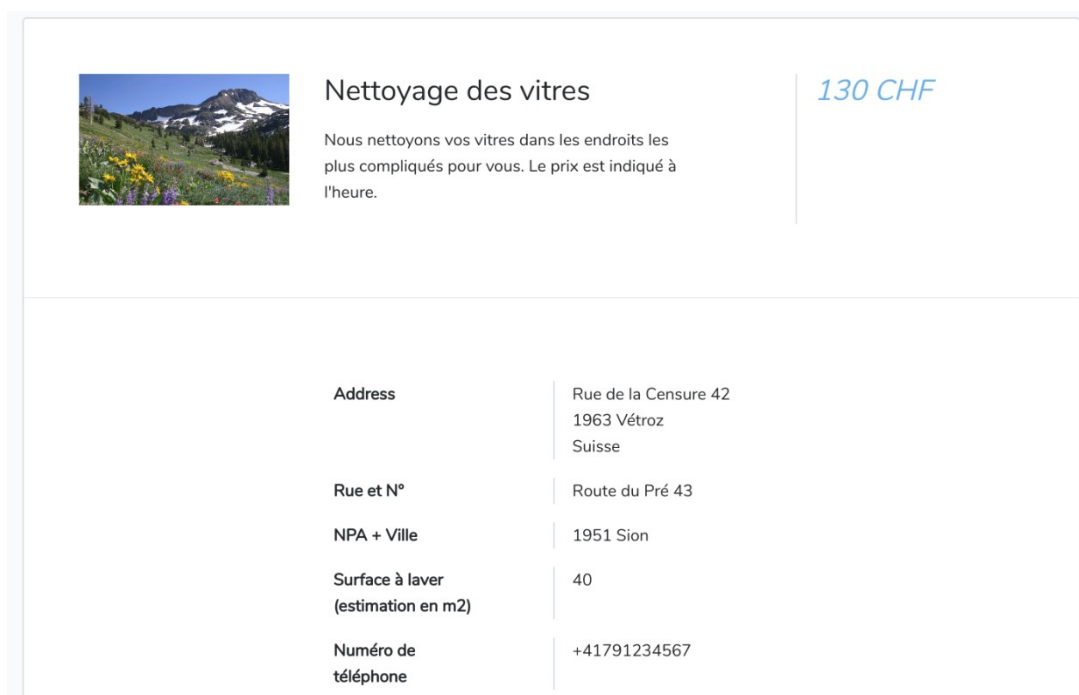
Surface à laver (estimation en m2)


Numéro de téléphone

[Order now](#)

Figure 28 : Détail d'un service pour un client

Lorsqu'un client commande un service il est redirigé vers les informations de sa commande et reçoit un email de confirmation. Le fournisseur de service reçoit à son tour un email de notification.



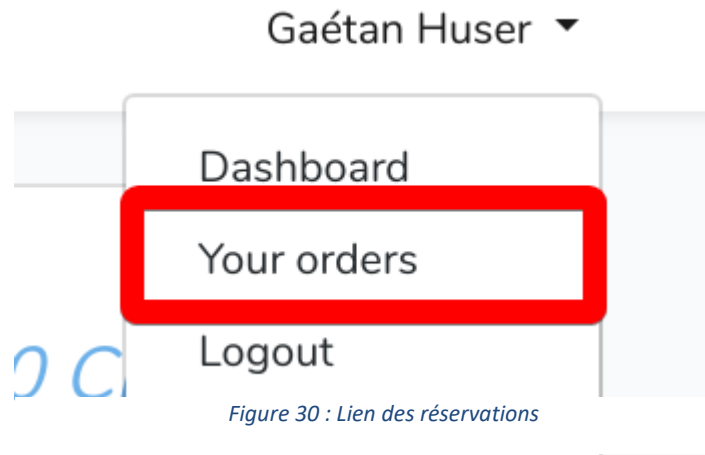
 **Nettoyage des vitres** **130 CHF**

Nous nettoyons vos vitres dans les endroits les plus compliqués pour vous. Le prix est indiqué à l'heure.

Address	Rue de la Censure 42 1963 Vétroz Suisse
Rue et N°	Route du Pré 43
NPA + Ville	1951 Sion
Surface à laver (estimation en m2)	40
Numéro de téléphone	+41791234567

Figure 29 : Confirmation de réservation

Il peut avoir un aperçu de ses commandes en cliquant sur le lien du menu de la barre de navigation.



Pour annuler une commande, il suffit de cliquer sur le bouton « Cancel » correspondant à la commande en question. Un modal de confirmation s'affiche pour éviter les annulations erronées.

