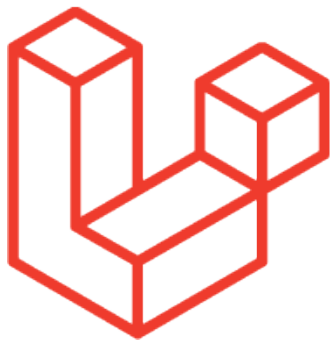


Rapport de stage
Développeur Web
04 Mai au 24 Juillet 2020



Laravel

Cercle SPA

Développement de fonctions sur le logiciel de gestion
clients avec le framework Laravel

Tuteur : Alain GAZONI



Remerciements

Je tiens tout d'abord à remercier mon maître de stage Alain GAZONI, associé et Responsable Développement Informatique au sein de Stratégies & Perspectives d'Avenir, pour avoir tout mis en œuvre pour que ce stage se déroule correctement malgré le contexte actuel, ainsi que pour avoir assuré un très bon suivi et m'avoir encouragé tout au long du stage.

Je remercie également Ali TAZI, associé et Responsable Coordination des Pôles au sein de l'entreprise, pour les félicitations dont il m'a fait part et sa proposition d'embauche, et je remercie aussi que toute l'équipe présente à l'agence durant ma dernière semaine de stage pour leur accueil chaleureux et leurs retours positifs sur mes travaux.

Enfin, je tiens à remercier Laurent THOMANN, professeur au sein de l'Université de Lorraine et responsable des stages, pour les nombreuses propositions de stage qu'il nous a transmis et dont l'offre pour Stratégies & Perspectives d'Avenir faisait partie, ainsi que Jeanine SOUQUIÈRES, ma professeure référente, pour son suivi.

Sommaire

Remerciements.....	2
Sommaire.....	3
I. Introduction.....	4
II. Stratégies & Perspectives d'Avenir.....	5
III. Le projet Cercle SPA.....	8
A. Le projet.....	8
B. Laravel.....	8
IV. Contenu du stage.....	12
A. Tâches confiées.....	12
B. Matériel, logiciels et langages utilisés.....	13
V. Mise en œuvre.....	14
A. Remplissage automatique de l'onglet Professionnel.....	14
B. Anomalies contrat et démarches commerciales.....	15
C. Format des dates et protection des formulaires.....	18
D. Édition du devoir de conseil.....	20
E. Exportation des événements clients.....	22
F. Filtrage des contrats.....	24
G. Page d'accueil et changement de portefeuille.....	25
H. Ajout des besoins clients.....	31
I. Réalisateurs et Apporteurs dans l'activité par date.....	32
J. Valeur du portefeuille, primes et événements produit.....	33
K. Protocoles de commissions.....	40
L. Profil investisseur.....	43
M. Gestion Électronique des Documents (GED).....	47
N. Signature électronique.....	54
VI. Conclusion.....	58
VII. Après-texte.....	59
A. Annexes.....	59
B. Glossaire.....	66
C. Bibliographie.....	67

I. Introduction

Dans le cadre de ma formation en Licence MIAHS (Mathématiques et Informatique Appliquées aux Sciences Humaines et Sociales) parcours Sciences Cognitives, et afin de clore cette dernière, j'ai réalisé mon stage dans le regroupement de courtiers en assurances Stratégies & Perspectives d'Avenir.

Durant ce stage, j'ai travaillé sur Cercle SPA, le logiciel de gestion clients de l'entreprise, développé sur le framework PHP Laravel. J'ai été amené à améliorer l'ergonomie générale et certaines fonctionnalités du logiciel, ainsi qu'à en ajouter d'autres totalement nouvelles, pour répondre à des besoins tels que l'accès simplifié à l'information et aux données ou la simplification des tâches administratives réalisées par les courtiers.

Dans un premier temps nous allons découvrir plus en détail l'entreprise dans laquelle j'ai réalisé mon stage, avant de poursuivre sur la présentation du logiciel Cercle SPA et une brève présentation du framework Laravel, dans le but de faciliter la compréhension de la suite de ce rapport. Ensuite, nous aborderons la liste des tâches qui m'ont été confiées ainsi que le matériel et les langages utilisés, puis nous continuerons avec la mise en œuvre de ces différentes tâches. Enfin nous clôturerons ce rapport avec une conclusion faisant un bilan global du stage, avec les tâches effectuées et ce que ce dernier m'a apporté.

II. Stratégies & Perspectives d'Avenir

1. Présentation

Stratégies & Perspectives d'Avenir (SPA) est une société par actions simplifiée (SAS) dont l'agence fondatrice est située Boulevard de Baudricourt à Villers-lès-Nancy. Cette entreprise regroupe des consultants en régimes sociaux et des courtiers en assurances dont les compétences permettent de proposer aux particuliers et aux entreprises des solutions adaptées à leurs besoins.



Logo de Stratégies & Perspectives d'Avenir

Leur travail consiste à récolter les besoins de leurs clients, ainsi que d'étudier, analyser et comparer les différentes solutions disponibles sur le marché pour mettre en place les solutions appropriées en terme de régime social, d'assurance vie (retraite, santé, épargne...) et IARD (Incendies, Accidents et Risques Divers : habitation, auto...).

L'entreprise est en collaboration avec une quarantaine de compagnies d'assurances, dont on peut citer ACORIS, Alptis, CARDIF, Generali ou SwissLife. Elle assure le rôle d'intermédiaire entre les clients et les sociétés d'assurances et dispose donc d'un large éventail de solutions pour une liste variée de besoins, afin de proposer aux clients un rapport qualité/prix optimal.

2. Organisation

L'entreprise possède une agence regroupant les associés fondateurs de la société et leur équipe composée de :

- Alain GAZONI, associé et Responsable Développement Informatique
- Ali TAZI, associé et Responsable Coordination des Pôles
- Sylvain MAILLARD, associé et Responsable Relations Compagnies
- Stéphane SCALABRINO, associé et Responsable Formation et Dev. réseaux
- Delphine TATON, assistante de direction et appui administratif
- Hakim TAZI, assistant commercial et appui techniques de vente

Stratégies & Perspectives d'Avenir regroupe plusieurs sociétés de courtage situées en France et les associés fondateurs disposent également de leur propre société de courtage

affiliée à l'entreprise. Ils travaillent donc pour leur propre société et pour le développement de l'entreprise.

À ce jour, voici les sociétés composant Stratégies & Perspectives d'Avenir :

Région (Département)	Société	Dirigeant
Alsace (67)	OG Conseils	Olivier GOLDSTEIN
Alsace (67)	Agil Conseils & Courtage	Patrick MUNTZINGER
Alsace (68)	Consilium Assurances	Fabien SCHNOEBELEN
Alsace (68)	Caveriviere Conseil Courtage	Guillaume CAVERIVIERE
Auvergne (63)	Maxi Prêts Auvergne	Laurent COUDEYRAS
Bourgogne (89)	GAP Assurances	Franck PEPIN
Bretagne (22)	DIWALL Courtage	Françoise CARRE
Franche-Comté (70)	EB Consulting	Emmanuel BOUTROU
Franche-Comté (70)	Cabinet MAISOUR Assurances & Conseil	Ahmid MAISOUR
Languedoc-Rouss. (66)	ESPINOSA	Olivier SANCHEZ
Lorraine (54)	Est Cap	Alain GAZONI
Lorraine (54)	ECAPS	Philippe L'HUILLIER
Lorraine (54)	AXIS CONSULTING	Stéphane SAULNIER
Lorraine (54)	PRE CO	Jocelyne FRESCHOTTE
Lorraine (54)	GRAMMAIRE Julien	Julien GRAMMAIRE
Lorraine (54)	WILIKO	Martine PFISTER
Lorraine (54)	PreviPro	Nicolas BOURGIN
Lorraine (54)	Moselle Assurances Conseils	Ali TAZI
Lorraine (54)	Lorraine Courtage en Assurances	Sylvain MAILLARD
Lorraine (54)	SPA Courtage VIE	
Lorraine (54)	Synergie Conseils & Courtage	
Lorraine (54)	SPA Courtage IARD	
Lorraine (55)	GD Conseils & Courtage	
Lorraine (57)	ACCAP	
Lorraine (57)	GD CROISSANCE PATRIMOINE	Gregory DESPREZ
Lorraine (57)	CMB Assurfinances	Marc ANTOINE
Lorraine (57)	AGENCE IMMO & Co	Adriano PAU

Région (Département)	Société	Dirigeant
Lorraine (88)	NovAssurSolution	Julien ETIENNE
Lorraine (88)	Solution Conseil Courtage	
Lorraine (88)	Stratosphere Courtage	Jean LACAVE
Pays de la Loire (85)	CLBJ	Ludovic BAUDILLON
PACA (13)	AFI Conseil	Jean-Michel RAUX
PACA (13)	Scalabrino Conseils	Stéphane SCALABRINO
PACA (84)	PHR Conseils	Philippe REUSSARD
Rhône-Alpes (38)	AssurAvenirCourtage	Julien ROGER
Rhône-Alpes (38)	Option Courtage	Claude POITOU
Rhône-Alpes (73)	SR Assurances Courtage	Christine BERGERON
Rhône-Alpes (74)	OK J	Olivier KARAFA

3. Informatique

Stratégies & Perspectives d'Avenir ne possède pas de service informatique à proprement parler. Mon maître de stage Alain GAZONI est en charge du développement et de la maintenance du logiciel de gestion clients Cercle SPA, développé sur le framework PHP Laravel. Ce logiciel est commercialisé aux cabinets rejoignant le regroupement, certaines fonctionnalités faisant office d'options payantes, et c'est au sein de ce projet que mon stage s'est effectué.

III. Le projet Cercle SPA

A. Le projet

Le projet Cercle SPA est le logiciel de gestion clients du regroupement Stratégies & Perspectives d'Avenir et est développé sur le framework* Laravel. Il permet aux conseillers de gérer leur portefeuille.

Le logiciel permet notamment de gérer les clients. Chaque client dispose d'une fiche sur laquelle on peut renseigner de nombreuses informations telles que le nom et le prénom du client, sa date de naissance, ses coordonnées, son statut professionnel, etc.

Depuis cette fiche on peut effectuer plusieurs choses. Tout d'abord, on peut ajouter les besoins exprimés par le client pour différents types de risques. On peut ensuite définir les contrats mis en place ainsi que les contrats qu'il possédait déjà (dossiers concurrents). À partir des besoins exprimés et des contrats, on peut éditer plusieurs documents PDF, notamment le devoir de conseil qui est le document le plus important dans la relation du conseiller avec son client. Enfin, on peut ajouter des démarches commerciales (appel, visite, mail...) pour le client choisi. Ces démarches sont résumées sur un tableau de bord dédié et dans un agenda intégré au logiciel.

Sur chaque contrat, on peut définir le montant des primes, le choix du protocole de commission, la répartition des fonds investis, etc. On peut y ajouter des événements tels que des avenant d'augmentation des primes et on peut y signaler des anomalies comme un document manquant par exemple.

Cercle SPA possède plusieurs tableaux de bord pour nous permettre d'obtenir et de visualiser rapidement certaines informations. En plus du tableau de bord pour les démarches commerciales, le logiciel possède également des tableaux de bord pour filtrer les contrats par type ou par compagnie, obtenir l'activité par date ou un récapitulatif de la valeur du portefeuille.

B. Laravel

Laravel est le framework PHP utilisé pour le projet Cercle SPA. Le choix de Laravel est en majeure partie dû à sa simplicité. En effet, le framework offre un code simple, lisible et agréable à lire, ce qui permet de développer de nouvelles fonctionnalités rapidement. De plus, Laravel utilise une architecture MVC (Modèle-Vue-Contrôleur), ce qui permet de conserver un code et un projet bien organisé et structuré.

Le framework fournit un système de définition des routes simple, un constructeur de formulaire, un moteur de template puissant et surtout un système d'authentification et de protection face aux injections SQL efficace contre le piratage informatique. Cela fait donc de Laravel un framework simple et performant pour développer rapidement un site ou une application web.

Pour faciliter la compréhension de la suite du rapport, il est important que je détaille ici, de manière un peu plus technique, la structure de fichiers d'un projet Laravel, notamment le rôle de certains fichiers ou dossiers, ainsi que d'expliquer ce qu'est l'architecture MVC.

1. L'architecture MVC

L'architecture MVC est une architecture logicielle dans laquelle le code est séparé en trois parties distinctes : le modèle, la vue et le contrôleur. Cela permet d'avoir un code clair et un projet structuré car on connaît le rôle de chacun des fichiers.

Le modèle va donc gérer les données, comme des clients par exemple, la vue va gérer l'affichage des informations à l'utilisateur et le contrôleur va se charger de faire fonctionner l'ensemble, notamment en traitant les actions de l'utilisateur et en modifiant les données du modèle et de la vue.

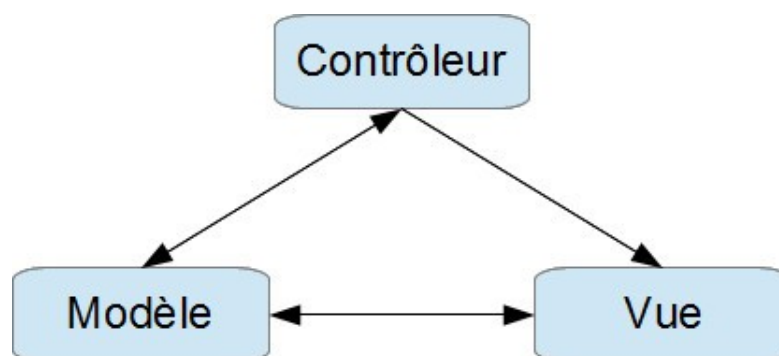


Schéma des interactions entre les fichiers du modèle MVC

Bien entendu, une application peut posséder plusieurs modèles, vues et contrôleurs, pour implémenter plusieurs fonctionnalités.

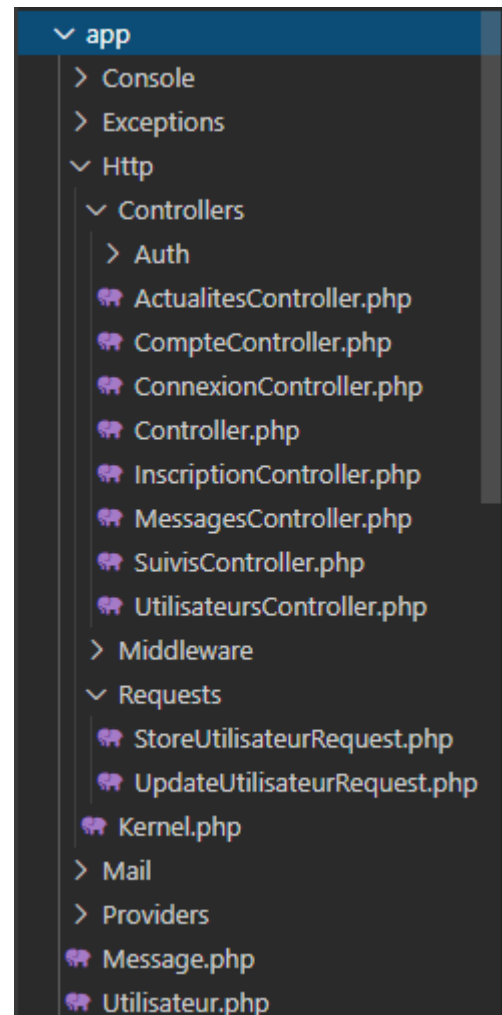
2. app/

Le dossier `app/` contient trois éléments importants : les modèles, les contrôleurs et les classes de validation.

À sa racine se trouve donc les modèles du projet Laravel. Ces derniers correspondent aux tables de la base de données. Le modèle `Client`, par exemple, correspondra à la table `clients` dans la base. Par défaut le nom du modèle doit être le nom de la table au singulier. Dans ces modèles on va définir les attributs modifiables, les accesseurs et les mutateurs, ainsi que les relations avec d'autres tables.

On retrouve ensuite dans le sous-dossier `Http/Controllers/` les contrôleurs du projet, dont certains sont liés à des modèles. Ils contiennent les différentes fonctions appelées par la table de routes. Ces dernières peuvent être utilisées pour l'affichage en retournant une vue ou pour agir sur la base de données afin de stocker, modifier ou supprimer des entrées. Certaines peuvent juste retourner des données, comme c'est le cas pour les fonctions privées ou celles utilisées pour des requêtes AJAX*.

Enfin, dans le sous-dossier `Http/Requests/` se trouve les classes de validation. Ces dernières sont de deux types : `Store` et `Update`, et correspondent à un modèle. Par exemple, `StoreClientsRequest` va se charger de la validation des champs lors de la création d'un client, et `UpdateClientsRequest` va se charger de la validation lors de l'édition d'un client. Ces classes contiennent notamment la fonction `rules`, qui retourne un tableau contenant les règles de validation définies. Pour chaque champ, on peut donc définir s'il est requis, son format, sa taille ou sa valeur minimale et maximale, etc.



Exemple de dossier `app/` dans un projet Laravel

3. resources/views/ et routes/web.php

Le dossier `resources/views/` contient toutes les vues de l'application, sous forme de templates Blade. L'utilisation des templates permet de nombreuses choses, comme l'appel de variables et de fonctions PHP et l'utilisation de conditions et de boucles, pour remplir un tableau par exemple.

Grâce à Blade, on peut définir des layouts, correspondant à la structure globale de la page, et pouvant être appelés dans les fichiers vues qui viendront compléter les sections définies dans le layout. On peut également créer des partials, de petits éléments affichés sur la plupart des pages, comme la topbar par exemple.

Le fichier `web.php` permet de définir les routes de l'application, en définissant l'URI* utilisée, ainsi que le contrôleur et la fonction appelés. On peut passer des variables à la fonction appelée en définissant l'élément de l'URI correspondant à la variable. Par exemple, l'URI `/client/1` avec une route `/client/{id}` passera la variable `id=1` à la fonction appelée.

4. .env et config/app.php

Le fichier `.env` permet de configurer certains éléments du projet, comme le nom de l'application et son URL*, les identifiants pour la base de données, pour un serveur mail ou encore pour un serveur Amazon Web Service (AWS).

Le fichier `app.php` quant à lui, permet de configurer d'autres éléments comme le fuseau horaire, le format des dates, la langue par défaut et les alias de classe, pour les extensions notamment.