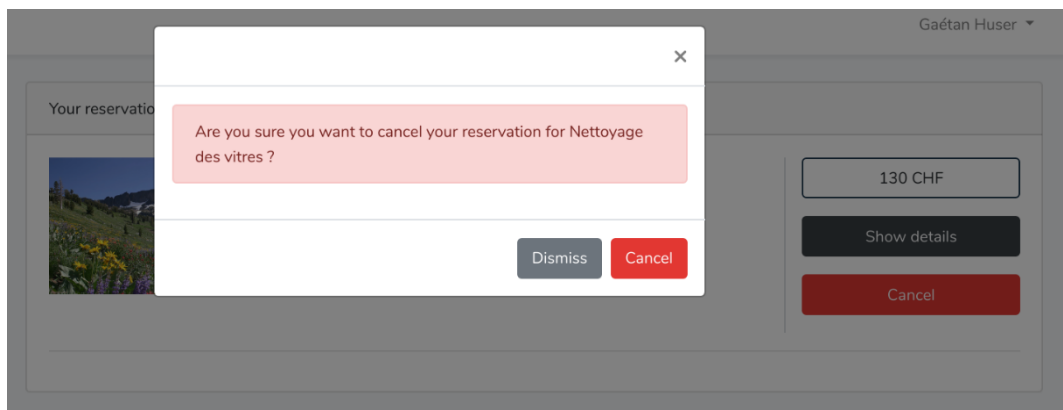


Figure 31 : Annulation d'une commande

4.3.3. Service Provider

Un client enregistré comme fournisseur de service à accès au panel de gestion grâce au menu.

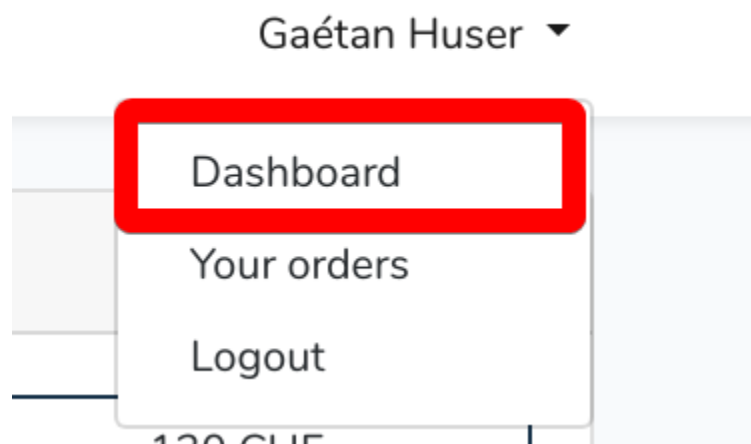


Figure 32 : Lien du panel de gestion du fournisseur de service

Il dispose alors d'un résumé global contenant les derniers service créé et commande client passée avec la possibilité d'ajouter d'autres informations.

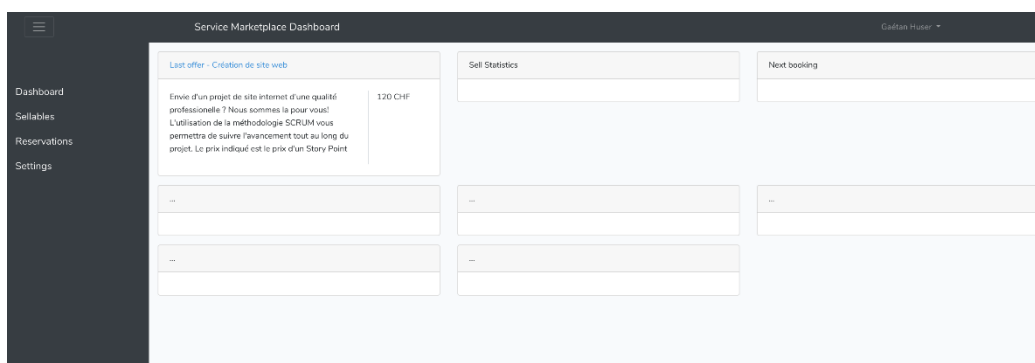


Figure 33 : Panel de gestion du fournisseur de service

Le menu latéral lui permet d'avoir accès à ses services et commandes.

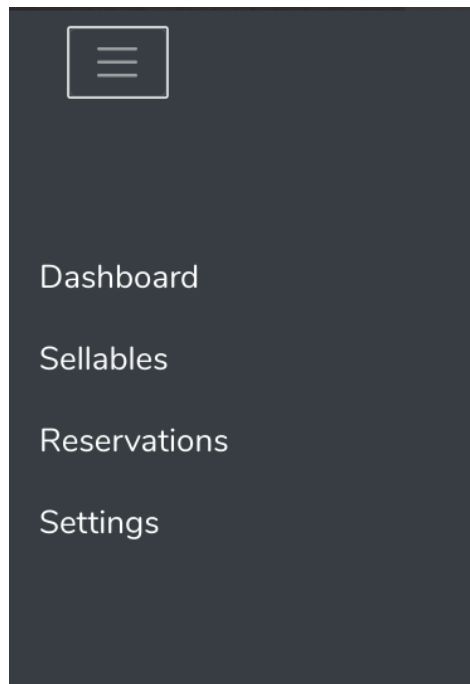


Figure 34 : Menu latéral

La page des services lui permet de gérer ses prestations et d'en ajouter de nouvelles.

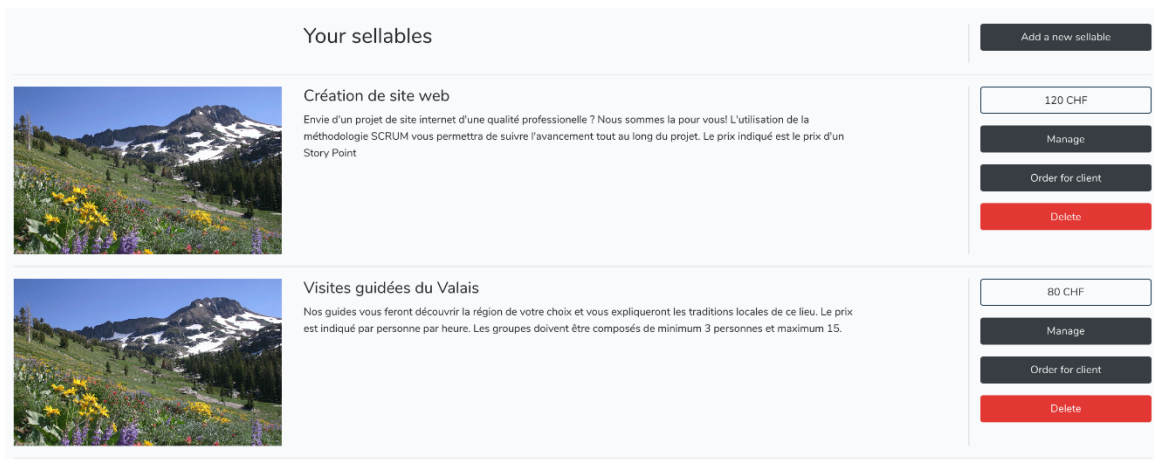


Figure 35 : Liste des services du fournisseur

Lorsqu'il souhaite créer un service il doit remplir les champs obligatoires et peut en ajouter de personnalisés.

Add a new sellable

Sellable image
Choisir un fichier WallpaperWindow.jpg

Name
Visite du musée de l'horreur

Price
30

Description
Venez visiter le musée de l'horreur.
Durée de la visite : 50 minutes
Prix par personnes.

Merci de nous indiquer deux choix d'heure de visite.

Add personalized fields
+

Figure 36 : Formulaire d'ajout d'un service

Une série de types de champs personnalisés peuvent être ajoutés et lui permettent de compléter son formulaire de commande en fonction des informations dont il requiert pour offrir son service.

Choose a field type

Text field

Date field

Number field

Text Area

Time field

Dismiss

Figure 37 : Liste des champs personnalisés

Dans l'exemple ci-dessous, nous avons limité le nombre maximal de personnes à vingt, la plage horaire entre 08 :00 et 16 :00 et nous avons également bloqué le choix des minutes. 3600 secondes étant égal à une heure.

Add personalized fields

Number field

Insert a value to put a default value, let the field empty if not needed

name	Nombre de personnes
value	
placeholder	
min	1
max	20
step	

Date field

Insert a value to put a default value, let the field empty if not needed

name	Date de la visite
value	
placeholder	
min	
max	

Time field

Insert a value to put a default value, let the field empty if not needed

name	Choix horaire n°1
value	
placeholder	
min	08:00
max	16:00
step	3600

Time field

Insert a value to put a default value, let the field empty if not needed

name	Choix horaire n°2
------	-------------------

Figure 38 : Champs personnalisés ajoutés au service

La page des réservations donne un aperçu des commandes, triées de la plus récente à la plus ancienne. Il peut gérer chaque réservation pour le client, l'annuler ou la supprimer. Une commande annulée s'affiche comme telle dans la liste.