IV. Contenu du stage

A. Tâches confiées

Pour débiter mon stage et pour me familiariser avec le projet, on m'a confié quelques tâches : la mise en place du remplissage des informations professionnelles d'un client, l'ajout d'un tableau de bord pour les anomalies contrat, la mise en place du format des dates en français (jj/mm/aaaa) sur tout le projet, la modification de l'édition de certains documents PDF ainsi que la mise en place de la protection des formulaires contre les erreurs de saisie.

Par la suite, j'ai été chargé de l'exportation des événements clients vers des services tels que Google Agenda pour permettre aux conseillers de consulter leur calendrier plus facilement. En parallèle de cela, j'ai réalisé plusieurs tâches dont l'amélioration du filtrage des contrats sur le tableau de bord "Filtrer mes contrats", la protection de la page d'accueil pour les nouveaux comptes et la mise en place du changement de portefeuille, afin que certains conseillers autorisés puissent consulter les autres portefeuilles du regroupement.

J'ai ensuite amélioré certaines fonctionnalités tels que l'expression des besoins clients, l'ajout d'une filtration par réalisateur ou apporteur au tableau de bord "Activité par date" et d'une autre par type d'événement dans "Démarches commerciales", avec la possibilité d'exporter les données obtenues en fichier PDF ou Excel.

Une fois les tests sur l'exportation terminés, on m'a chargé de déplacer et d'améliorer les calculs d'estimation de la valeur du portefeuille. Dans la continuité, j'ai ajouté une fonctionnalité pour modifier rapidement la prime périodique pour des contrats sélectionnés, cette dernière augmentant naturellement tous les ans. Afin d'alléger le travail des conseillers, j'ai amélioré l'ajout d'événements produits pour que certains champs soient repris de l'événement précédent.

Par la suite, on m'a demandé de mettre en place différents widgets* sur la page d'accueil pour fournir aux conseillers une synthèse de l'activité de leur portefeuille. On m'a également demandé d'ajouter la définition d'un apporteur dans les démarches commerciales avec un filtre pour celui-ci sur le tableau de bord associé.

Plus tard, on m'a confié plusieurs tâches liées aux protocoles de commissions, notamment la mise en place de l'importation des protocoles via un fichier CSV* et d'une page listant les produits sans protocole. On m'a également demandé d'ajouter un autre widget sur la page d'accueil, donnant la répartition du chiffre d'affaire par type de contrats, ainsi qu'une page pour relever et corriger les erreurs de saisie des avenants.

Pour mes dernières semaines de stage, on m'a demandé de mettre en place une fonctionnalité totalement nouvelle pour renseigner le profil investisseur d'un client, puis on m'a demandé de mettre en place la Gestion Électronique des Documents (GED) sur un serveur Amazon S3. Pour finir, on m'a confié la mise en place de la signature électronique pour les documents générés par le logiciel.

B. Matériel, logiciels et langages utilisés

Le stage s'étant déroulé en télétravail, j'ai utilisé mon ordinateur personnel pour le réaliser. Mon ordinateur fut également utilisé comme serveur de développement, je n'ai donc pas eu besoin de matériel supplémentaire.

Pour la mise en place du serveur, j'ai utilisé Laragon, un environnement de développement web complet avec un serveur Apache et MySQL, permettant d'installer rapidement un projet sous différents CMS* et dans mon cas, un projet Laravel, utilisé en amont du stage, pour découvrir le framework.

Afin d'installer la version de développement du projet sur mon serveur, mon maître de stage a mis en place un dépôt GitHub et j'ai utilisé le logiciel GitHub Desktop pour mettre en place mon dépôt local. Le développement s'est fait sur Visual Studio Code sur lequel j'ai ajouté les extensions Laravel Snippets et Blade Snippets pour simplifier l'utilisation des syntaxes propres à Laravel et au moteur de template Blade.

Pour communiquer avec mon maître de stage, nous avons utilisé Skype. Cela m'a permis de lui montrer facilement ma progression sur les tâches confiées, notamment grâce au partage d'écran pour lui faire démonstration de mes travaux.



Les langages utilisés ont été contraints par le framework Laravel. J'ai donc utilisé PHP, HTML 5, CSS 3 avec Bootstrap 3.3.7 et JavaScript avec jQuery. Pour certaines fonctionnalités, j'ai été amené à utiliser les extensions DataTables, Chart.js, jQuery Steps, jQuery Validator et DropzoneJS, ainsi que Laravel Phone et MediaLibrary.



V. Mise en œuvre

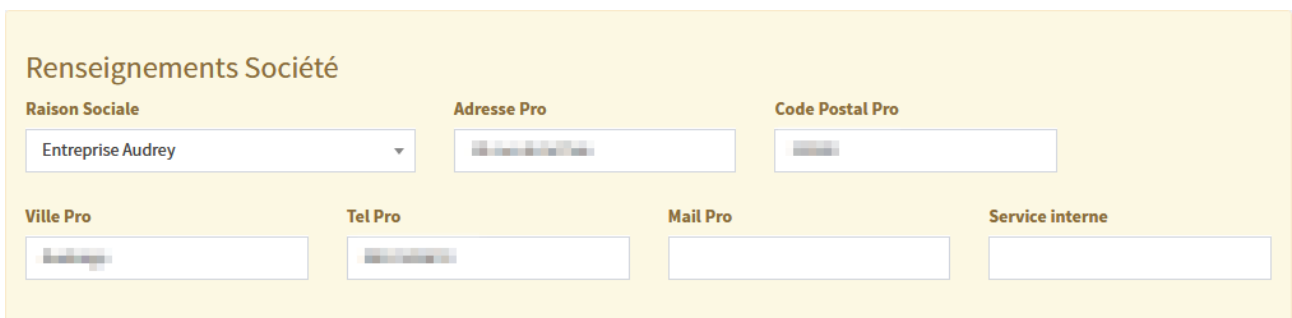
Durant ce stage, le développement a été réalisé sur mon serveur de développement. Une fois une fonctionnalité achevée, je faisais une démonstration à mon maître de stage via Skype afin qu'il valide ou non mon travail. Une fois validé, je réalisais un commit du projet sur GitHub pour qu'il puisse récupérer mon code, le tester de son côté et le mettre en ligne sur le serveur de production.

On faisait régulièrement un point sur la mise en production afin de régler les éventuels bugs qui ont pu lui être signalés.

A. Remplissage automatique de l'onglet Professionnel

Dans la version initiale de Cercle SPA, lorsque l'on remplissait les informations professionnelles d'un client, on devait choisir sa raison sociale (son entreprise) et saisir toutes les informations sur l'entreprise, bien que l'entreprise ait été renseignée au préalable avec toutes ces informations. La première tâche que l'on m'a confié a donc été d'automatiser cette saisie.

Pour ce faire, j'ai mis en place une requête AJAX, appelée lorsque l'on sélectionne une entreprise et récupérant les informations de l'entreprise sélectionnée pour les insérer dans les champs correspondants du formulaire. J'ai ensuite ajouté l'URL de la requête AJAX aux routes et une fonction `getInformations($raison_sociale_pro_id)`, appelée par cette nouvelle route, au `DropDownListController`. Cette fonction retourne les informations sur l'entreprise portant l'id passé en paramètre.



The screenshot shows a web form titled "Renseignements Société" on a light yellow background. The form contains several input fields, some of which are pre-filled with data. The fields are arranged in two rows. The first row includes "Raison Sociale" (a dropdown menu showing "Entreprise Audrey"), "Adresse Pro" (a text box with a blurred address), and "Code Postal Pro" (a text box with a blurred postal code). The second row includes "Ville Pro" (a text box with a blurred city name), "Tel Pro" (a text box with a blurred phone number), "Mail Pro" (an empty text box), and "Service interne" (an empty text box).

Renseignements Société avec les champs complétés automatiquement


B. Anomalies contrat et démarches commerciales

Afin de permettre aux conseillers de visualiser toutes les anomalies contrat, on m'a demandé de mettre en place un nouveau tableau de bord affichant toutes les anomalies contrat du portefeuille, en m'inspirant du tableau de bord des démarches commerciales, avec un filtre pour afficher uniquement les anomalies pour la compagnie sélectionnée. On m'a également demandé d'ajouter une signalétique dans la liste des contrats d'un client.

1. Tableau de bord pour les anomalies contrat

J'ai donc commencé par mettre en place le nouveau tableau de bord pour les anomalies contrat. À l'état initial du projet, aucun outil permettait de visualiser toutes les anomalies. Le conseiller était obligé de vérifier tous ses contrats un à un.

Cette fonctionnalité nécessitant une page dédiée, j'ai dans un premier temps créé la route, la vue ainsi qu'un bouton d'accès sur le tableau de bord, puis j'ai ajouté la fonction `list` au `AnomalieParProduitsController`. Elle se charge de récupérer toutes les anomalies contrat non clôturées du portefeuille ainsi que les compagnies correspondantes à ces contrats puis retourne la vue avec les données obtenues. La vue affiche ensuite, compagnie par compagnie, la liste des anomalies contrat dans des tableaux. Grâce à deux boutons, on peut modifier ou supprimer une anomalie depuis cette page.

 **Filtrer par compagnie**

Compagnie

ALLIANZ

Choisir

Retour à toutes les compagnies

Afficher

100

 éléments

Copier

CSV

Excel

PDF

Imprimer

Visibilité de la colonne

Rechercher :

ALLIANZ

Produit	Client	Type	Commentaire	Date	
Composio	Dupont Esther	Ajournement		28/05/2020	<div>Modifier</div> <div>Supprimer</div>
Habitation	Dupont Benjamin	Réclamation	Grosse Anomalie	04/05/2020	<div>Modifier</div> <div>Supprimer</div>

Affichage de l'élément 1 à 2 sur 2 éléments

Précédent

1

Suivant

Anomalies produits pour Allianz

Pour permettre au conseiller de visualiser les anomalies pour une compagnie en particulier, j'ai ajouté dans la vue un formulaire avec un menu déroulant permettant de choisir une compagnie parmi celles présentes. J'ai ensuite ajouté une nouvelle route pour différencier les deux versions de la page. La vue et la fonction appelées restent les mêmes, seul le contenu affiché par la vue change.

2. Signalétique dans la liste des contrats

Afin de permettre aux utilisateurs de savoir rapidement si un contrat contient une anomalie, j'ai ajouté une signalétique dans la liste des solutions d'assurance d'un client.

Pour se faire, j'ai ajouté une requête SQL à la fonction `index` du `ProduitsparclientsController` afin de récupérer les anomalies non clôturées des produits du client. Puis j'ai modifié la vue pour qu'un panneau danger soit affiché pour les produits dont l'id correspond au `NumDossier` d'une des anomalies récupérées. Plus tard, j'ai également ajouté l'indication de la formalisation des contrats sur cette vue, ainsi que dans les événements par produits des contrats.

Produit	Numéro Contrat	Souscripteur	Fractionnement	Situation contrat	Art 83 ?		
ACS TNS		Carbonnier	Non Defini	En cours (Attente Emission)	<input type="checkbox"/>		Détails Supprimer
AVIVA Retraite Plurielle Madelin		Carbonnier	Non Defini	En cours (Attente Emission)	<input type="checkbox"/>		Détails Supprimer
Garantie Chômage		Carbonnier	Non Defini	En cours (Attente Emission)	<input type="checkbox"/>		Détails Supprimer
Adip Optimum		Carbonnier	Non Defini	En cours (Attente Emission)	<input type="checkbox"/>		Détails Supprimer
Protection Juridique Vie Quotidienne		Carbonnier	Non Defini	En cours (Attente Emission)	<input type="checkbox"/>		Détails Supprimer

Liste des solutions d'assurance d'un client avec la signalétique des anomalies

3. Tableau de bord pour les démarches commerciales

Plus tard, on m'a demandé d'ajouter un filtre, similaire à celui présenté juste avant, dans le tableau de bord des démarches commerciales. J'ai donc mis en place un formulaire contenant un menu déroulant avec les types de démarche présents. Le filtre fonctionne donc de la même façon que celui pour les anomalies.

J'ai ensuite modifié les tables de ce tableau de bord pour qu'elles utilisent le plug-in `DataTables` de `jQuery`, déjà utilisé sur d'autres pages de Cercle SPA. Ce plug-in permet entre autre d'ajouter une pagination aux tableaux, de sélectionner les colonnes visibles, de changer l'ordre de tri et surtout d'exporter un tableau en fichier Excel ou PDF. J'ai donc fait les modifications nécessaires dans la vue, en ajoutant la classe `datatable` aux tables.

J'ai alors découvert qu'il existait une intégration de DataTables pour Bootstrap, afin d'avoir un design uniforme. J'ai donc modifié les CDN* des scripts et des feuilles de styles pour utiliser ceux adaptés à Bootstrap. Ainsi, les pages du projet contenant des datatables ont un visuel beaucoup plus uniforme d'aujourd'hui.

Plus tard on nous a signalé que le tri sur des dates ne se faisait pas correctement, les dates étant considérées comme des nombres normaux. Après quelques recherches, j'ai trouvé une extension du plug-in qui permet de gérer correctement les dates. J'ai donc ajouté les CDN nécessaires et un script afin de définir le format des dates à utiliser.

Par la suite, on m'a demandé d'ajouter la définition d'un apporteur pour ces démarches, ainsi que de prévoir un filtre sur cet élément dans le tableau de bord. J'ai donc commencé par ajouter une colonne apporteur dans la table events de la base de donnée, puis j'ai ajouté un menu déroulant contenant les apporteurs dans le formulaire de création et d'édition des démarches, ainsi que l'affichage de ce nouveau champ dans la liste des démarches pour chaque client et dans le tableau de bord. Pour finir, j'ai ajouté le filtre, fonctionnant de la même manière que celui par type. Grâce à des champs cachés, on a la possibilité de combiner les deux filtres de ce tableau de bord.

Filtrer par Apporteur

Apporteur

GAZONI Alain

Choisir

Filtrer par type

Type historique

Tous les types

Choisir

Retour à toutes les démarches

Afficher 100 éléments Copier CSV Excel PDF Imprimer Visibilité de la colonne Rechercher :

Date événement	Apporteur	Demande	Client	Commentaire	Date max
23/06/2020	GAZONI Alain		Nicolas Carbonnier		Non renseigné

Affichage de l'élément 1 à 1 sur 1 éléments

Précédent 1 Suivant

Démarches commerciales filtrées sur l'apporteur Alain GAZONI

4. Problèmes de redirection

Durant la mise en place et l'amélioration de ces fonctionnalités, j'ai fait face à plusieurs problèmes de redirection, notamment suite à la modification ou la suppression d'une anomalie ou d'une démarche. En effet, après avoir validé les modifications, l'utilisateur était redirigé vers la page définie par la fonction update des contrôleurs et non

vers le tableau de bord correspondant. Pour résoudre cela, j'ai modifié les fonctions `list` du `AnomalieParProduitsController` et `result` du `HistoriqueClientController` pour qu'elles stockent l'URL de la page dans la variable de session. La fonction `update` vérifie alors si la route est présente en session et redirige vers celle-ci si c'est le cas, sinon elle redirige vers la page initialement définie.

La route est supprimée de la variable de session lorsque l'on accède aux modifications des anomalies ou des démarches depuis une autre page.

Ce problème de redirection étant également présent pour la suppression, j'ai ajouté une fonction `destroyFromList`, identique à la fonction `destroy`. Au lieu d'effectuer une redirection vers la page définie, la fonction retourne simplement en arrière, sur le tableau de bord correspondant.

Par la suite, je me suis rendu compte que lorsque l'on modifiait une anomalie ou une démarche après avoir appliqué un filtre, on était toujours redirigé vers la page générale, sans filtre appliqué.

Pour résoudre cela, j'ai changé les filtres pour qu'ils utilisent la méthode `GET` au lieu de `POST`. J'ai ensuite supprimé les secondes routes, devenues inutiles, et mis en place le stockage de l'id de la compagnie ou du type en session. Ainsi lorsque l'on modifie une anomalie ou un événement depuis le tableau de bord, la fonction `update` des contrôleurs ajoute l'id de la compagnie (ou du type), si il est présent, à la redirection pour que l'utilisateur se retrouve toujours sur la bonne page. J'ai par la suite fait de même avec l'apporteur pour le tableau de bord "Démarches commerciales".

C. Format des dates et protection des formulaires

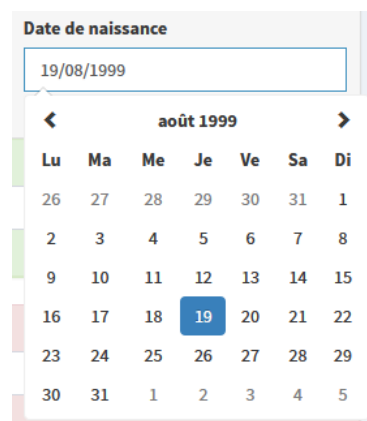
Par la suite, on m'a confié deux tâches liées en partie aux formulaires et à la base de données. On m'a tout d'abord demandé de modifier le format des dates, puis de protéger les formulaire du projet.

1. Format des dates en français

Afin d'améliorer la lisibilité et la saisie des dates sur Cercle SPA, on m'a confié la tâche de mettre le format des dates en français, au format `jj/mm/aaaa`.

Dans un premier temps, j'ai modifié les variables de format des dates dans le fichier `app.php`. Ces variables sont utilisées par différent éléments du projet pour la gestion des dates, notamment par le widget `Date Time Picker` du framework `Bootstrap` dans les champs date des formulaires. Après configuration, l'affichage des dates saisies via le widget de `Bootstrap` était au format demandé.

Pour pouvoir stocker les dates sans problème dans la base de données, j'ai mis en place des accesseurs et des mutateurs dans les modèles contenant des dates. Les mutateurs se chargent de mettre les dates au format DATE ou DATETIME, suivant ce que la base de données attend. Les accesseurs font le processus inverse et retournent des dates avec la syntaxe française. Ainsi, les dates sont envoyées à la base de données au format attendu et sont récupérées avec la syntaxe que nous connaissons. Afin de simplifier la manipulation des dates et les changements de format, j'ai utilisé l'extension PHP Carbon, intégrée à Laravel, qui permet de manipuler aisément les dates.



DateTime Picker de Bootstrap, avec la date au format français

Pour que la saisie des dates par les utilisateurs soit plus facile, j'ai modifié certaines vues pour qu'elles proposent un DatePicker à la place d'un DateTimePicker lorsqu'il n'était pas nécessaire de préciser l'heure, puis j'ai ajouté un paramètre aux DateTimePicker pour que l'on puisse saisir les minutes par créneau de 15 minutes.