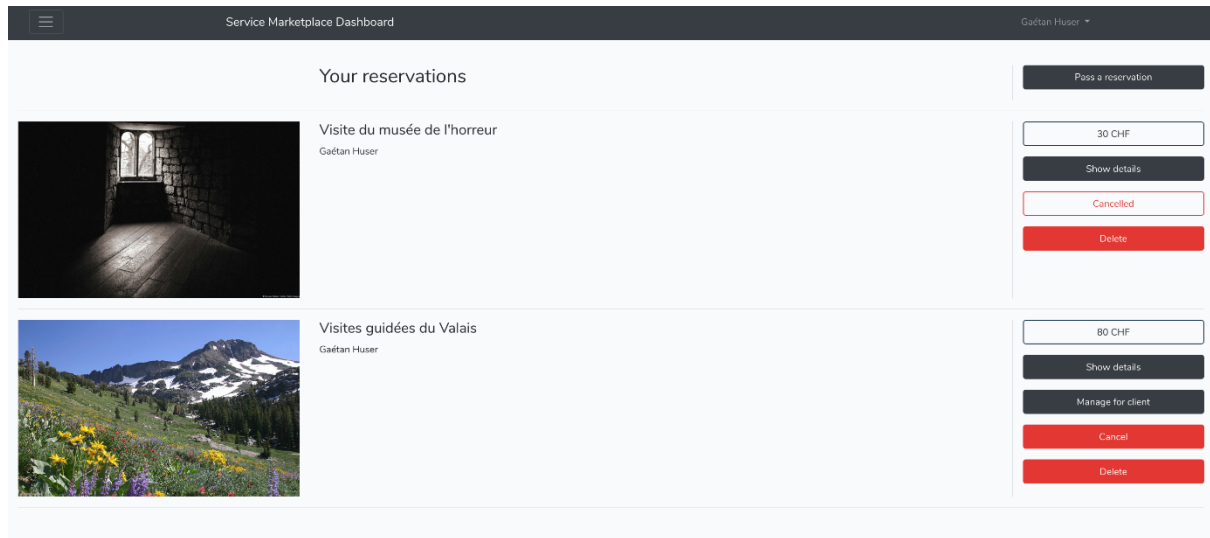


Figure 39 : Liste des réservations du fournisseur

Il peut également passer une réservation pour un client. Un modal s'affiche pour la sélection du service concerné.

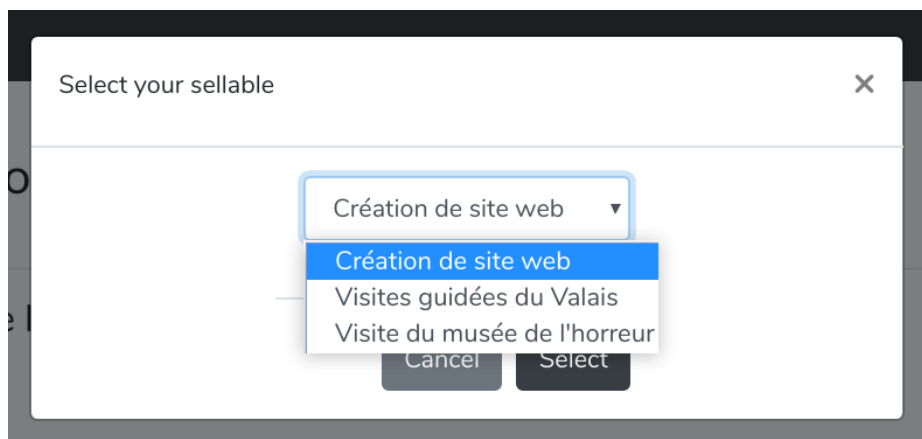


Figure 40 : Choix du service à réserver

Il se retrouve alors avec le même formulaire que le client, avec un champ supplémentaire, celui de l'adresse email du client pour lequel il passe la réservation. Nous retrouvons ici les minutes qui ne sont pas modifiables, car bloquées à l'ajout du service.

The screenshot shows a reservation form for 'Visite du musée de l'horreur'. At the top, it says 'Selected sellable: Visite du musée de l'horreur'. Below this, there is a 'Client Mail' field with the email 'huser.gaetan@hotmail.com'. The 'Nombre de personnes' field is set to '5'. The 'Date de la visite' field is set to '13/11/2019'. There are two time selection fields: 'Choix horaire n°1' set to '11:00' and 'Choix horaire n°2' set to '14:00'. At the bottom right, there is a 'Place order' button.

Figure 41 : Formulaire de réservation d'un fournisseur pour un client

Le client reçoit alors un email de confirmation de commande et le fournisseur un email de notification.

The screenshot shows an email notification for a reservation. The header reads 'Gaétan Huser has reserved Visite du musée de l'horreur.'. The main content is a list of reservation details for 'Client Gaétan Huser':

E-mail	huser.gaetan@hotmail.com
Address	Rue de la Censure 42 1963 Vétroz Suisse
Nombre de personnes	5
Date de la visite	2019-11-13
Choix horaire n°1	11:00
Choix horaire n°2	14:00

At the bottom, there is a copyright notice: '© 2019 Service Marketplace.'

Figure 42 : Exemple d'email de notification envoyé au fournisseur

4.4. Difficultés

4.4.1. Réservation pour un client

Notre mandataire souhaite que le fournisseur de services puisse enregistrer ses propres clients sur la plateforme et que ce dernier reçoive un mail concernant cette réservation. Nous utilisons l'adresse email pour contrôler si le client existe déjà sur la plateforme. Les informations de la réservation sont enregistrées et la commande est passée par le fournisseur.

Les fournisseurs ont accès à la gestion des commandes de leurs clients. Ils peuvent modifier des valeurs, annuler la commande et la supprimer définitivement.

4.4.2. Formulaire dynamiques

Le plus gros défi de ce travail est sans doute la gestion dynamique des formulaires de réservation. La plateforme étant censée proposer tous types de services au travers de fournisseurs, il est essentiel que ceux-ci aient une liberté au niveau des informations concernant le client. Il peut s'agir d'une date, d'une durée, d'un nombre de personnes ou encore ou encore d'une surface habitable – les possibilités ne manquent pas.

Nous avons donc tout d'abord créé une classe générique pour les types de champs de formulaires. Cette classe contient un nom, un type de champ et un tableau d'attributs, qui serviront à leur création. La figure suivante montre le code de cette classe.

```
InputType.php ×
1 <?php
2
3
4 namespace App\DynamicInputTypes;
5
6
7 class InputType
8 {
9     public $name;
10
11     public $input_type;
12
13     protected $attributes = [];
14
15     public function getAttributes(){
16         return $this->attributes;
17     }
18
19     public function clearDefaultAttributes() {
20         $this->attributes = [];
21     }
22
23     public function setAttribute($name, $value) {
24         $this->attributes[$name] = $value;
25     }
26
27 }
```

Figure 43 : Classe générique "InputType.php"

Les classes de champs étendent toutes la classe principale « InputTypes.php » afin d'utiliser la même structure. Dans le tableau d'attributs nous retrouverons les informations nécessaires au type de champ. Ici nous avons l'exemple du type de champ « date ».

```

DateInput.php x
1  <?php
2
3
4  namespace App\DynamicInputTypes;
5
6
7  class DateInput extends InputType
8  {
9      function __construct() {
10         $this->name = "Date field";
11         $this->input_type = "date";
12         $this->attributes = [
13             "name" => "The name of the field",
14             "value" => "Default value for the field",
15             "placeholder" => "Display help text in the field if no value",
16             "min" => "The minimal date valid formatted as yyyy-MM-dd",
17             "max" => "The maximal date valid formatted as yyyy-MM-dd"
18         ];
19     }
20 }
21
22 }
```

Figure 44 : Classe "DateInput.php"

Nous avons créé un fichier de configuration contenant les noms de classes pour pouvoir retrouver les types disponibles ainsi qu'instancier les classes en question.

```

dynamicInputTypes.php x
1  <?php
2
3  return [
4
5      'TextInput' => '\App\DynamicInputTypes\TextInput',
6      'DateInput' => '\App\DynamicInputTypes\DateInput',
7      'NumberInput' => '\App\DynamicInputTypes\NumberInput',
8      'TextAreaInput' => '\App\DynamicInputTypes\TextAreaInput',
9      'TimeInput' => '\App\DynamicInputTypes\TimeInput',
10 ];
```

Figure 45 : Fichier de configuration "DynamicInputTypes.php"

Puis, ces classes sont utilisées pour générer le html qui affichera les champs du formulaire de création de service où le fournisseur entrera les informations qu'il souhaite demander au client et qu'il personnalisera à sa guise.

```
DynamicInputTypesController.php x
...
}
}

public static function getDynamicInputTypesHTMLFormAdd() {
    $json = ['inputs' => []];

    foreach (config( key: 'dynamicInputTypes') as $typeClass) {

        $type = new $typeClass();

        $html = '
        <hr>
        <div class="form-group border-dark">
        <p class="mb-0">'. $type->name. '</p>
        <small>Insert a value to put a default value, let the field empty if not needed</small>
        <div class="form-group p-2 dynamic-field-group">
        <input type="hidden" name="field" value="'. $typeClass . '" class="field">
        ';

        foreach($type->getAttributes() as $name => $attribute) {
            $html .= '
            <div class="row pb-1" data-toggle="tooltip" data-placement="top" title="'. $attribute .'">
            <label class="col-4 dynamic-field-label" for="'. str_replace( search: ' ', replace: '_', $type->name) . '-' . $name.'">'. $name. '</label>
            <input class="col-8 dynamic-field" type="text" id="'. str_replace( search: ' ', replace: '_', $type->name) . '-' . $name.'" name="'. $name.'">
            </div>
            ';
        }

        $html .= '
        </div>
        </div>
        ';

        $json['inputs'][$type->name] = $html;
    }

    return $json;
}
```