2. Protection des formulaires

La modification du format des dates ayant mis en lumière le fait que les utilisateurs peuvent se retrouver face à des erreurs liées aux requêtes SQL, on m'a chargé de mettre en place la protection des formulaires.

Dans un premier temps, j'ai ajouté des mutateurs pour tous les attributs éditables des modèles. Cela m'a permis de définir la valeur par défaut d'un attribut si ce dernier n'était pas saisi dans un formulaire, ce qui était notamment important pour les nombres, car la base de données refuse d'enregistrer une chaîne vide pour un nombre et attend donc par défaut la valeur NULL. Ensuite, j'ai ajouté des règles des validations dans les fichiers Store et Update, en précisant les champs requis, la taille maximale des champs et leur format.

Pour gérer correctement les numéros de téléphone, j'ai utilisé l'extension Laravel-Phone, une extension Laravel qui permet de manipuler facilement les numéros de téléphone et qui propose des règles de validation pour les formulaires. J'ai donc utilisé la règle phone:FR fournie par l'extension pour valider les numéros de téléphone. Ainsi, les utilisateurs peuvent saisir un numéro de téléphone avec ou sans espace tous les deux

chiffres, ce qui leur offre assez de souplesse. J'ai ensuite standardisé l'enregistrement des numéros dans la base de données. Ils sont enregistrés au format international (+33123456789) et sont récupérés au format français (01 23 45 67 89). Étant conseillé d'utiliser un numéro au format international pour les liens tel:, j'ai utilisé Laravel-Phone dans les vues pour mettre les numéros à ce format.

Bien plus tard, on a été confronté à un problème lié aux numéros de téléphone commençant par 08. En effet, l'extension ne considère pas ces numéros comme des numéros français, ce qui posait problème pour la validation et la conversion au format international. Pour résoudre cela, j'ai ajouté la condition LENIENT à la règle phone:FR pour laisser plus d'indulgence lors de la validation. Puis j'ai modifié les accesseurs et les mutateurs pour ne pas changer le format du numéro si celui-ci commençait par 08. Enfin, j'ai modifié les vues pour que les liens tel: en 08 utilisent le numéro sans espace, sans le convertir au format international.

D. Édition du devoir de conseil

Afin de donner plus de souplesse dans l'édition du devoir de conseil, on m'a demandé de le modifier afin que la date de signature ne soit pas récupérée dans la base de données mais saisie manuellement par l'utilisateur.

Devoir de conseil pour Nicolas Carbonnier

Vous allez éditer un devoir de conseil

À quelle date voulez-vous l'éditer ?

Générer

Vous allez éditer un devoir de conseil

IARD

À quelle date voulez-vous l'éditer ?

Générer

[Retour à la liste des solutions d'assurance](#)

Page d'édition du devoir de conseil

Pour cela, j'ai créé une vue intermédiaire à laquelle on accède lorsque l'on clique sur le bouton "Devoir de conseil en PDF", qui lançait initialement l'édition du document PDF. Une fois sur la vue, l'utilisateur peut saisir une date et éditer le document. Certains conseillers étant spécialisés dans les assurances IARD, ils peuvent éditer un autre devoir de

conseil, lié aux besoins en IARD. J'ai donc ajouté un second formulaire dans la vue intermédiaire, qui s'affiche uniquement lorsque le conseiller connecté est déclaré conseiller IARD.

Afin de simplifier le code, j'ai déplacé la fonction chargée d'éditer le devoir de conseil IARD dans le CustomerController, qui contient maintenant les fonctions printPDF et printPDFiard, chargées d'éditer les deux types de devoir de conseil. J'ai également supprimé la vue utilisée initialement par la seconde fonction, étant identique à la première. Seul le contenu passé à la vue par les fonctions diffère.

Plus tard, j'ai de nouveau modifié l'édition du devoir de conseil pour indiquer explicitement lorsqu'il n'y a aucun besoin non exprimé pour un type de produit, l'affichage initial laissant penser à un oubli dans l'indication des besoins non exprimés.

Je l'ai également modifié pour ne pas afficher la mention "OUI" lorsqu'un besoin n'a qu'une seule occurrence et pour régler une erreur qui empêchait les adéquations des solutions mises en place de s'afficher. J'ai également ajouté des conditions sur la vue afin de m'assurer que la partie signature ne soit pas séparée sur deux pages, car cela arrivait lorsqu'un certain nombre de contrat était listé sur le document.

En lien avec ce document, on m'a demandé d'ajouter un champ pour les événements par produits afin de renseigner le pourcentage d'adéquation du contrat par rapport aux besoins exprimés du client. J'ai donc ajouté une colonne PourcentageAde dans la table evenementsparproduits, puis j'ai modifié le formulaire d'édition des événements produit pour y ajouter ce nouveau champ, prenant la forme d'un texte à trous avec le champ texte déjà prévu pour l'expression de l'adéquation. J'ai par la suite modifié la vue du PDF pour que l'affichage des adéquations reprenne la tournure de phrase utilisée dans le formulaire.



Adéquation solution/besoins ?

La solution mise en place correspond à % de vos besoins exprimés et notamment

Indication du pourcentage d'adéquation dans un événement produit

E. Exportation des événements clients

Le logiciel Cercle SPA permet aux conseillers de renseigner des démarches commerciales (appel, mail, visite...) depuis une fiche client. Ces événements sont affichés dans un agenda grâce à l'extension FullCalendar. Afin de permettre aux conseillers de consulter rapidement leur agenda Cercle SPA, depuis leur smartphone par exemple, on m'a demandé de mettre en place l'exportation des événements pour pouvoir les ajouter sur des agendas tels que Google Agenda, Outlook ou Calendrier de iOS.

1. Génération des fichiers .ics

Le moyen le plus simple d'exporter un calendrier, et celui utilisé par la plupart des applications, est l'utilisation du format iCalendar. Il s'agit d'un format de fichier texte qui permet d'exporter et importer des calendriers avec tous les événements, leurs horaires, leur nom, une description, etc.

Afin de générer les fichiers .ics (l'extension du format iCalendar), j'ai créé le contrôleur CalendarExportController avec une fonction `getEventsICal($id)` que se charge de récupérer tous les événements du conseiller portant l'id passé en paramètre, puis de construire la chaîne de caractère correspondant au contenu du fichier .ics pour enfin retourner un fichier `cal.ics` que l'utilisateur peut directement télécharger.

Pour construire le fichier, on commence par créer l'entête, avec notamment le nom du calendrier (ici Cercle SPA – Nom du conseiller) et la méthode (ici PUBLISH), puis on ajoute tous les événements un à un, avec notamment les dates de début et de fin, le nom, ainsi que l'identifiant unique (ici l'id de l'événement dans la base de données) qui permet aux agendas de le reconnaître lors d'une mise à jour.

Lien d'exportation du calendrier

AGENDA
[Retour à l'accueil](#)

< >

Aujourd'hui

juin 2020

Mois
Semaine
Jour
Planning

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
1	2	3	4	5	6	7
8	9	10	11	12 11:45 Courrier : Carbc 11:45 Appel : Carbonr	13	14
15	16	17	18	19	20	21

Agenda FullCalendar avec le lien d'exportation

Pour que les utilisateurs puissent accéder au téléchargement, j'ai ajouté une route afin d'avoir une URL unique pour chaque conseiller, indiquant l'id de ce dernier. Puis j'ai modifié la vue de l'agenda pour renseigner l'URL de téléchargement.

2. Importation sur Google Agenda et Outlook

Après m'être assuré que les fichiers générés étaient sans erreurs et ouvrables, j'ai débuté les tests d'importation sur Google Agenda et Outlook.

Mon serveur de développement fonctionnant en local, j'ai utilisé l'outil ngrok fourni par Laragon pour créer un tunnel afin d'accéder au serveur depuis l'extérieur, et donc depuis Google Agenda ou d'autres services. Pour m'assurer que les agendas sachent que le lien fourni correspond à un abonnement et non à une importation unique, j'ai modifié le lien pour qu'il commence par `webcal://` à la place de `http://` car il semblerait que beaucoup d'agendas considèrent seulement les liens `webcal` comme des liens d'abonnement.

Après l'importation, il était important de s'assurer que la synchronisation s'effectuait correctement. J'ai donc modifié et ajouté des événements sur l'agenda de Cercle SPA pour voir si ces derniers étaient bien ajoutés sur Google Agenda et Outlook. Malheureusement, la synchronisation n'est pas instantanée et n'est pas paramétrable sur les agendas utilisés. De plus, le tunnel créé par ngrok n'est disponible que durant huit heures. Il était donc difficile de vérifier le bon fonctionnement de la synchronisation.

3. Mise en production

Du fait des limitations techniques liées à ngrok qui rendaient les tests de synchronisation difficiles, nous avons décidé de mettre en production cette fonctionnalité, bien qu'elle n'était pas encore achevée, afin de réaliser des tests plus approfondis.

Après la mise en production, je me suis rendu compte que les fuseaux horaires étaient gérés différemment par les agendas. Pour m'assurer que le fuseau horaire utilisé est le bon, j'ai ajouté sa définition dans l'entête du fichier. Je me suis également rendu compte que les fichiers créés contenaient beaucoup d'événements, j'ai donc modifié la requête SQL pour limiter l'exportation aux événements datant de moins de deux semaines.

En testant les calendriers exportés avec un validateur, j'ai appris que les fichiers iCalendar étaient limités à 75 caractères par ligne. La description de certains événements s'est alors révélée trop longue. J'ai donc ajouté une fonction privée `breakString($string, $maxLength)` dans le `CalendarExportController` afin de découper la chaîne `string`

fournie en paramètre en plusieurs lignes. Puis, pour m'assurer de l'unicité des identifiants de chaque événement par rapport à d'autres calendriers, j'ai crypté l'id des événements avec la fonction MD5 et j'ai ajouté "@cerclespa.fr" à la fin de l'identifiant.

Après quelques jours de tests, la synchronisation s'est montrée parfaitement fonctionnelle sur Google Agenda et Outlook. J'ai alors étendu l'exportation aux événements datant de moins d'un mois afin que les conseillers puissent avoir une vision de leur agenda sur le dernier mois.

F. Filtrage des contrats

Dans la version initiale de Cercle SPA, il était possible de filtrer ses contrats par compagnie ou par type de produit (épargne, retraite, santé...) grâce au tableau de bord "Filtrer mes contrats". Afin de rendre cette fonctionnalité plus complète et précise, on m'a donc demandé de l'améliorer pour pouvoir filtrer sur un produit en plus de la compagnie, ainsi que sur une compagnie et un produit en plus du type de produit.

Page de recherche pour filtrer les contrats par compagnie ou type de risques

J'ai commencé par modifier la fonction create du MescontratsController pour que les listes des compagnies et des types de produits contiennent uniquement ceux ou celles pour lesquels le conseiller a des contrats actifs. J'ai ensuite modifié la vue pour ajouter un second menu déroulant pour le filtrage par compagnie et deux autres pour le filtrage par type de produit. Ces derniers s'affichent uniquement lorsque l'on sélectionne une compagnie ou un type de produit. Des requêtes AJAX se chargent alors de remplir les menus déroulants affichés.

Par la suite, j'ai ajouté les routes et les fonctions appelées par les requêtes AJAX. Les trois fonctions ajoutées au `DropDownListController` se chargent alors de retourner la liste des produits ou compagnies pour lesquels le conseiller a des contrats actifs et qui correspondent à un type de produit ou à une compagnie passé en paramètre. Afin d'afficher les bons éléments dans la page des résultats, j'ai modifié les fonctions `res` et `compagnies` du `MescontratsController` pour effectuer les bonnes requêtes SQL en fonction des éléments sélectionnés dans les menus déroulants.

Enfin, pour que le conseiller sache sur quels éléments il a filtré ses contrats une fois sur la page des résultats, j'ai modifié les vues pour qu'elles indiquent pour quelle compagnie, quel produit ou type de produit elles donnent les résultats.

G. Page d'accueil et changement de portefeuille

Par la suite, on m'a confié plusieurs tâches liées à la page d'accueil, à savoir la protection de celle-ci pour les nouveaux comptes et l'amélioration du changement de portefeuille. On m'a demandé plus tard d'ajouter des widgets sur cette page pour informer les conseillers sur leur activité.

1. Protection de la page d'accueil

Lorsqu'un utilisateur se crée un compte sur Cercle SPA, ce dernier renseigne plusieurs informations telles que le nom, le prénom, l'adresse mail, le mot de passe et un numéro ORIAS, qui permet d'identifier le portefeuille conseiller, les utilisateurs et les conseillers étant deux choses distinctes sur le logiciel. Hors, si ce numéro n'est pas renseigné ou s'il ne correspond pas encore à un portefeuille, l'utilisateur se retrouve face à une erreur SQL, étant donné que le logiciel ne trouve pas de portefeuille associé.

On m'a donc demandé de modifier le fonctionnement de la page d'accueil pour éviter de laisser l'utilisateur devant une erreur SQL.

J'ai donc commencé par modifier la fonction `index` du `HomeController`. Maintenant, c'est uniquement si elle trouve un portefeuille conseiller qu'elle enregistre l'id du conseiller et qu'elle récupère ses informations. Ainsi le nouvel utilisateur ne se retrouve plus devant une erreur SQL et peut accéder à l'interface d'administration.

La page d'accueil laissant alors l'accès à de nombreuses fonctionnalités, j'ai modifié la vue pour que, dans le cas où le compte ne soit pas rattaché à un portefeuille, elle affiche uniquement le bouton de déconnexion, ainsi qu'un message indiquant que le compte est limité car il n'est pas encore rattaché à un portefeuille conseiller.

2. Rattachement ORIAS

J'ai ensuite suggéré d'ajouter une page pour permettre à l'administrateur de lier un utilisateur à un portefeuille directement depuis Cercle SPA, sans aller modifier manuellement la base de données comme s'était le cas jusque là.

J'ai donc créé une nouvelle vue contenant un formulaire avec deux menus déroulants pour sélectionner un utilisateur et un portefeuille conseiller auquel le rattacher. J'ai ensuite ajouté le contrôleur `RattachementOriasController` avec une fonction `index` et une fonction `store`. La fonction `index` se charge de récupérer la liste des utilisateurs, triés par ordre des plus récents, ainsi que la liste des portefeuilles, puis retourne la vue. La fonction `store`, quant à elle, se charge de récupérer l'utilisateur sélectionné et d'y mettre à jour la colonne `numero_orias` avec le numéro ORIAS du portefeuille sélectionné.

Pour finir, j'ai créé la route et une entrée dans le menu de gestion des comptes pour accéder à cette nouvelle page.

Plus tard, mon maître de stage m'a demandé de rajouter la définition des rôles dans le formulaire que j'ai créé. J'ai alors ajouté un champ select contenant les rôles dans la vue, puis j'ai mis en place une requête AJAX afin de sélectionner automatiquement les rôles déjà définis pour l'utilisateur sélectionné pour ne pas supprimer ses anciens rôles lors de la validation. J'ai donc ajouté une fonction `getRoles($user)` dans le `DropDownListController` qui retourne l'utilisateur sélectionné avec ses rôles.

Rattachement ORIAS

Utilisateur

Smith John (js@gmail.com)

Rôle* Sélectionner tous Supprimer tous

x Courtier

Portefeuille conseiller auquel le rattacher

GAZONI Alain

Enregistrer

Formulaire de Rattachement ORIAS

3. Changement de portefeuille

On m'a ensuite demandé d'améliorer le changement de portefeuille afin de le rendre fonctionnel sur tout le site. En effet, à l'état initial de Cercle SPA, seuls les clients étaient accessibles grâce au changement de portefeuille.

Dans un premier temps, j'ai modifié la plupart des fonctions utilisant l'id d'un conseiller pour qu'elles le récupèrent via la variable de session et non le numéros ORIAS de l'utilisateur. Puis j'ai modifié la fonction `res` du `PortefeuilleController` pour que, au lieu de retourner une vue affichant la liste des clients du portefeuille sélectionné, elle stocke dans la variable de session l'id du conseiller sélectionné puis retourne sur la page d'accueil.

J'ai dû ensuite modifier la fonction `index` du `HomeController` afin de ne pas écraser l'id du conseiller enregistré dans la variable de session si celui-ci est différent de celui du conseiller connecté. Si les deux id sont différents, la fonction récupère les informations du portefeuille conseiller sélectionné et les passe à la vue.

J'ai ensuite modifié la vue de la page d'accueil pour qu'elle affiche un encadré contenant le nom et le prénom du conseiller pour lequel on a sélectionné le portefeuille, ainsi qu'un bouton pour retourner sur notre portefeuille. J'ai également modifié la topbar du site pour préciser lorsque l'on est sur un autre portefeuille que le nôtre, ainsi on peut savoir en permanence si on est sur notre portefeuille ou non.



Alain GAZONI connecté sur le portefeuille de Ali TAZI

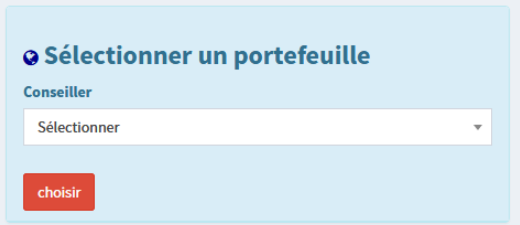
Indication du portefeuille sélectionné dans la topbar

Pour finir j'ai ajouté une fonction `disconnect` au `PortefeuilleController`, ainsi que la route correspondante, qui sera appelée par le bouton que j'ai ajouté sur la page d'accueil. Cette fonction se charge de réinitialiser l'id du conseiller dans la variable de session, puis nous renvoie sur la page d'accueil.



Vous êtes connecté sur le portefeuille de Ali TAZI

Se déconnecter du portefeuille



Sélectionner un portefeuille

Conseiller

Sélectionner

choisir

Formulaires de déconnexion et de changement de portefeuille

4. Widgets sur la page d'accueil

Plus tard durant mon stage, on m'a demandé d'ajouter des widgets sur la page d'accueil pour fournir des informations clés aux conseillers. Dans un premier temps, on m'a demandé de créer deux widgets : un premier pour résumer l'activité du portefeuille sur le mois en cours et un second pour donner la répartition des contrats du portefeuille par type de risque.