Figure 46 : Méthode de génération des champs personnalisables

Une fois le service enregistré, la page de commande du service contient les champs définis par le fournisseur ainsi que les attributs de ces champs. S'il souhaite ne pas valider le service en dessous d'un nombre minimal de personnes, le client ne pourra pas choisir une valeur en dessous de ce nombre. Ces champs sont ensuite réutilisés pour afficher la réservation au travers de la méthode suivante.

```
DynamicInputTypesController.php x
class DynamicInputTypesController extends Controller
{
    public static function getDynamicInputTypesHTMLFormFill($fieldable) {
        $json = [];

        foreach ($fieldable->fields as $inputType) {

            $textAreaType = config(key: 'dynamicInputTypes.TextAreaInput');
            $textAreaInput = new $textAreaType();

            if ($inputType->input_type == $textAreaInput->input_type) {
                $html = '
                    <div class="form-group">
                        <label for="' . str_replace(search: ' ', replace: '_', $inputType->name) . '"> . $inputType->name . '</label>
                        <textarea class="form-control" id="' . str_replace(search: ' ', replace: '_', $inputType->name) . '"
                            name="' . str_replace(search: ' ', replace: '_', $inputType->name) . '"
                            aria-describedby="' . str_replace(search: ' ', replace: '_', $inputType->name) . '-Help" ' . $inputType->attributes . '>' . $inputType->value . '</textarea>
                    </div>
                ';
            }
            else {
                $html = '
                    <div class="form-group">
                        <label for="' . str_replace(search: ' ', replace: '_', $inputType->name) . '"> . $inputType->name . '</label>
                        <input class="form-control" type="' . $inputType->input_type . '" id="' . str_replace(search: ' ', replace: '_', $inputType->name) . '"
                            name="' . str_replace(search: ' ', replace: '_', $inputType->name) . '" ' . $inputType->attributes . ' value="' . $inputType->value . '">
                    </div>
                ';
            }

            $json[$inputType->name] = $html;
        }

        return $json;
    }
}
```

Figure 47 : Méthode de génération du formulaire de réservation

En utilisant le polymorphisme d'Eloquent, l'ORM utilisé par Laravel, nous pouvons utiliser les champs dynamiques pour les lier à un service et également à une réservation. Ainsi, nous ne gardons pas seulement les valeurs mais aussi le nom du champ dans la réservation, au cas où le service venait à être supprimé par le fournisseur.

```
14     'name', 'description', 'price', 'image', 'type',
15 ];
16
17 public function owner() {
18     return $this->belongsTo(related: User::class);
19 }
20
21 public function fields() {
22     return $this->MorphMany(related: SellableField::class, name: 'fieldable');
23 }
24
25 public function orders() {
26     return $this->hasMany(related: Order::class);
27 }
28
29 \App\Models > Sellable
30
31 Order.php x
32 protected $fillable = [
33     'price', 'cancelled_at'
34 ];
35
36 public function sellable(){
37     return $this->belongsTo(related: Sellable::class);
38 }
39
40 public function client() {
41     return $this->belongsTo(related: User::class);
42 }
43
44 public function fields() {
45     return $this->MorphMany(related: SellableField::class, name: 'fieldable');
46 }
47
48 }
49
50 \App\Models > Order
51
52 SellableField.php x
53 <?php
54 namespace App\Models;
55 use Illuminate\Database\Eloquent\Model;
56
57 class SellableField extends Model
58 {
59     protected $fillable = [
60         'name', 'description', 'input_type', 'attributes', 'value'
61     ];
62
63     public function fieldable() {
64         return $this->morphTo();
65     }
66 }
67
68 \App\Models > SellableField
```

Figure 48 : Modèles Eloquent "Sellable.php", "Order.php et "SellableField.php"