J'ai donc commencé par le widget de résumé du portefeuille. J'ai tout d'abord ajouté une requête SQL qui récupère l'activité du portefeuille sur le mois dans la fonction `index` du `HomeController`, en donnant une date de début et de fin, à l'image du tableau de bord "Activité par date". Puis, j'ai utilisé la fonction `calculs` de `CalculsPortefeuille` ([cf. J.1.](#)) avec les produits obtenus pour récupérer les totaux et les passer à la vue. J'ai ensuite ajouté le widget à la vue. Celui-ci prend la forme d'un panel Bootstrap indiquant le nombre de contrat sur le mois, les primes périodiques et uniques encaissées ainsi que les commissions d'acquisition et sur les primes uniques.

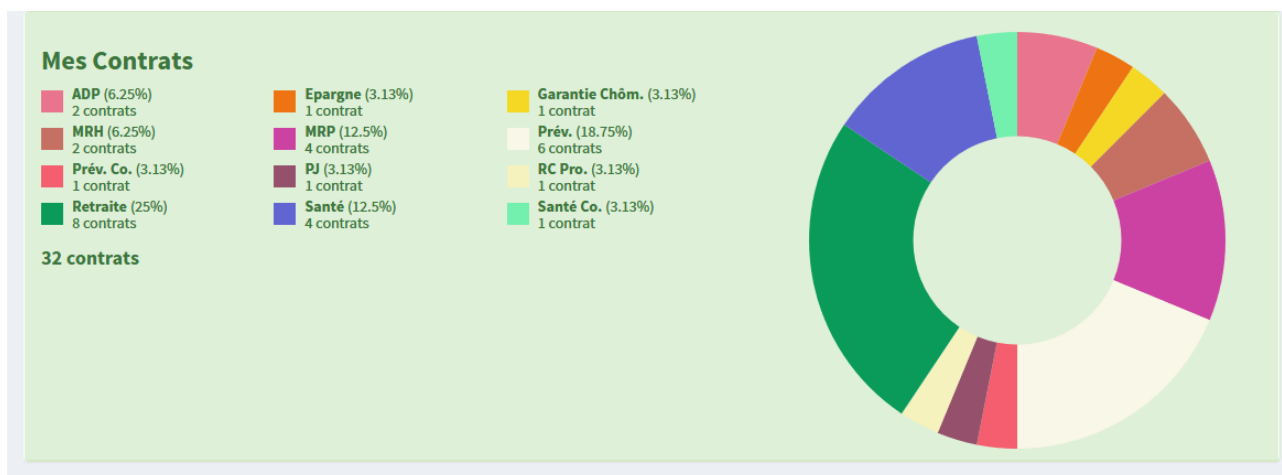
Après réflexion, j'ai proposé à mon maître de stage d'ajouter le détail des contrats, à savoir le nombre de nouvelles affaires, d'avenants d'augmentation et de diminution, de versements uniques, etc. J'ai alors ajouté une requête pour récupérer le compte de chaque type d'événement produit avec le nom du type, puis j'ai modifié le widget pour qu'il affiche les nouvelles données.



Widget résumant l'activité du mois en cours

Par la suite, je me suis occupé du widget donnant la répartition des contrats par type de risque. J'ai donc ajouté une requête SQL pour compter le nombre total de contrats dans le portefeuille et une seconde pour compter le nombre de contrats par type de risque. J'ai ensuite ajouté le widget à la vue. Mon maître de stage souhaitant l'utilisation d'un graphique pour ces données, j'ai mis en place un diagramme circulaire grâce à la librairie `JavaScript Chart.js`.

La librairie demandant de définir des couleurs pour l’affichage des items, j’ai modifié la table `types_produits` dans la base de données ainsi que le modèle pour y ajouter une colonne `color`, contenant le code couleur utilisé. J’y ai également ajouté une colonne `nomCourt` qui sera utilisée pour la légende du graphique, le nom de certains types étant trop long pour un affichage correct. J’ai ensuite modifié les formulaires d’ajout et d’édition pour prendre en compte les champs ajoutés, le champ pour la couleur sera généré par l’extension `ColorPicker` de Bootstrap.



Widget de la répartition des contrats par type de risque

Pour la légende, les items sont affichés avec leur couleur, le nom court du type, le pourcentage et le nombre de contrat que cela représente. Dans un premier temps les items étaient disposés sur une seule colonne et le widget était aligné avec celui de l’activité du mois. Je me suis alors rendu compte que le nombre d’items moyen était plus important en production qu’en développement, et nécessitait plusieurs colonnes pour éviter que le widget soit trop haut. J’ai donc modifié l’affichage de la légende pour que celle-ci répartisse



Widget des démarches commerciales

les items sur plusieurs colonnes. La légende prenant plus de place, j’ai modifié le widget pour que celui-ci prenne toute la largeur de la page. Ce widget n’est donc plus aligné avec le premier.

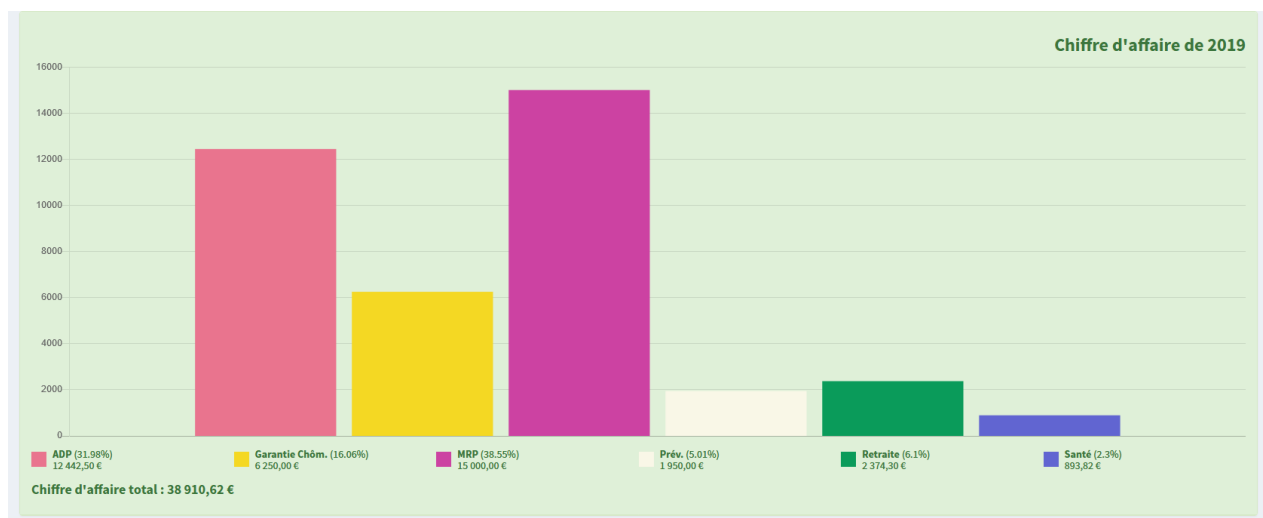
Afin de combler le vide laissé à côté du widget de l’activité et suite à une demande de la secrétaire de mon maître de stage, on m’a demandé d’ajouter un

autre widget indiquant les démarches commerciales en attente. J'ai donc ajouté une requête qui récupère le nombre d'événements pour chaque type de démarche demandé (Adhésion, Courrier, Envoi, Gestion, Projet, Mail, Mail reçu, Édition) ainsi que le nom de ces démarches. J'ai ensuite ajouté le widget dans la vue. Celui-ci indique le nombre total de démarches en attente, puis donne le détail pour chaque type de démarche. Le widget contient également un bouton pour accéder rapidement au tableau de bord "Démarches commerciales".

Suite à une demande de l'ACPR, l'organisme chargé de surveiller l'activité des banques et des assurances en France, souhaitant avoir la répartition du chiffre d'affaire en fonction des types de contrat, on m'a demandé de mettre en place un autre widget. Ce widget devait donc donner le chiffre d'affaire de l'année n-1 type par type.

J'ai donc ajouté des requêtes SQL dans la fonction index du HomeController pour obtenir le total d'acquisition en première année, de commissions sur les primes périodiques et uniques pour chaque type de contrat. Le total de ces trois valeurs donne donc le chiffre d'affaire pour le type. Les données sont affichées dans le widget avec un histogramme généré avec Chart.js. Le graphique prend donc la largeur de la page et la légende est affichée en dessous, avec une génération similaire à celle de l'autre widget affichant un graphique. Pour ce widget, le pourcentage est calculé par rapport au chiffre d'affaire pour le type donné et au chiffre d'affaire total de l'année n-1.

Afin d'alléger le contenu et le chargement de la page d'accueil, on m'a demandé plus tard de déplacer ce widget dans une page dédiée, accessible depuis les tableaux de bord. J'ai donc créé le ChiffreAffaireController avec pour seule fonction index, qui se charge de récupérer les données auparavant récupérées dans le HomeController. J'ai ensuite ajouté une vue qui n'affiche donc que ce widget.



Widget du chiffre d'affaire

H. Ajout des besoins clients

Dans la version initiale de Cercle SPA, lorsqu'un conseiller ajoute des besoins pour un client, il ne peut pas voir ceux qui ont déjà été ajoutés en étant sur la page de sélection des besoins et cela cause des doublons dans la base de données. On m'a donc chargé de modifier cette fonctionnalité pour en améliorer l'ergonomie et éviter ces doublons.

Dans un premier temps, j'ai mis en place la visualisation des besoins déjà ajoutés. Pour cela, j'ai mis en place une requête AJAX récupérant les besoins déjà présents pour ce type de besoin et ce client. La requête se charge ensuite de cocher dans le formulaire les besoins récupérés. J'ai donc ajouté la route et la fonction `getBesoinParClient($type, $client)`, appelée par la requête AJAX, dans le `BesoinParClientsController`.

J'ai ensuite remplacé les checkboxes de la vue par des boutons radio pour que le conseiller ne puisse sélectionner qu'une seule occurrence par besoin. Une fonction JavaScript permet de totalement décocher les boutons d'un groupe, car cela n'est pas possible de base, on peut ainsi supprimer un besoin exprimé directement depuis le formulaire.

Par la suite, j'ai modifié la fonction `store` du `BesoinParClientsController` pour qu'elle puisse gérer l'ajout ou la modification d'un besoin, sans doublon, ainsi que la suppression depuis le formulaire d'un besoin précédemment exprimé.

Pour chaque besoin, la fonction essaie de récupérer l'entrée dans la base de données à partir du type de besoin, du client et du besoin. Si elle la récupère et que l'occurrence sélectionnée diffère de l'occurrence enregistrée, elle met à jour l'entrée avec la nouvelle occurrence choisie, sinon elle ne fait rien. Et c'est uniquement si aucune entrée n'est

bénéficier d'une rente avec des annuités garanties sur une durée déterminée

☐ OUI

☐ sur au moins un des dossiers cités ci après

assurer une rente de reversion à votre conjoint

☒ jusqu'à 100%

☐ jusqu'à 200%

☐ à 100%

☐ à 60%

bénéficier d'un cadre fiscal particulier

☐ plus précisément dans le double cadre fiscal de l'article 163 du CGI (PERP) et de l'Assurance Vie.

☐ article 154bis

☒ article 83

☐ article 163

☐ article 82

☐ article 39

☐ assurance Vie

☐ Ass.Vie + article 154bis

Page de sélection des besoins exprimés

trouvée que la fonction ajoute le besoin dans la base de données. Ainsi, il ne peut pas y avoir de doublon. Pour finir, la fonction se charge de supprimer les besoins pour lesquels le conseiller aurait totalement décoché les boutons.

I. Réalisateurs et Apporteurs dans l'activité par date

Cercle SPA possède un tableau de bord "Activité par date" qui permet de visualiser l'activité de son portefeuille entre deux dates données. On m'a demandé de l'améliorer afin d'indiquer pour chaque contrat du portefeuille l'apporteur et l'apporteur interne et d'ajouter un filtrage par apporteur (interne). Cela permettra aux conseillers de visualiser rapidement l'apport de chacun.

Dans un premier temps, j'ai ajouté les colonnes "Apporteur" et "Apporteur Interne" à la vue puis j'ai modifié la fonction `res` du `MonactiviteController` afin que la requête SQL qui récupère les contrats récupère également les apporteurs de ces contrats. Pour ce faire j'ai ajouté des jointures externes (afin de conserver les contrats sans apporteurs) sur la table `conseillers` pour les apporteurs et sur la table `apporteurs` pour les apporteurs internes.

Ensuite, je me suis occupé des filtres. Pour cela, j'ai commencé par récupérer la liste des apporteurs et apporteurs internes présents dans les contrats pour remplir les menus déroulants. J'ai donc ajouté deux formulaires à la vue, un pour sélectionner un apporteur et l'autre pour sélectionner un apporteur interne. Ils renvoient tous les deux vers la page en elle même. Les deux formulaires contiennent des champs cachés dans lesquels sont mis les dates de début et de fin définies auparavant, ainsi la requête contient l'identifiant d'un apporteur et les dates déjà définies. J'ai de nouveau modifié la fonction `res` pour qu'elle récupère `l'apporteur_id` et `l'apporteurint_id` de la requête et pour qu'elle ajoute une condition à la requête SQL si un apporteur ou un apporteur interne est sélectionné. Enfin, j'ai modifié le titre affiché dans la vue pour que celui-ci indique la fourchette de date sélectionnée et l'apporteur sélectionné si il y en a un.

Par la suite, on m'a demandé d'ajouter le réalisateur en plus des apporteurs dans les filtres, car même si par défaut le réalisateur est le conseiller ajoutant le contrat, parfois il se peut qu'il s'agisse d'un autre conseiller. On m'a également demandé de rendre l'affichage des filtres possible uniquement si la colonne `modeagence` du portefeuille connecté est à vrai, pour cet outil soit une option du logiciel.

Plus tard, on m'a demandé de faire également cet ajout pour le tableau de bord "Valeur portefeuille".

Filtrer par Réalisateur

Réalisateur

Sélectionner

Choisir

Filtrer par Apporteur

Apporteur

ANTOINE Marc

Choisir

Filtrer par Apporteur Interne

Apporteur interne

Sélectionner

Choisir

Activité commerciale du 01/01/2019 au 31/12/2019 pour l'apporteur
ANTOINE Marc

Retour à l'export global

Afficher 100 éléments Copier CSV Excel PDF Imprimer Visibilité de la colonne Rechercher :

Compagnie	Produit	Client	Début du contrat	Statut	Date signature	Date Effet	Situation contrat	Réalisateur	Apporteur	Apporteur interne	PP Init €	PU Init €	Tx
SWISSLIFE	SwissLife Retraite Madelin N°	Dupont Vanessa (50 ans)	04/06/2018	Avenant Augmentation	15/05/2019	15/05/2019	En cours (Attente Emission) Non Defini	GAZONI Alain	ANTOINE Marc	EYGONNET Christophe	16000		1
ACS	ACS TNS N°	Carbonnier Nicolas (20 ans)	09/06/2019	Affaire Nouvelle	09/06/2019	09/06/2019	En cours (Attente Emission) Non Defini	GAZONI Alain	ANTOINE Marc		6000	755	0.8

Affichage de l'élément 1 à 2 sur 2 éléments

Précédent 1 Suivant

Nombre de contrats	Primes périé encaissées (année en cours)	Primes uniques encaissées	Com Acquisition PP	Com Linéaire PP	Com Acquisition PU	Encours PU €	Encours PU UC
2	10 000,00 €	755,00 €	2 660,57 €	180,80 €	113,25 €	0,00 €	0,00 €

Activité par date filtrée sur l'apporteur Marc ANTOINE

J. Valeur du portefeuille, primes et événements produit

Après avoir totalement fini l'exportation des calendriers, on m'a confié plusieurs tâches liées au portefeuille du conseiller, notamment par rapport aux calculs des estimations de la valeur du portefeuille.

1. Déplacement des calculs

Dans la version initiale de Cercle SPA, les calculs étaient effectués directement dans le fichier vue pour les tableaux de bord affichant une estimation du portefeuille, à savoir "Valeur portefeuille", "Activité par date" et "Filtrer mes contrats". On m'a donc demandé de déplacer ces calculs, afin d'avoir un code mieux organisé.

Dans un premier temps, j'ai déplacé tous les calculs dans les fonctions associées aux tableaux de bord, à savoir les fonctions res et compagnieres du MescontratsController, la fonction res du MonactiviteController et la fonction index du ValueController.

Certains calculs étant faits dans une boucle foreach dans la vue, j'ai dû construire une structure de données afin de passer tous les résultats à la vue et de pouvoir les afficher

correctement. Pour cela, j'ai créé un tableau d'objets `calculs` contenant les résultats des calculs pour chacun des contrats, ainsi qu'un objet `totaux` contenant les résultats finaux, dont la valeur du portefeuille. Le tableau `calculs` étant rempli dans l'ordre des contrats, il m'a suffi alors de modifier la boucle `foreach` de la vue pour qu'elle soit de la forme `index => objet` afin d'utiliser l'`index` pour récupérer les résultats associés au contrat affiché.

Je me suis ensuite rendu compte de la requête SQL récupérant les contrats utilisant le premier événement produit de chaque contrat. Hors, un contrat peut posséder plusieurs événements, dans lequel les montants de primes peuvent changer. Il est alors important d'utiliser le dernier événement pour chaque contrat afin d'utiliser les bons montants. J'ai donc modifié la jointure avec la table `evenementsparproduits` pour qu'elle se fasse sur le dernier événement et non le premier.

En vue d'améliorer les calculs, j'ai créé une classe `CalculsPortefeuille` avec une fonction statique `calculs($produits)`, réalisant tous les calculs et retournant le tableau `calculs` et l'objet `totaux`, ainsi qu'un attribut statique `select` contenant les attributs à sélectionner dans les requêtes SQL des différents contrôleurs. Grâce à cela, je peux modifier le fonctionnement des calculs sans avoir à modifier chacun des contrôleurs un à un.

2. PASS

La montant de la prime périodique (PP) augmentant chaque année, on a cherché avec mon maître de stage une solution pour automatiser cette augmentation annuelle sans avoir à modifier le montant de la prime en allant sur chaque contrat. Certaines primes étant indexées sur le PASS (Plafond Annuel de la Sécurité Sociale), qui augmente chaque année, on a décidé de mettre en place le calcul de la PP en se basant sur le PASS.

Dans un premier temps, j'ai mis en place la table pour stocker le PASS pour chaque année, ainsi que le modèle associé, le contrôleur et les vues utiles pour ajouter un PASS. J'ai donc créé une table `pass` contenant une colonne `annee` de type `YEAR` et une colonne `pass` de type entier. J'ai ensuite ajouté le contrôleur `PassController` qui possède les fonctions `index`, `create` et `edit` pour les vues, puis j'ai créé les vues associées. J'ai ensuite ajouté un accès à l'`index` du PASS dans la sidebar, puis j'ai renseigné le PASS de 2000 à 2020 et j'ai exporté la table complétée pour que mon maître de stage puisse facilement l'ajouter à la base de données du serveur de production.

Par la suite, j'ai modifié le mutateur du montant de la prime périodique dans le modèle `Evenementsparproduit` afin que le montant renseigné dans le formulaire soit

divisé par le PASS de l'année en cours. J'ai également modifié l'accesseur pour qu'il multiplie la valeur enregistrée dans la base de données par le PASS, afin de retrouver le montant de la PP. Ainsi, la valeur stockée reste la même, mais le montant augmentera naturellement car il est indexé sur le PASS.

Finalement, l'indexation sur le PASS s'est révélé être un cas particulier pour certaines compagnies, ce n'était donc pas un moyen fiable pour augmenter automatiquement les montants de la prime périodique et fut donc abandonné. Malgré cela, on a conservé le code créé, au cas où il pourrait se révéler utile plus tard. Le montant de la prime période est donc de nouveau stocké directement dans la base de données, sans être divisé par le PASS.

3. Mise à jour des Primes

L'utilisation du PASS étant abandonnée et une automatisation n'étant au final pas envisageable du fait des nombreux cas particulier qu'il existe, nous avons finalement pris la décision de mettre en place un outil pour pouvoir modifier manuellement les primes périodiques sans avoir à consulter les contrats un à un.

J'ai donc créé le contrôleur PrimeController qui possède les fonctions create, list et update. La première fonction retourne une vue qui permet de filtrer les contrats du portefeuille par type, puis par compagnie et par produit.

La fonction list se charge de retourner la vue contenant les contrats obtenus, présentés dans un tableau, qui est un formulaire dans lequel l'utilisateur peut modifier le montant de la prime périodique pour chacun des contrats. La valeur des champs textes est passée à la fonction update dans un tableau produit avec un index correspondant à l'id du dernier événement produit du contrat.

Liste								
Type	Compagnie	Produit	Client	Statut	Date Effet	Situation contrat	Montant PP €	
Assurance De Prêt	AVIVA	AVIVA Emprunteur N° numero bis	Dupont michel	Avenant Diminution	03/06/20	En cours (Attente Emission) Non Defini	12000	<input type="text" value="12000"/>
Assurance De Prêt	AVIVA	AVIVA Emprunteur N° 102	Entreprise Audrey	Avenant Diminution	02/06/20	En cours sous prime Annuel	20000	<input type="text" value="20000"/>

Mettre à jour

Liste des primes sur les contrats en Assurance de Prêt, avec un champ pour les mettre à jour

Ainsi, la fonction update se charge alors d'ajouter les événements produits pour les contrats dont la PP a été modifiée. Pour chacun des produits du tableau, la fonction récupère le dernier événement produit grâce à son id. Si le nouveau montant est différent de l'ancien, la fonction crée un nouvel événement produit, en récupérant certaines informations de l'événement précédent, notamment le numéro de produit, ainsi que les réalisateurs et apporteurs. Pour déterminer si il s'agit d'un "Avenant Augmentation" ou un "Avenant Diminution", on compare la PP précédente avec la nouvelle.

4. Pourcentage de variation et nouveaux événements produit

Afin d'aider les conseillers à se faire une idée de la variation de la prime périodique, j'ai mis en place l'indication du pourcentage de variation dans la liste des événements produit. Pour cela, j'ai modifié la vue d'édition d'un contrat, car c'est dans cette dernière qu'est affichée la liste des événements du produit. Pour chaque événement autre que le premier, je calcule donc le pourcentage d'augmentation ou de diminution de la prime périodique par rapport à la PP précédente.

Dans la continuité des ajouts d'événements liés à la modification des primes ([cf. J.3.](#)), j'ai modifié la fonction store du `EvenementsparproduitsController` pour qu'elle récupère les informations du dernier événement du produit en question afin de les utiliser pour le nouvel événement créé. Cela permet de faire gagner du temps aux conseillers, qui n'auront alors qu'à modifier les éléments qu'il faut dans le nouvel événement, sans avoir à ajouter des informations déjà renseignées comme les apporteurs ou le choix de commission.

Ajouter un événement sur ce produit					
Événements liés au produit					
Événement	Date effet	Formalisé	Montant PP	Montant PU	
Affaire Nouvelle	31/05/2011	<input checked="" type="checkbox"/>	10000	10000	Afficher les détails Supprimer
Avenant Augmentation	16/06/2020	<input type="checkbox"/>	12000 (+ 2000) (+ 20% par rapport à la PP précédente)	()	Afficher les détails Supprimer
Versement Unique	16/06/2020	<input type="checkbox"/>	(12000)	667	Afficher les détails Supprimer

Liste des événements d'un produit, avec le pourcentage de variation et la mise en valeur des montants clés

Par la suite j'ai ajouté la valeur de variation en plus du pourcentage, et j'ai modifié la mise en page pour les valeurs modifiées pour chaque événement ressortent plus. Ces ajouts permettent une meilleure compréhension des informations pour les utilisateurs.

5. Amélioration des estimations

Dans la version initiale de Cercle SPA, le calcul des commissions touchées sur les primes périodiques et les primes uniques (PU) était fait de manière assez générique, à savoir en multipliant la prime par le taux de taxe et le taux de commission. Hors, certaines compagnies ont une manière beaucoup plus précise et spécifique pour calculer les commissions. On m'a donc chargé de modifier les calculs pour les traiter au cas par cas pour plus de précision dans les estimations.

Dans un premier temps, je me suis occupé des contrats Retraite pour Aviva et SwissLife. Avec un calculateur fait sur Excel que m'a fourni mon maître de stage, j'ai pu mettre en place les calculs spécifiques à ces deux types de contrats. Ces derniers nécessitant des informations manquantes dans les protocoles de commission, telles que les taux d'application pour la prime unique ou les seuils d'investissement en euro et en unité de compte, j'ai ajouté des colonnes à la table commissionsparproduits, puis j'ai modifié les vues pour l'ajout et la modification des protocoles afin de prendre en compte ces nouveaux attributs.

Dans la version initiale des calculs, les commissions calculées étaient dans tous les cas ajoutées aux récapitulatifs annuels des escomptes (première année des contrats), des commissions linéaires et des PU. Hors, la commission sur l'escompte et sur la PU ne sont, en règle générale, touchées que la première année, tandis que la commission linéaire est touchée après la période d'escompte. J'ai donc fait les modifications nécessaires afin d'ajouter les commissions aux récapitulatifs que sous certaines conditions, en se basant notamment sur l'année de début du contrat et la période d'escompte.

Par la suite, j'ai ajouté le détail des surcommissions sur la PP et la PU aux tableaux "Valeur de portefeuille" et "Activité par date" et j'ai amélioré l'affichage afin que les utilisateurs sachent quelles valeurs de chaque contrat ont été prises en compte pour le calcul des totaux. Puis j'ai amélioré les calculs pour trouver d'éventuels autres événements produit sur la période donnée, afin de prendre en compte, par exemple, des avenants d'augmentations ou des versements uniques qui ont pu être fait sur cette période. J'ai ensuite mis en place les calculs pour les contrats Retraite de Generali et Cardif.

PP Init €	PU Init €	Tx Taxe	% Esc	% lin	Surco PP €	Com acq. PP €	Com lin PP €	Surco PU €	Com acq. PU €	Encours PU €	Encours PU Uc
20000	1000	0.91	45	15	(0)	(0)	2730	0	150	(0)	(0)
15000	7500	0.8673	10	10	(0)	1300.95	(1300.95)	0	750	(0)	(0)
12000	667	0.81	22.5	4.5	523.8	364.5	437.4	17.47	30.02	(0)	(0)
5000		0.8673	15	15	(0)	(0)	650.48	0	0	(0)	(0)

Détail des commissions des contrats, avec la mise en valeur des commissions comptabilisées sur la période choisie

6. Anomalies dans les Avenants

Dans la version initiale de Cercle SPA, lorsqu'un conseiller renseignait un avenant d'augmentation ou de diminution, celui-ci indiquait uniquement la valeur de variation. Hors, avec les modifications des calculs sur le portefeuille, il était nécessaire que la valeur renseignée corresponde à l'ancien montant de la PP plus ou moins cette variation. Une indication pour l'expliquer aux utilisateurs a été mise en place, afin qu'ils ne fassent pas l'erreur pour les nouvelles saisies.

Mais pour pouvoir corriger rapidement les événements produits déjà présents, on m'a demandé de mettre en place une page permettant de visualiser et de corriger tous les événements produits ayant une erreur.

Je me suis donc basé sur la page que j'avais mis en place pour mettre à jour les primes ([cf. J.3.](#)) et j'ai créé le contrôleur `AnomaliesAvenantsController` contenant les fonctions `list` et `update`. Pour trouver les anomalies, je commence par récupérer tous les avenants du portefeuille conseiller, puis je récupère tous les événements précédents aux avenants afin d'obtenir l'ancien montant de la prime périodique. Pour filtrer les avenants ayant une erreur, je teste si la PP est négative pour un avenant de diminution et je teste si la PP est plus petite que la PP de l'événement précédent pour un avenant d'augmentation.

Je passe ensuite les anomalies à la vue qui se charge de les afficher en indiquant l'ancien montant de la PP, ainsi que celui indiqué dans l'avenant et qui affiche un champ texte contenant la suggestion de correction de la prime, qui correspond au total de l'ancien

montant et du montant de l'avenant. Pour un avenant de diminution, la valeur étant négative, on obtient bien une suggestion inférieure à l'ancienne PP.

Anomalies Avenants

Si des anomalies subsistent après la mise à jour, veuillez corriger vos contrats à la main. Il s'agit sûrement d'une erreur de saisie indépendante dans l'ancienne méthode de saisie.

Liste

Avenants Diminutions

Type	Compagnie	Produit	Client	Date Effet	Situation contrat	Ancien montant PP €	Montant PP €		Corriger
Garantie Chômage	GSC - GAN EUROCOURTAGE	Garantie Chômage N°	Carbonnier Nicolas	22/06/20	En cours (Attente Emission) Non Defini	50000	-10000	40000	<input checked="" type="checkbox"/>
Santé	ACS	ACS TNS N°	Carbonnier Nicolas	22/06/20	En cours (Attente Emission) Non Defini	6000	-1000	5000	<input checked="" type="checkbox"/>

Avenants Augmentation

Type	Compagnie	Produit	Client	Date Effet	Situation contrat	Ancien montant PP €	Montant PP €		Corriger
Retraite	AVIVA	AVIVA Retraite Plurielle Madelin N°	Carbonnier Nicolas	16/06/20	En cours (Attente Emission) Non Defini	10000	2000	12000	<input checked="" type="checkbox"/>

Mettre à jour

Liste des anomalies sur les avenants avec une proposition de correction

Pour mettre à jour les événements erronés, je commence par les trier par ordre croissant d'id, afin de les corriger par ordre chronologique, puis je calcule le montant de variation pour chacun des événements, au cas où des suggestions aient été modifiées et que la variation ne corresponde plus au montant de l'avenant.

Je boucle ensuite sur chaque événement pour le récupérer ainsi que le montant de l'ancienne PP, puis je met à jour la PP de l'avenant avec pour valeur la somme de l'ancienne PP et de la variation. Afin que l'avenant s'applique également sur les événements suivants du produit et que l'on soit sûr de tomber sur le bon montant sur le dernier événement, je met également à jour les événements suivants avec ce montant calculé. Il était important de récupérer l'ancienne PP et de ne pas utiliser celle passée dans la requête car elle peut être amenée à changer à cause de l'ajout des avenants aux événements suivants.

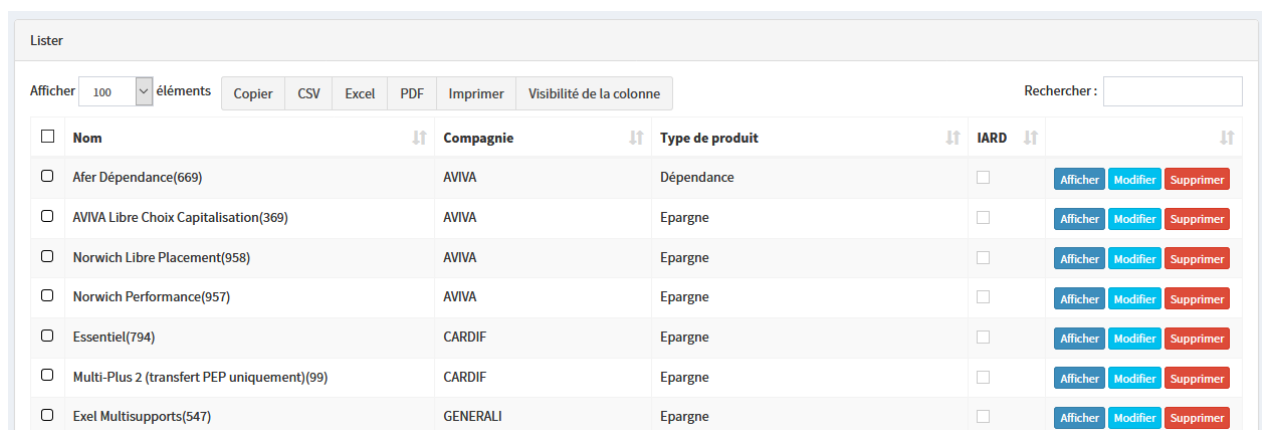
K. Protocoles de commissions

En parallèle des travaux sur les calculs du portefeuille conseiller, on m'a confié plusieurs tâches liées aux protocoles de commissions, nécessaires aux calculs.

1. Produits sans protocoles

Les produits commercialisés par le regroupement disposent de différents protocoles de commissions permettant de calculer le chiffre d'affaire d'un portefeuille. Ces protocoles sont donc essentiels car les contrats sans protocole défini n'apparaissent pas dans les tableaux de valorisation du portefeuille. Afin d'identifier les produits n'ayant pas encore de protocoles on m'a confié la mise en place d'une page listant ces derniers.

Pour se faire, j'ai mis en place la fonction `withoutProtocole` dans le `ProduitsController`, qui effectue une requête SQL récupérant tous les produits pour lesquels aucune `commissionsparproduits` n'a été définie et qui retourne une vue similaire à celle utilisée pour la liste des produits. J'ai ensuite ajouté la route et un accès à cette page depuis la sidebar.



<input type="checkbox"/>	Nom	Compagnie	Type de produit	IARD	
<input type="checkbox"/>	Afer Dépendance(669)	AVIVA	Dépendance	<input type="checkbox"/>	Afficher Modifier Supprimer
<input type="checkbox"/>	AVIVA Libre Choix Capitalisation(369)	AVIVA	Epargne	<input type="checkbox"/>	Afficher Modifier Supprimer
<input type="checkbox"/>	Norwich Libre Placement(958)	AVIVA	Epargne	<input type="checkbox"/>	Afficher Modifier Supprimer
<input type="checkbox"/>	Norwich Performance(957)	AVIVA	Epargne	<input type="checkbox"/>	Afficher Modifier Supprimer
<input type="checkbox"/>	Essentiel(794)	CARDIF	Epargne	<input type="checkbox"/>	Afficher Modifier Supprimer
<input type="checkbox"/>	Multi-Plus 2 (transfert PEP uniquement)(99)	CARDIF	Epargne	<input type="checkbox"/>	Afficher Modifier Supprimer
<input type="checkbox"/>	Exel Multisupports(547)	GENERALI	Epargne	<input type="checkbox"/>	Afficher Modifier Supprimer

Liste des produits sans protocoles

Dans la continuité, j'ai mis en place un protocole générique disponible pour tous les produits afin que l'on puisse renseigner un protocole même pour les contrats sur des produits qui n'en disposent pas encore. Ainsi les contrats concernés peuvent apparaître dans la valorisation du portefeuille.

Suite à cet ajout et au fait que la date de signature est utilisée à la place de la date d'effet dans la sélection des contrats, j'ai créé un script SQL pour modifier la table `evenementsparproduits`. Pour les entrées n'ayant pas de `NumCom` et de `DateSignature`, le `NumCom` est défini avec l'id du protocole générique et la `DateSignature` avec la `DateEffet`. Ainsi les conseillers n'ont pas eu besoin de modifier tous leurs contrats un à un pour les voir apparaître dans les tableaux de bord dédiés.

Afin de gagner un peu de précision lors de l'utilisation du protocole générique, on m'a par la suite demandé de mettre en place l'indication du taux de taxe directement dans les événements produits. J'ai donc commencé par ajouter une colonne TauxTaxe à la table evenementsparproduits, puis j'ai modifié le formulaire d'édition.

Choix de commissions	Taux taxe/Coef pond. (en %)
Générique (2020) => Escompte : 1 - Linéaire : 1	

Indication du taux de taxe lors de l'utilisation du protocole générique

Le formulaire affiche donc un champ pour le taux de taxe à côté de la liste de sélection du

protocole. Ce champ s'affiche uniquement lorsque le protocole générique est sélectionné et se réinitialise si un protocole propre au produit est choisi. Ainsi, si ce n'est pas le protocole générique qui est choisi, le taux de taxe dans l'événement est effacé et c'est le taux renseigné dans le protocole qui est utilisé. Pour finir, j'ai modifié les calculs pour prendre en compte ce taux de taxe lorsqu'il est renseigné.

2. Taux de commissions négociés

Sur certains contrats, comme en Santé ou en Prévoyance par exemple, les conseillers ont la possibilité de négocier les taux de commissions qu'ils toucheront. Jusque là, ce genre de produit disposait d'une cinquantaine de protocoles avec pour seules différences les taux de commissions en escompte et en linéaire. Pour éviter cette redondance de protocoles, on m'a demandé de mettre en place l'ajout de taux de commissions négociés dans les événements produit.

Pour commencer, j'ai ajouté les colonnes ComEsc et ComLin à la table evenementsparproduits. J'ai ensuite modifié la classe CalculsPortefeuille pour que ces nouveaux attributs soient pris en compte et soient utilisés dans les calculs à la place des taux renseignés dans le protocole. J'ai également modifié les vues de "Valeur Portefeuille" et "Activité par date" pour afficher ces taux à la place de ceux du protocole lorsqu'ils sont définis. Pour finir, j'ai modifié le formulaire d'édition des événements produit pour qu'il affiche des champs pour renseigner les taux négociés uniquement pour certains types de contrats.