4.5. Améliorations futures possibles

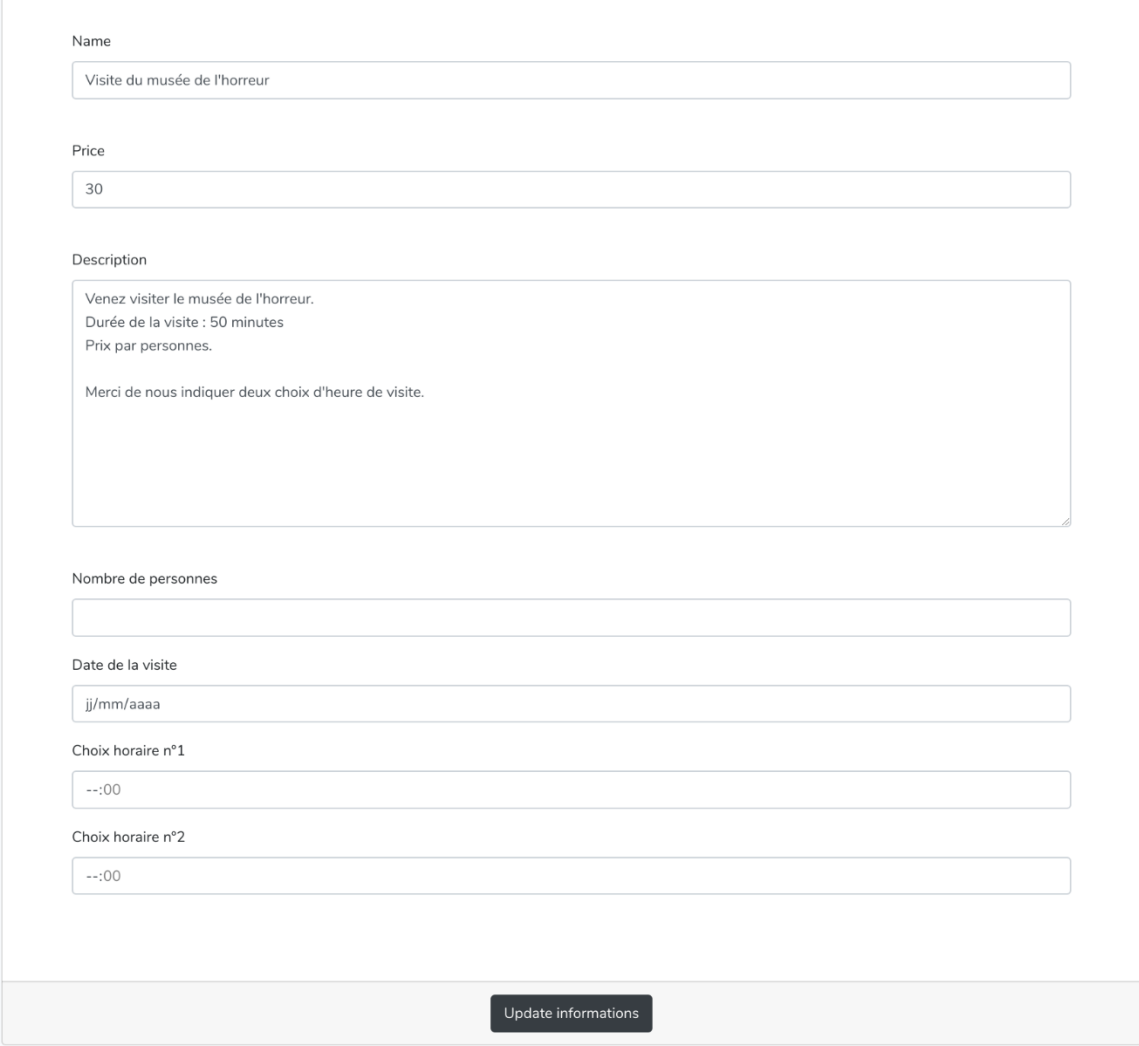
4.5.1. Champs dynamiques

Les champs dynamiques actuels ceux d'un formulaire basique. Nous avons intégré le champ « textarea », qui n'est pas un champ implanté avec la balise HTML « input », pour pouvoir implémenter la solution en prenant en compte les différentes balises. Un champ intéressant à implanter par la suite serait le champ « select », qui correspond à une liste déroulante. Ce champ permettrait à un fournisseur de service de prédéfinir une liste pour ne pas donner la possibilité au client de rentrer une valeur qui ne correspondrait pas.

Par exemple, si un guide touristique souhaite proposer un service de visites guidées, il n'a pas besoin de créer un service pour chaque lieu de visite, mais il peut proposer une liste déroulante avec la liste des visites disponibles. Le client peut de son côté sélectionner l'activité de son choix.

4.5.2. Modification poussée des services

Actuellement les champs dynamiques des services ne peuvent pas être supprimés ou ajoutés une fois le service créé.



The screenshot shows a form for modifying a service. It contains the following fields and content:

- Name:** A text input field containing "Visite du musée de l'horreur".
- Price:** A text input field containing "30".
- Description:** A large text area containing the text: "Venez visiter le musée de l'horreur. Durée de la visite : 50 minutes. Prix par personnes. Merci de nous indiquer deux choix d'heure de visite."
- Nombre de personnes:** An empty text input field.
- Date de la visite:** A text input field containing the placeholder "jj/mm/aaaa".
- Choix horaire n°1:** A text input field containing "--:00".
- Choix horaire n°2:** A text input field containing "--:00".