Choix de commissions	Taux de commission négociés (en %):	Com Esc	Com Lin
SPA (2017) - => Escompte : 15 - Linéaire : 15		10	12

Indication des taux de commissions négociés

3. Importation des protocoles avec un fichier CSV

Le renseignement des protocoles pour un nouveau produit est une tâche relativement longue. Afin d'alléger cette tâche, on m'a demandé de mettre en place l'importation des protocoles via un fichier CSV.

Le projet disposait déjà d'une fonction d'importation CSV, dans le `CsvImportController`, utilisées dans une autre partie du logiciel. J'ai donc adapté cette dernière pour fonctionner avec les protocoles. Ce contrôleur dispose de deux fonctions : `parse` et `process`. La première se charge de lire le fichier, de récupérer les champs du modèle renseigné et de retourner une vue listant les champs et les valeurs fournies dans le fichier pour ces derniers. Cette page intermédiaire nous permet de vérifier les données avant l'importation. La seconde fonction, quant à elle, se charge de construire le tableau de données pour chaque nouvelle entrée et d'enregistrer ces entrées dans la base de données.

J'ai donc commencé par modifier la fonction `parse` pour qu'elle puisse lire plusieurs lignes, celle-ci étant limitée à deux lignes au départ. J'ai également ajouté la récupération de l'id du produit auquel on ajoute les protocoles afin d'éviter à l'utilisateur de le renseigner dans chaque ligne de son fichier CSV. J'ai ensuite modifié la seconde fonction pour qu'elle ajoute l'attribut `numprd` avec pour valeur l'id du produit à chaque entrée lorsque le modèle est `Commissionsparproduit`. Ainsi ce contrôleur reste fonctionnel pour les autres parties du logiciel qui l'utilise et s'adapte au cas particulier des protocoles de commissions.

The screenshot shows a web interface titled "Importer CSV.". It contains a table with the following data:

detailExplicatif	protocoleAnnee	protocoleNom	protocoleCommentaire	tauxTaxe	repriseMois	avenantMois	comEsc	comLin
	2020	Test		81			20	15
	2020	Test 2		72			15	12
	2020	Test 3		60			14	10

Below the table, there are dropdown menus for each column header: `detailexplicatif`, `protocoleannee`, `protocolenom`, `protocolecommentaire`, `tauxtaxe`, `reprisemois`, `avenantmois`, `comesc`, and `comlin`. At the bottom left, there is a blue button labeled "Importer les données".

Page intermédiaire de l'importation CSV permettant de vérifier les données avant de les enregistrer dans la base

Juste après, mon maître de stage m'a demandé d'adapter cette fonctionnalité pour pouvoir importer des Clients.

L. Profil investisseur

Après avoir bien progressé sur les calculs pour le portefeuille et sur les protocoles de commissions, on m'a demandé de mettre en place une nouvelle fonctionnalité sur les fiches clients : le profil investisseur. À l'origine, il s'agit d'un questionnaire papier destiné à évaluer un client sur l'aspect financier, pour lui conseiller un comportement à adopter dans la gestion d'un investissement.

On m'a donc confié la mise en place du questionnaire, de l'évaluation et de la génération du document PDF résumant les réponses et résultats du client.

1. Nouvelles tables dans la base

Cette fonctionnalité étant totalement nouvelle, j'ai tout d'abord dû réfléchir à la mise en place de nouvelles tables dans la base de données pour stocker les résultats pour chaque client, ainsi que les questions, leurs occurrences et le nombre de points associé.

J'ai donc créé la table `profil_investisseurs`, identifiée par un `id` et l'`id` du client concerné. Cette dernière contient notamment les réponses à chaque question, sous forme d'un booléen lorsqu'il s'agit d'une question "Oui/Non" et sous la forme d'un entier correspondant à l'`id` d'une occurrence lorsqu'il s'agit d'une question à plusieurs occurrences. Elle contient également un entier `score`, calculé suite au questionnaire, ainsi que les suggestions et choix de comportement adoptés par le client.

Pour les questions, j'ai créé les tables `profil_investisseur_questions` et `profil_investisseur_occurrences`. La première contient les questions avec leur étape, leur ordre d'apparition dans l'étape, l'attribut de la table `profil_investisseurs` associé à la question et la question telle qu'elle sera posée dans le questionnaire. La seconde table contient les occurrences de chaque question, identifiées par la question associée, leur ordre, leur nom, leur nombre de points, et pour certaines, d'autres champs tel que `lettre` pour la question sur la sensibilité, car cette dernière est identifiée avec les lettres A, B ou C.

2. Création du profil

Une fois les tables nécessaires créées, j'ai mis en place le formulaire de création du profil. Mon maître de stage souhaitant un formulaire progressant étape par étape (identifiées comme sur la version papier), j'ai utilisé l'extension jQuery Steps de jQuery.

Création du profil investisseur de **Vanessa Dupont**

Connaissances Investissement Sensibilité Principe de gestion

Connaissances et expériences de l'épargnant

Êtes-vous un professionnel de la finance ? ☐ Oui ☐ Non

	Connaissez-vous ce support ?	Avez-vous déjà investi dessus ?
<input type="checkbox"/> Action	<input type="radio"/> Oui <input type="radio"/> Non	<input type="radio"/> Oui <input type="radio"/> Non
<input type="checkbox"/> Obligations	<input type="radio"/> Oui <input type="radio"/> Non	<input type="radio"/> Oui <input type="radio"/> Non
<input type="checkbox"/> OPCVM	<input type="radio"/> Oui <input type="radio"/> Non	<input type="radio"/> Oui <input type="radio"/> Non
<input type="checkbox"/> Fonds structures	<input type="radio"/> Oui <input type="radio"/> Non	<input type="radio"/> Oui <input type="radio"/> Non

Précédent Suivant

Retour au profil investisseur

Formulaire de création du profil investisseur

Le questionnaire se présente donc avec plusieurs onglets, auxquels on accède en complétant et validant l'étape précédente, la plupart des champs étant requis. La majeure partie du formulaire est générée grâce à des boucles dans le template Blade pour afficher chaque question puis les occurrences associées. Les boutons radio des occurrences ont pour attribut name l'attribut renseigné dans la question, ce qui permet d'enregistrer les réponses directement dans la table `profil_investisseurs` sans avoir à faire de manipulation préalable.

La validation de chaque étape s'est faite avec l'extension jQuery Validation, qui m'a permis de mettre en place l'affichage d'un message d'erreur dynamique suivant les champs manquants, et qui permet également de valider uniquement les champs de l'étape en cours. Une fois le formulaire complété, les réponses aux questions sont directement enregistrées dans la table `profil_investisseur`.

Un formulaire identique a été ensuite mis en place pour modifier le profil.

Champs requis : `proportionPatrimoine`, `partReservee`

Projet de l'investissement : ☒ Retraite ☐ Epargne ☐ Transmission

Pour une prime périodique, quel est le montant d'investissement annuel envisagé ?
☐ De 0 à 1000 € ☐ De + 1000 € à 7000 € ☒ De + 7000 € à 20 000 € ☐ De + 20 000 €
 Durée qui vous sépare de la retraite ou durée de placement envisagée :
☐ - de 4 ans ☐ De 4 à 8 ans ☒ De 9 à 13 ans ☐ De 13 à 20 ans ☐ Plus de 20 ans

Pour une prime unique, quel est le montant d'investissement envisagé ?
☐ De 0 à 2500 € ☒ De + 2500 € à 50 000 € ☐ De + 50 000 € à 150 000 € ☐ De + 150 000 €
 Durée minimale de placement envisagée dans le cadre d'une prime unique :
☐ - de 4 ans ☐ De 4 à 8 ans ☒ De 9 à 13 ans ☐ De 13 à 20 ans ☐ Plus de 20 ans

Quelle proportion de votre patrimoine total représente ce montant ?
☐ Un investissement principal (70% et +) ☐ Une part importante (50%) ☐ Une diversification (30%) ☐ Une part peu importante (15%)
 Dans votre épargne actuelle, quelle part est réservée aux actions ?
☐ 0% ☐ De 0% à 10% ☐ De 11% à 30% ☐ + de 30%

Seconde étape du formulaire avec un message d'erreur car les réponses aux deux dernières questions manquent

3. Score et suggestions

Juste après l'enregistrement du profil, la fonction `store` du `ProfilInvestisseurController` appelle la fonction privée `score($profil)`. Cette fonction se charge de calculer le score du profil, notamment grâce aux points associés aux occurrences choisies par le client, puis elle retourne ce score. Le profil est ensuite mis à jour avec la valeur du score.

Une fois le score calculé et enregistré, la fonction `store` appelle une autre fonction privée `suggestion($profil)`. Cette fonction permet de déterminer le mode de gestion et l'approche à conseiller au client, en se basant sur le score, la sensibilité, la gestion renseignée dans le questionnaire et une grille de scoring. La fonction retourne ensuite les suggestions et le profil est mis à jour pour les champs `suggestionGestion` et `suggestionApproche`.

L'intérêt de séparer le processus en plusieurs fonctions est que cela me permet d'également utiliser ces fonctions lors de l'édition du profil, le score et les suggestions étant susceptibles de changer.

4. Accueil du profil investisseur

Après avoir mis en place l'enregistrement et l'évaluation du profil, j'ai ajouté une page d'accueil à cette fonctionnalité, accessible depuis une fiche client.

Cette page présente un simple bouton "Enregistrer un profil" si jamais aucun profil n'a été enregistré pour le client en question. Si un profil a été renseigné, cette page présente alors beaucoup plus d'information. Tout d'abord, le score ainsi que la sensibilité et le mode de gestion du profil sont indiqués. La grille de scoring est affichée juste en dessous, mettant en valeur la position dans laquelle se situe le profil.

La vue contient également deux petits formulaires. Le premier en lecture seule montrant les suggestions de méthode de gestion et d'approche déterminées grâce à la grille de scoring. Et le second, identique au premier, dans lequel on peut renseigner les choix du client. La validation de ce second formulaire ira mettre à jour les champs choixGestion et choixApproche du profil. Deux boutons sont également présents pour modifier ou supprimer le profil.

Profil investisseur de **Nicolas Carbonnier**

Score : 130 - Sensibilité : A - Gestion : **Gestion Libre**

Nombre de points		Moins de 140 points			Entre 140 et 169 points			Entre 170 et 210 points			Plus de 210 points		
Sensibilité		A	B	C	A	B	C	A	B	C	A	B	C
Gestion Profilée	Gestion	Gestion Libre	Gestion Libre	Gestion Profilée	Gestion Libre	Gestion Profilée	Gestion Profilée	Gestion Libre	Gestion Profilée	Gestion Profilée	Gestion Profilée	Gestion Profilée	Gestion Profilée
	Approche	1	2	3	2	3	4	4	5	6	5	6	7
Gestion Pilotée	Gestion	Gestion Libre	Gestion Libre	Gestion Pilotée	Gestion Libre	Gestion Pilotée	Gestion Pilotée	Gestion Libre	Gestion Pilotée	Gestion Pilotée	Gestion Pilotée	Gestion Pilotée	Gestion Pilotée
	Approche	1	2		2			4					
Gestion Libre	Gestion	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre
	Approche	1	2	3	2	3	4	4	5	6	5	6	7

1 Sans Risques 2 à 3 Prudent 3 à 4 Équilibré 4 à 6 Dynamique 6 à 7 Offensive

Suggestion

Gestion

☒ Gestion Libre ☐ Gestion Profilée ☐ Gestion Pilotée

Approche

☒ Approche Sans Risque ☐ Approche Prudente ☐ Approche Équilibrée

☐ Approche Dynamique ☐ Approche Offensive

Choix du client

Gestion

☐ Gestion Libre ☐ Gestion Profilée ☒ Gestion Pilotée

Approche

☐ Approche Sans Risque ☐ Approche Prudente ☐ Approche Équilibrée

☒ Approche Dynamique ☐ Approche Offensive

Valider les choix

Modifier le profil Éditer le document PDF Supprimer le profil

[Retour à la fiche Client](#)

Page d'accueil du profil investisseur, avec la grille de scoring et le choix de gestion du client

5. Génération du document PDF

Il était important pour cette fonctionnalité que l'on puisse éditer un document PDF similaire à la version papier originale. Pour cela j'ai repris la vue intermédiaire que j'avais mis en place pour le devoir de conseil ([cf. D.](#)) afin de définir une date de signature.

J'ai ensuite mis en place la vue utilisée pour générer le PDF. Elle reprend la mise en page des autres documents que le logiciel peut nous fournir. Elle contient les mêmes éléments que le document papier, avec toutes les occurrences de chaque question, celles choisies par le client étant cochées.

M. Gestion Électronique des Documents (GED)

Par la suite, mon maître de stage m'a demandé de mettre en place la GED sur un stockage Amazon S3 pour les documents des clients et des conseillers.

1. Documents Clients

Dans la version initial de Cercle SPA, la GED pour les documents clients était déjà en place, mais cette dernière stockait tous les documents à la racine du dossier public du serveur, sans distinguer le dossier et l'onglet renseignés lors de l'ajout du document sur le logiciel. On m'a donc confié l'amélioration de cette fonctionnalité pour, notamment, qu'elle utilise le stockage Amazon S3 et enregistre les fichiers avec une arborescence claire.

a. Laravel MediaLibrary

Pour simplifier la gestion des documents sur Laravel, mon maître de stage avait installé l'extension Laravel MediaLibrary, mais ne l'avait pas encore utilisée. J'ai donc commencé par me documenter sur l'extension.

Pour indiquer à l'extension qu'un modèle peut posséder des "Media", il faut implémenter l'interface HasMedia, c'est donc ce que j'ai fait pour le modèle ClientDocument. Ainsi, je peut appeler les fonctions de l'extension sur des entrées de la table `client_documents`. Cette implémentation permet également de supprimer les fichiers de la table `media` et du stockage si on supprime l'entrée associée dans la table du modèle.

J'ai ensuite modifié la fonction `store` du `ClientDocumentsController` pour qu'elle enregistre l'entrée dans la base puis utilise la fonction `addMediaFromRequest` pour stocker le fichier sur le serveur Amazon. J'ai dû alors modifier la vue `show` pour qu'elle récupère le lien du fichier via le Media. Les fichiers étant par défaut inaccessibles au public sur un serveur Amazon S3, j'ai configuré ce dernier pour autoriser l'accès en lecture via les liens.

Par défaut, l'extension stocke les fichiers dans un dossier correspondant à leur id dans la table media. Pour remplacer le fonctionnement par défaut, j'ai créé une classe CustomPathGenerator implémentant l'interface PathGenerator de l'extension, puis j'ai configuré l'extension pour utiliser cette nouvelle classe. J'ai ensuite modifié la fonction getPath, retournant à l'origine le chemin par défaut.

Lorsque le media passé en paramètre a pour modèle ClientDocument, je génère un chemin avec la structure suivante :

```
conseillers/NOM_Prénom/clients/id_NOM_Prénom/Dossier/(Onglet/)titre_id
```

Ainsi les fichiers sont stockés par conseiller, client, dossier et facultativement par onglet. Le fonctionnement de l'extension impose que le dernier dossier soit propre au fichier, ce qui explique le fonctionnement par défaut. Pour faciliter la distinction des documents, j'ai changé le nom de ce dossier pour qu'il corresponde au titre donné lors de l'ajout et à l'id du media.

Afin de m'assurer que les documents soient enregistrés dans le bon dossier lors d'une modification, j'ai modifié la fonction update pour qu'elle duplique le fichier dans le bon dossier et supprime l'ancien enregistrement.

b. Amélioration de l'interface

Afin d'améliorer l'interface, j'ai dans un premier temps modifié l'accueil pour que la liste des dossiers soit plus compact et n'affiche que les dossiers contenant un document. J'ai également ajouté l'affichage des documents sur cette vue, pour ne pas avoir obligatoirement besoin d'ouvrir un dossier. J'ai ensuite modifié la vue des dossiers pour que l'on puisse sélectionner un onglet parmi ceux ayant des documents dans ce dossier.

Par la suite, j'ai modifié le bouton d'ajout de document pour que cela sélectionne automatiquement un dossier (et un onglet) si le conseiller clique sur le bouton depuis un dossier (ou un onglet). Pour cela, je passe l'id du dossier et de l'onglet au formulaire d'ajout via la variable GET de l'URL.

J'ai ensuite ajouté un bouton "Afficher" pour chaque document dans la liste, afin que cela ouvre le document dans un nouvel onglet, puis j'ai modifié le lien pour qu'il permette de télécharger le fichier. Pour lancer le téléchargement des fichiers, j'ai créé un contrôleur MediaController avec une fonction download(\$id), l'id étant l'id du fichier dans la table media. La fonction retourne simplement le Media portant cet id, car l'extension permet de télécharger un fichier en le retournant de cette manière.

Plus tard, on m'a demandé d'ajouter la possibilité de choisir une compagnie associée au document, afin de retrouver plus facilement les documents d'un client pour une compagnie donnée. J'ai donc commencé par ajouter une colonne `compagnie_id` dans la table `client_documents`, puis j'ai modifié les formulaires d'ajout et d'édition pour afficher un menu déroulant pour ce nouvel attribut. J'ai ensuite modifié la vue show pour afficher, à l'image des onglets, la liste des compagnies présentes dans les documents du dossier sélectionné.

Documents électroniques de **Nicolas Carbonnier**

[Ajouter des documents](#)

[Adhésions](#) [Documents](#)

Documents électroniques

Afficher 100 éléments Copier CSV Excel PDF Imprimer Visibilité de la colonne Rechercher :

<input type="checkbox"/>	Client	Titre	Description	Conseiller	Dossier	Onglet	Compagnie	Fichier	
<input type="checkbox"/>	Nicolas Carbonnier	Démonstration		Alain GAZONI	Documents	Prévoyance	SWISSLIFE	devoir_de_conseil.pdf	Afficher Modifier Supprimer
<input type="checkbox"/>	Nicolas Carbonnier	Test Compagnie		Alain GAZONI	Adhésions	Prêts	ALBINGIA	devoir_de_conseil2.pdf	Afficher Modifier Supprimer
<input type="checkbox"/>	Nicolas Carbonnier	Démonstration Envoi Multiple		Alain GAZONI	Adhésions			devoir-de-conseil.pdf	Afficher Modifier Supprimer
<input type="checkbox"/>	Nicolas Carbonnier	Envoi multiple		Alain GAZONI	Documents	Prévoyance		Financier-1.pdf	Afficher Modifier Supprimer
<input type="checkbox"/>	Nicolas Carbonnier	Envoi multiple		Alain GAZONI	Documents			financier-3.pdf	Afficher Modifier Supprimer

Affichage de l'élément 1 à 5 sur 5 éléments [Supprimer les éléments sélectionnés](#) Précédent 1 Suivant

Page d'accueil de la GED

Adhésions de **Nicolas Carbonnier**

[Ajouter des documents](#)

[ALBINGIA](#) [Prêts](#)

Adhésions

Afficher 100 éléments Copier CSV Excel PDF Imprimer Visibilité de la colonne Rechercher :

<input type="checkbox"/>	Client	Titre	Description	Conseiller	Dossier	Onglet	Compagnie	Fichier	
<input type="checkbox"/>	Nicolas Carbonnier	Test Compagnie		Alain GAZONI	Adhésions	Prêts	ALBINGIA	devoir_de_conseil2.pdf	Afficher Modifier Supprimer
<input type="checkbox"/>	Nicolas Carbonnier	Démonstration Envoi Multiple		Alain GAZONI	Adhésions			devoir-de-conseil.pdf	Afficher Modifier Supprimer

Affichage de l'élément 1 à 2 sur 2 éléments [Supprimer les éléments sélectionnés](#) Précédent 1 Suivant

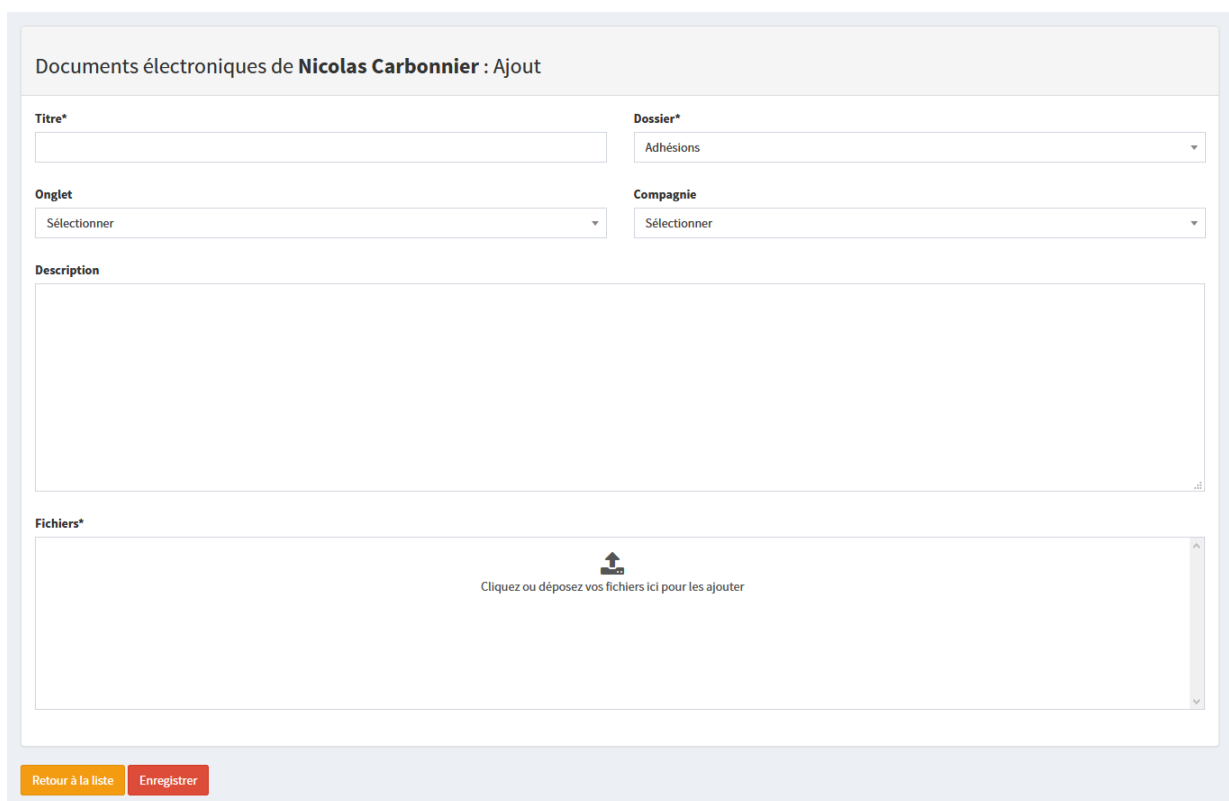
Page du dossier Adhésions

c. Glisser-déposer et envoi de fichiers multiples

Pour améliorer un peu plus l'outil, on m'a demandé de mettre en place l'envoi de fichiers multiples ainsi que la possibilité d'ajouter des fichiers avec un glisser-déposer.

L'extension DropzoneJS, conseillée par mon maître de stage, permet de faire les deux. Je me suis donc renseigné sur cette extension et je l'ai ajoutée au projet. J'ai ensuite modifié le formulaire d'ajout pour qu'il utilise l'extension.

Le stockage des fichiers étant géré par MediaLibrary, j'ai dû ensuite adapter le fonctionnement de DropzoneJS pour fonctionner de paire avec l'extension de Laravel. Pour cela, j'ai ajouté une fonction `storeMedia` au `MediaController`, appelée lors de l'ajout d'un fichier dans le formulaire, qui se charge de stocker le fichier sur le serveur du logiciel, dans un dossier temporaire. Une fois le fichier envoyé, l'extension ajoute au formulaire un champ caché ayant pour valeur le nom du fichier, afin que l'on puisse le récupérer après la validation. J'ai donné à ce champ le nom `file[]` pour pouvoir passer plusieurs fichiers dans la requête sous forme d'un tableau.



The screenshot shows a web form titled "Documents électroniques de **Nicolas Carbonnier** : Ajout". The form is organized into several sections:

- Titre***: A text input field.
- Dossier***: A dropdown menu with "Adhésions" selected.
- Onglet**: A dropdown menu with "Sélectionner" selected.
- Compagnie**: A dropdown menu with "Sélectionner" selected.
- Description**: A large text area for entering details.
- Fichiers***: A file upload area with a central icon of a folder and an upward arrow. Below the icon, it says "Cliquez ou déposez vos fichiers ici pour les ajouter".

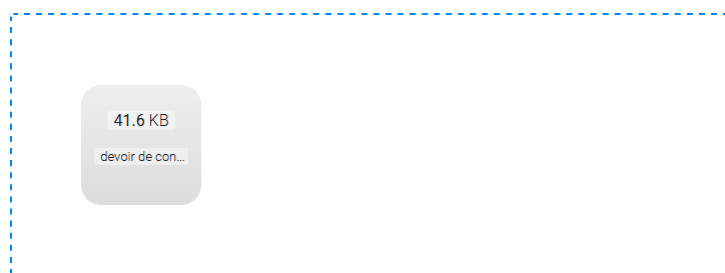
At the bottom of the form, there are two buttons: "Retour à la liste" (orange) and "Enregistrer" (red).

Formulaire d'ajout de documents électroniques

Afin de gérer ce tableau de fichiers, j'ai ensuite modifié la fonction `store` du `ClientDocumentController` pour qu'elle récupère le tableau `file` et boucle sur chaque fichier. La boucle crée une nouvelle entrée dans la table `client_documents` pour chaque

fichier, avec pour seule différence le fichier en question. Le titre, la description, le dossier et l'onglet reste les mêmes. Les fichiers sont ensuite stockés sur le serveur Amazon S3 en étant récupérés depuis le dossier temporaire.

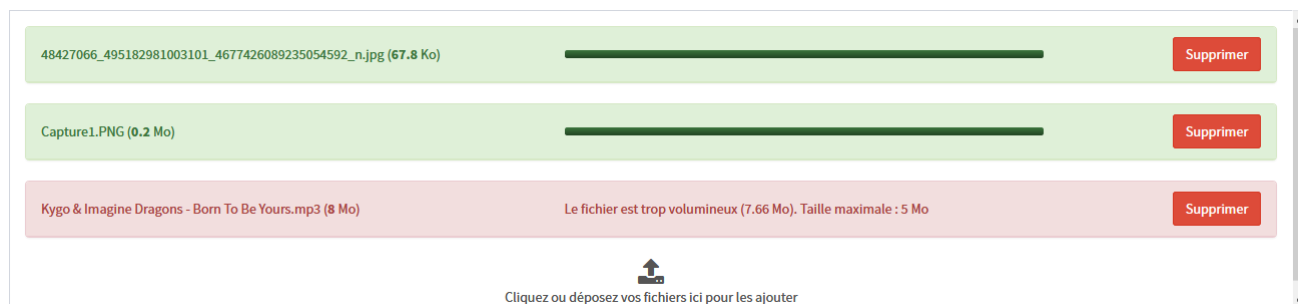
La suppression du fichier dans le dossier temporaire est automatique lors de la validation du formulaire. Elle n'était en revanche pas lors de la suppression d'un fichier de la zone de dépôt. J'ai donc ajouté une fonction `removeMedia` au `MediaController`, appelée via une requête AJAX lors de la suppression d'un fichier et qui se charge de supprimer le fichier du dossier temporaire. Et pour supprimer les fichiers pouvant encore rester, lors que l'abandon d'un formulaire par exemple, j'ai mis en place une tâche automatique CRON*. Pour cela, j'ai créé un fichier `cron.php` à la racine du projet, qui sera appelé par le serveur de production après configuration, et qui appelle le Task Scheduler de Laravel. J'ai ensuite modifié la fonction `schedule` de la classe `Kernel`, appelée par le Task Scheduler, pour qu'elle récupère les fichiers présents dans le dossier temporaire et les supprime.



Zone de dépôt originale de DropzoneJS

La mise en page de la zone de dépôt manquant de cohérence avec le reste de la page, je l'ai par la suite modifiée pour qu'elle soit plus claire. Chaque fichier est affiché sous forme d'un panel sur une ligne du système de grille de Bootstrap. Le panel est par

défaut de type info, puis passe au type success si le fichier est bien envoyé ou au type danger si il y a une erreur, comme un fichier trop volumineux ou de format incompatible par exemple. Le message d'erreur, à l'origine présenté sous forme d'une info-bulle, est maintenant affiché à la place de la barre de progression lorsqu'il est présent. Pour finir, je me suis chargé d'ajouter des messages d'erreurs personnalisés afin qu'ils soient en français.



Nouvelle zone de dépôt utilisant les classes de Bootstrap

2. Documents Conseillers (ACPR)

Pour exercer leur métier, les conseillers doivent fournir à l'ACPR (Autorité de Contrôle Prudentiel et de Résolution) un certain nombre de documents. Jusque là, chaque conseiller stockait ces documents sur son propre matériel. Afin de centraliser le stockage et de retrouver plus facilement ces documents, on m'a demandé de mettre en place la GED pour ces derniers, en m'inspirant du travail réalisé pour les documents clients.

a. Nouvelles tables dans la base

Avant de reprendre le code de la gestion des documents clients, il m'a fallu mettre en place plusieurs tables dans la base de données, ainsi que leur modèle, à commencer par la table `conseiller_documents`. Elle possède les colonnes `title`, `description`, `file`, `conseiller_id`, `dossier_id` et `onglet_id`. Le modèle implémente l'interface `HasMedia` pour pouvoir être utilisé avec `MediaLibrary`.

J'ai ensuite créé les tables `dossiers_acpr` et `onglets_acpr`. Les deux possèdent les colonnes `ordre` et `name`, l'ordre étant défini par une lettre pour les dossiers et un nombre pour les onglets. Ce dernier était nécessaire pour l'arborescence des fichiers, car les dossiers doivent être rangés dans un ordre précis. La table `onglets_acpr` possède également une colonne `dossier_id` car ici les onglets sont propres à chaque dossier, il était donc important de pouvoir les distinguer.

b. Mise en place de la fonctionnalité

Après avoir mis en place toutes les tables nécessaires à cette fonctionnalité, j'ai mis en place cette dernière, en reprenant le travail fait pour les documents clients ([cf. M. 1.](#)) et en l'adaptant aux nouvelles tables. Je ne développerai donc ici que les modifications et ajouts importants pour cette nouvelle fonctionnalité, les autres modifications étant simplement des changements de nom pour les variables ou l'adaptation des requêtes aux nouvelles tables.

Les onglets étant ici propres à un dossier, j'ai dû mettre en place le remplissage dynamique du select de l'onglet. J'ai donc ajouté une fonction `getOngletsACPR($dossier_id)` au `DropDownListController`, appelée via une requête AJAX lorsque l'utilisateur sélectionne un dossier dans le formulaire. Pour que l'onglet se sélectionne automatiquement si on ajoute un fichier depuis un onglet, j'ai ajouté une ligne au JavaScript pour aller sélectionner l'option adéquate après le remplissage du select.

Pour définir l'arborescence des fichiers pour ces documents, j'ai modifié le CustomPathGenerator pour que le chemin retourné par la fonction getPath soit de la structure suivante :

conseillers/*NOM_Prénom*/documents/*ordre_Dossier/ordre_Onglet/titre_id* lorsque le media passé en paramètre a pour modèle ConseillerDocument.

c. Gestion des réclamations

Plus tard, mon maître de stage m'a demandé d'ajouter une fonctionnalité accessible depuis la GED conseiller, permettant de visualiser toutes les anomalies de type "Réclamation" du portefeuille pour pouvoir les exporter facilement en document Excel.

Le document de sortie nécessitant des informations absentes à l'origine, j'ai dans un premier temps ajouté les colonnes nature, gravite, accuse, date_limite et procedure à la table anomalie_par_produits. J'ai ensuite modifié les formulaires d'ajout et d'édition pour afficher les champs associés à ses nouvelles colonnes uniquement lorsque l'utilisateur sélectionne le type "Réclamation".

Pour cette nouvelle fonctionnalité, j'ai ajouté une fonction reclamations au AnomalieParProduitsController. Cette dernière se charge de récupérer toutes les réclamations du portefeuille à partir du début de l'année précédente. Ainsi, les conseillers peuvent voir toutes les réclamations pour l'année n-1 et l'année en cours, sachant que l'exportation sera notamment utilisée pour les données de l'année n-1. L'exportation est directement gérée par l'extension DataTables, utilisée avec le tableau affiché.

Réclamations pour Alain GAZONI du 01/01/2019 au 27/07/2020												
Afficher	100	éléments	Copier	CSV	Excel	PDF	Imprimer	Visibilité de la colonne	Rechercher : <input type="text"/>			
Références réclamation	Compagnie	Références contrat	Nom	Prénom	Nature de la réclamation	Contenu	Niveau de gravité	Date de réception	Accusé de réception à effectuer sous 10 jours	Date limite de traitement	Procédure escalade éventuelle	Date de clôture
Anomalie N°18	ALLIANZ	Habitation N°	Dupont	Benjamin		Grosse Anomalie		04/05/2020				
Anomalie N°35	ACORIS	Santé N°	Dupont	Christophe				11/06/2020				
Anomalie N°42	AVIVA	AVIVA Retraite Plurielle Madelin N°	Carbonnier	Nicolas	Normale modifiée	Drôle de réclamation	42	01/07/2020	15/07/2020	18/07/2020	Procédure habituelle	
Affichage de l'élément 1 à 3 sur 3 éléments										Précédent	1	Suivant

Liste des réclamations sur les contrats de Alain GAZONI du début de l'année précédente à aujourd'hui

N. Signature électronique

Après avoir mis en place la GED, mon maître de stage m'a demandé de mettre en place la signature électronique pour les documents administratifs, le devoir de conseil et le profil investisseur, afin de dématérialiser cette procédure et ainsi la simplifier.

Mon maître de stage m'a proposé d'utiliser DocuSign, un service de signature électronique permettant notamment d'envoyer des documents à signer directement depuis leur site. Ce service dispose également d'une API REST*, pour permettre de l'intégrer à un autre logiciel web. C'est donc cette API que nous avons utilisé.

1. Modifications pour accueillir DocuSign

Avant de mettre en place cette nouvelle fonctionnalité, j'ai dû adapter ce qui était déjà mis en place. Pour cela, j'ai ajouté des seconds formulaires dans les pages d'édition des documents PDF, avec des champs cachés pour le type de document envoyé et la date de signature. Cette dernière est mise à jour via un script lorsque l'on modifie la date de signature dans le formulaire original. Cela évite à l'utilisateur de la saisir une seconde fois et allège le formulaire. On m'a alors demandé d'ajouter un champ prérempli avec l'adresse mail du client, afin de vérifier son adresse et de la modifier si nécessaire avant l'envoi.

Si on prend l'exemple du devoir de conseil, j'ai ensuite modifié le CustomerController pour séparer la génération du PDF en deux fonctions. La fonction printPDF a été renommée generatePDF et retourne simplement le PDF et la nouvelle fonction printPDF appelle generatePDF et lance le téléchargement du PDF retourné. Ainsi, la route reste inchangée pour le téléchargement du devoir de conseil et on peut appeler generatePDF pour obtenir le PDF à envoyer via DocuSign. J'ai appliqué ces modifications avec le même principe pour les autres documents.

Plus tard, j'ai modifié les vues des PDF pour ajouter les textes "sign here client" et "sign here conseiller" sous les noms du client et du conseiller. Ces textes vont permettre à l'API de définir facilement les zones de signatures pour chacun des tiers. Afin de ne pas être visibles par l'utilisateur, j'ai mis ces textes en blanc.

2. Développement sur la Sandbox

Après avoir préparé le projet pour accueillir l'API, j'ai débuté le développement. Pour cela, DocuSign propose un environnement sandbox (de démonstration), ainsi que plusieurs guides pour utiliser l'API. J'ai donc commencé par suivre les guides pour mettre en place la fonctionnalité et on a créé un compte sur le service pour Stratégies & Perspectives d'Avenir.

Pour simplifier l'utilisation de l'API sur Laravel, j'ai installé les wrappers Laravel DocuSign et DocuSign Rest Client en plus du client PHP officiel de DocuSign.