At the bottom of the form is a dark button labeled "Update informations".

Figure 49 : Modification d'un service

Il serait intéressant de permettre cette modification. Pour cela il faut afficher les champs de base, comme lors de l'ajout d'un type de champ, et remplir les valeurs en fonction des données des champs du service. En recréant le modal permettant d'ajouter de nouveaux champs dans la page de modification, des champs supplémentaires peuvent être insérés. La suppression doit être gérée dans la génération des champs, lors de l'ajout à l'aide du modal par un bouton de suppression du bloc de champs.

4.5.3. Intégration d'un système de paiement

Coucou&Co utilise le système Stripe pour la gestion des paiements auprès des clients. Ce système peut facilement être intégré à notre plateforme, car Laravel propose un paquet officiel « Cashier », utilisant Stripe et possédant une documentation sur leur site officiel. Cela correspond à un simple middleware qui est inséré juste après la validation des données du formulaire. Un middleware est une application qui agit entre la View et le Controller, à savoir, dans le cas présent, avant que les données du formulaire soient enregistrées dans la base de données.

4.5.4. Intégration de la synchronisation via API

Utiliser la BookingSync API pour pouvoir créer des comptes clients lors de réservations sur la plateforme, synchroniser les données des réservations et commander automatiquement des services en fonction de celles-ci (p. ex. ménage automatique après le séjour)

Conclusion

La réalisation de ce travail nous a permis de travailler dans un cadre professionnel avec une entreprise cliente ayant des besoins et attentes. Nous avons travaillé en collaboration avec Coucou&Co, en la personne de M. Fischer, au développement d'une idée, à l'analyse de cette entreprise et à l'établissement d'une liste priorisée de fonctionnalités à développer. Nous avons ensuite analysé et choisi une technologie qui correspondait au mieux à notre projet. Les fonctionnalités ont été développées et implémentées dans notre prototype en utilisant la méthodologie SCRUM.

Grâce à cette méthodologie, nous avons eu un contact régulier avec notre client et avons mené à bien les fonctionnalités choisies ayant évolué en correspondant aux désirs dudit client. Ce fut une expérience professionnelle enrichissante et motivante qui, de plus, comprend une application future réelle.

Lors de ce travail, nous avons pour but de démarrer un projet d'application web correspondant à un cas réel, de choisir une technologie adaptée et de fournir un prototype de la plateforme. Malgré le travail fourni de nombreuses fonctionnalités sont encore à intégrer pour que la plateforme puisse être utilisée en production du fait du nombre de story points nécessaires à son aboutissement ainsi que de la complexité des fonctionnalités.

Références

- BookingSync. (2019, 05 09). Who's Behind BookingSync and Why We Care. Récupéré sur <https://www.bookingsync.com/en/about-us>
- Bexio. (2019, 05 09). Business Software for Small Businesses and Startups. Récupéré sur <https://www.bexio.com/en-CH/>
- Microsoft Excel. (2019, 05 09). Microsoft Excel. Récupéré sur <https://products.office.com/fr-ch/excel>
- Rental Ninja. (2019, 05 09). Manage your rentals effortlessly in one place. Récupéré sur <https://try.rental-ninja.com/en/>
- Warnimont, J. (2019, 05 24). 11 Meilleures plates-formes Open Source et Free Ecommerce pour 2019. Récupéré sur <https://ecommerce-platforms.com/fr/articles/open-source-ecommerce-platforms>
- Automattic. (2019, 05, 24). WooCommerce – Extension WordPress. Récupéré sur <https://fr.wordpress.org/plugins/woocommerce/>
- Codeur.com. (2019, 05 24). ZenCart – Comparatif des boutiques en ligne. Récupéré sur <https://www.codeur.com/comparatifs/solution/zencart/>
- Shopify. (2019, 05, 24). Solution Ecommerce - Créer votre Boutique - Shopify France. Récupéré sur <https://fr.shopify.com/>
- Magento. (2019, 05, 26). plateforme eCommerce | Meilleur logiciel de commerce électronique pour la vente en ligne. Récupéré sur <https://magento.com/fr>
- Laravel. (2019, 05, 26). Laravel - The PHP Framework For Web Artisans. Récupéré sur <https://laravel.com/docs/5.8/>
- Express. (2019, 05, 26). Express - Infrastructure d'application Web Node.js. Récupéré sur <https://expressjs.com/fr/>
- Django. (2019, 05 26). The Web framework for perfectionists with deadlines | Django. Récupéré sur <https://www.djangoproject.com/>

Déclaration de l'auteur

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- Jean-Pierre Rey
- Andreas Fischer

Sierre, le 30 Juillet 2019

Gaétan Huser