Afin de centraliser l'envoi des DocuSign, j'ai mis en place le contrôleur `DocuSignController` possédant une fonction `signPDF`, appelée par les nouveaux formulaires. Cette fonction se charge de récupérer les informations du client et du conseiller, puis récupère le type de document passé par la requête. Le type va permettre de définir le sujet du mail, ainsi que le nom du fichier, puis d'appeler la fonction associée pour générer le PDF, qui sera enregistré temporairement sur le stockage du serveur. La date de signature et le mail du client sont également récupérés à ce moment là, car les champs associés n'ont pas le même nom dans les formulaires.

Une fois que la fonction a récupéré toutes les informations nécessaires suivant le type de document, elle récupère le document précédemment généré, définit les zones de signature pour le client et le conseiller ainsi que ces deux signataires en question. La fonction crée ensuite une enveloppe, qui correspond au mail, avec un sujet, une langue, un document et des signataires. Pour finir, la fonction se charge d'envoyer l'enveloppe via l'API de DocuSign, puis supprime le document stocké temporairement, étant maintenant envoyé.

Une fois cela mis en place, c'est l'API qui prend le relai et qui se charge d'envoyer un mail au premier signataire (ici le client), puis au second après que le premier ait signé. Pour finir, les deux tiers reçoivent le document final signé par tous.

3. Historique DocuSign

Avant de passer cette fonctionnalité de la sandbox de DocuSign à la version de production, mon maître de stage m'a demandé si il était possible de connaître le nombre de DocuSign envoyés par chacun des conseillers. Le service étant payant, il souhaite pouvoir facturer chaque cabinet suivant son utilisation de la fonctionnalité. Je lui ai proposé de mettre en place un historique simple comportant uniquement les noms du conseiller et du client, ainsi que la date d'envoi et le type de document envoyé.

Pour enregistrer l'historique, j'ai créé une table `docu_signs` avec les colonnes `conseiller_id`, `client_id`, `envoi` et `type`. Après chaque envoi, la fonction `signPDF` ajoute une nouvelle entrée à cette table avec les informations du document envoyé. Pour afficher cet historique, j'ai ajouté une fonction `historique` au `DocuSignController` qui se charge de récupérer l'intégralité de l'historique, trié par conseiller et par date, avec les jonctions sur `conseillers` et `clients` pour avoir leurs noms et prénoms. Les données sont affichées dans un tableau utilisant `DataTables` pour être facilement exportable en Excel.

DocuSign : Historique			
Afficher	100	éléments	Copier CSV Excel PDF Imprimer Visibilité de la colonne
Rechercher :			
Conseiller	Client	Date d'envoi	Type de document
GAZONI Alain	Carbonnier Manon	01/07/2020	Profil investisseur
GAZONI Alain	Carbonnier Nicolas	16/07/2020	Devoir de conseil IARD
GAZONI Alain	Carbonnier Nicolas	17/07/2020	Devoir de conseil
GAZONI Alain	Carbonnier Nicolas	17/07/2020	Devoir de conseil
GAZONI Alain	Carbonnier Nicolas	20/07/2020	Informations précontractuelles
GAZONI Alain	Carbonnier Nicolas	20/07/2020	Mandat administratif
GAZONI Alain	Carbonnier Nicolas	20/07/2020	Profil investisseur
GAZONI Alain	Carbonnier Nicolas	20/07/2020	Envoi groupé
GAZONI Alain	Carbonnier Nicolas	21/07/2020	Envoi groupé IARD
GAZONI Alain	Carbonnier Nicolas	22/07/2020	Envoi groupé IARD
Affichage de l'élément 1 à 10 sur 10 éléments			
Précédent 1 Suivant			

Historique de l'envoi de document via DocuSign

4. Application API et connexion à DocuSign

Pour pouvoir utiliser réellement le service, il y a fallu définir une application API dans l'interface développeur de DocuSign. Le logiciel doit se connecter à cette dernière pour authentifier l'utilisateur, obtenir l'identifiant unique de son compte ainsi que le token. Sans ces données, l'envoi de documents via DocuSign est impossible. J'ai donc commencé par configurer le logiciel pour se connecter à l'application, en reprenant en partie le code fourni par DocuSign dans ses exemples, et notamment les classes d'authentification DocuSign et DocuSignResourceOwner.

Le reste du code a été fait dans le DocuSignController, notamment avec les fonctions getAuthProvider, retournant un objet de la classe d'authentification DocuSign, login qui envoie vers la page de connexion de DocuSign et callback qui est appelée par l'API après authentification et qui va stocker en variable de session les informations essentielles telles que le token, l'identifiant de l'utilisateur ainsi que son nom et son adresse mail.

Les utilisateurs devant se connecter à leur propre compte, j'ai mis en place un système de connexion à DocuSign dans les vues ayant des formulaires d'envoi de documents. Lorsqu'un utilisateur est sur une de ces vues, un bouton lui permet de se connecter au service. Après authentification, la page est actualisée et les formulaires d'envoi DocuSign apparaissent. À la place du bouton de connexion s'affiche alors le nom et l'adresse mail du compte DocuSign connecté, ainsi qu'un bouton de déconnexion.

5. Envoi groupé

Les informations précontractuelles, le devoir de conseil et optionnellement le profil investisseur étant envoyés ensemble par courrier papier, j'ai par la suite mis en place l'envoi groupé de ces trois documents pour éviter d'envoyer plusieurs mails aux clients et pour réduire la consommation d'enveloppes DocuSign, ces dernières ayant un coût.

Pour cela j'ai ajouté deux nouveaux formulaires dans l'édition du devoir de conseil, le premier pour le devoir normal et le second pour le devoir IARD. Ils renvoient tous les deux vers la fonction signPDF du DocuSignController, que j'ai adapté pour ces deux types d'envoi. Ainsi la fonction va appeler les fonctions de génération des deux ou trois documents demandés et tous les ajouter à l'enveloppe, là où un seul document est ajouté pour les autres types d'envoi.

The screenshot shows a web interface titled "Devoir de conseil pour 4 PIERRES". At the top right, it indicates the user is connected to a DocuSign account for "Nicolas Carbonnier" with a "Déconnexion" (Logout) button. The main content area is divided into two sections: a light blue section for "Vous allez éditer un devoir de conseil" and a light yellow section for "Vous allez éditer un devoir de conseil IARD". Each section contains an "Envoi groupé" (Grouped sending) form. The blue section form includes fields for "À quelle date voulez-vous l'éditer ?" (When do you want to edit it?), "Date de signature pour les info. précontractuelles" (Signature date for pre-contractual info), "Date de signature pour le devoir" (Signature date for the duty), "Adresse mail du client" (Client email address), and a checkbox for "Inclure le profil investisseur" (Include investor profile). Buttons for "Générer" (Generate), "Envoyer en DocuSign" (Send via DocuSign), and "Envoyer en DocuSign groupé" (Send via grouped DocuSign) are present. The yellow section form has a similar layout but without the investor profile checkbox. A "Retour à la liste des solutions d'assurance" (Return to the list of insurance solutions) button is at the bottom left.

Page d'édition du devoir de conseil avec la zone de connexion à DocuSign et les nouveaux formulaires

6. Passage à la version de production

Après avoir implémenté les dernières fonctionnalités et validé la clé d'intégration, qui nécessitait vingt appels de l'API valides et consécutifs, on était prêt à transférer la clé de la sandbox à un compte de production. Malheureusement, nous n'avons pas eu le temps de mettre en place le contrat pour le compte de production avec le service client de DocuSign après notre demande de conseil et les négociations quant au tarif de souscription.

VI. Conclusion

Au cours de ces douze semaines de stage, j'ai été amené à réaliser et à mener à bien de nombreuses tâches différentes de par leur fonctionnement, leur utilité ou leur importance. J'ai pu améliorer l'ergonomie globale du logiciel grâce notamment à la protection des formulaires, au changement du format des dates et au support des numéros de téléphone, tout comme j'ai pu développer des fonctionnalités totalement inédites pour Cercle SPA, avec par exemple l'exportation des calendriers vers des services tiers ou la mise en place de la signature électronique.

Ce stage m'a donné l'occasion de mettre en pratique et de renforcer mes compétences en programmation Web, notamment en PHP, JavaScript et HTML. Il m'a permis de découvrir de nouveaux outils, comme le framework Laravel, que j'ai trouvé simple et agréable à utiliser, ainsi que des extensions puissantes telles que Chart.js et DropzoneJS. J'ai également pu découvrir plus en détail l'utilisation des API, une technologie de plus en plus utilisée dans le monde du Web.

Bien que le contexte de la crise sanitaire actuelle ne m'ait pas permis d'avoir une expérience en entreprise complète, j'ai pu découvrir lors de ma dernière semaine de stage une ambiance agréable et conviviale de travail, avec une équipe accueillante, encourageante et reconnaissante. J'ai ainsi pu avoir de nombreux retours positifs sur les travaux que j'ai pu mener durant ce stage. Celui-ci m'a permis de me rendre compte de l'impact que peut avoir l'ajout de nouvelles fonctionnalités au sein de l'outil de travail d'une entreprise.

Le contexte actuel m'a tout de même permis d'apprendre à travailler et à communiquer de manière différente avec les collaborateurs, notamment en utilisant des outils tels que Skype pour assurer le suivi et montrer le travail réalisé.

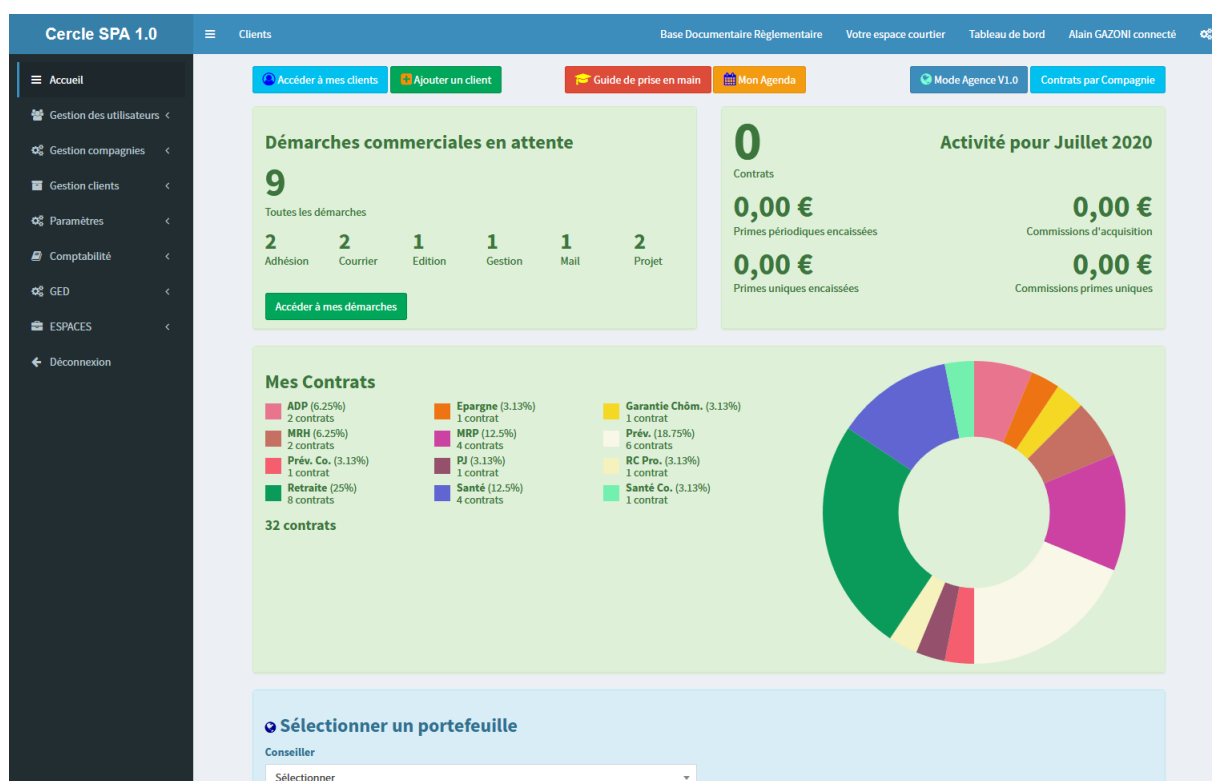
Pour finir, je peux dire que mon stage m'a pleinement satisfait, m'a conforté dans mon choix d'étude et m'a permis de cibler un peu plus le domaine sur lequel j'aimerais me concentrer, à savoir le développement Web.

Ce stage m'ouvre maintenant les portes du monde professionnel car mon aventure avec Stratégies & Perspectives d'Avenir ne s'arrête pas là. En effet, mon maître de stage et son associé Ali TAZI, étant satisfaits de mon travail, m'ont proposé de poursuivre mes travaux sur Cercle SPA en CDI à temps partiel, en parallèle de ma poursuite d'étude en Master.

VII. Après-texte

A. Annexes

1. Captures d'écrans supplémentaires



Accueil de Cercle SPA

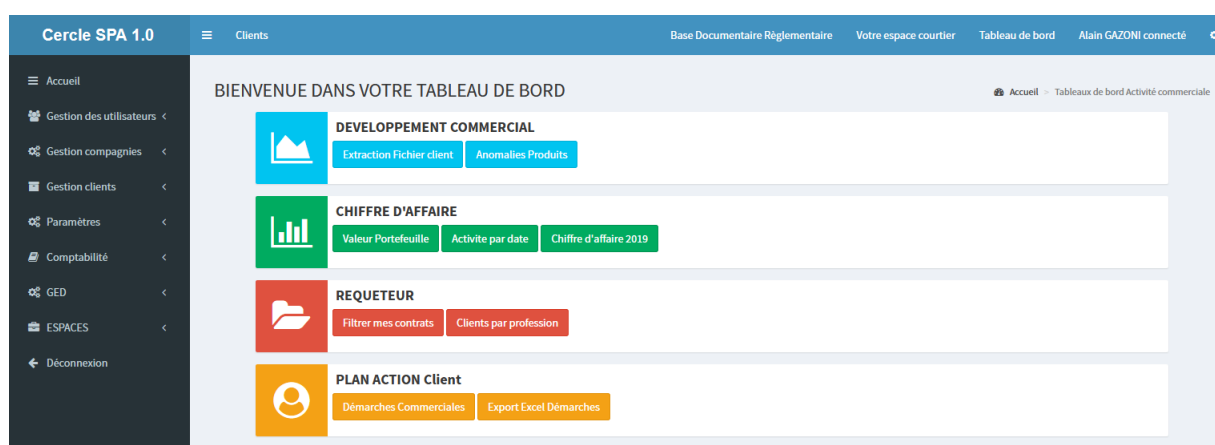


Tableau de bord

Cercle SPA 1.0 Clients Base Documentaire Règlementaire Votre espace courtier Tableau de bord Alain GAZONI connecté

Accueil
Gestion des utilisateurs <
Gestion compagnies <
Gestion clients <
Clients
Paramètres <
Comptabilité <
GED <
Dossiers
Dossiers ACPR
Onglets
Onglets ACPR
Conseillers
Clients
Documents
DocuSign
ESPACES <
Déconnexion

Liste des clients **Cabinet Est Cap**

Ajouter un client Importer CSV.

Afficher 100 éléments Copier CSV Excel PDF Imprimer Visibilité de la colonne Rechercher :

Civilité	Nom	Statut	Conseiller	Conseiller Vie	Conseiller IARD	
Monsieur	Carbonnier Nicolas	Salarié(e) Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Madame	Dupont Vanessa	Salarié(e) Non Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Mademoiselle	Dupont Esther	Salarié(e) Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Benjamin	Salarié(e) Non Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Quentin	Salarié(e) Non Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Claude	Gérant Majoritaire	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Christophe	Salarié(e) Non Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Mademoiselle	Dupont Sophie	Salarié(e) Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Maxime	Non Défini	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Madame	Dupont Karine	Non Défini	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Matthias	Salarié(e) Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Jean-Luc	Non Défini	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Saïd	Gérant Majoritaire	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Lilian	Salarié(e) Non Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Lanig	Salarié(e) Cadre	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Monsieur	Dupont Fabrice	Profession Libérale	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer
Docteur	Dupont Alain	Non Défini	Client de Est Cap	Est Cap	Prospect de Non Affecté	Afficher Supprimer

Liste des clients du cabinet Est Cap



Cercle SPA 1.0 Clients Base Documentaire Règlementaire Votre espace courtier Tableau de bord Alain GAZONI connecté



Accueil
Gestion des utilisateurs <
Gestion compagnies <
Gestion clients <
Paramètres <
Comptabilité <
GED <
ESPACES <
Déconnexion


accès GED Infos Générales **Professionnel** Historique Besoins Solutions d'assurance Profil investisseur Espace documents

Client : M. Nicolas Carbonnier

Note :

 Courtier IARD : Non Affecté
  Courtier CGPI : Non Affecté

 Célibataire
  Développeur(euse) informatique

 Salarié(e) Cadre

Revenus Relationnel

[Créer revenus client pour Nicolas Carbonnier](#)

Montant	Année	Nom client	Type revenu
Aucun enregistrement			

[Retour à la liste](#)

Fiche Client

Cercle SPA 1.0

Accueil

Gestion des utilisateurs <

Gestion compagnies <

Gestion clients <

Paramètres <

Comptabilité <

GED <

ESPACES <

Déconnexion

Clients

Base Documentaire Règlementaire

Votre espace courtier

Tableau de bord

Alain GAZONI connecté

Ajouter une anomalie sur ce produit

Anomalie par produit

Date	Cloture	Date cloture	Commentaire	Type	N° dossier	
01/07/2020	<input type="checkbox"/>			Réclamation		<a>Afficher <a>Modifier <a>Supprimer

Détails du contrat

AVIVA Retraite Plurielle Madelin (AVIVA)

Nicolas Carbonnier

ref : 123

Produit

AVIVA Retraite Plurielle Madelin

Code Courtier

Sélectionner

Client

Carbonnier

Numéro Contrat

Contrat Maître

Sélectionner

Option

Situation contrat

Fractionnement

Souscripteur

Type de prescripteur

Age Terme Retraite

Dossier concurrent

En cours (Attente Em...

Non Défini

Carbonnier

Non défini

☐

Avie ☐

Art 154 bis ☐

Art 62 ☐

PEP ☐

PERP ☐

Art 82 ☐

Art 83 ☐

Art 39 ☐

Facturation Audit ☐

Clause Type ☐

Clause Démembrée ☐

Clause Nominative ☐

Clause Acceptée ☐

Commentaire

Valider les modifications

Ajouter un événement sur ce produit

Événements liés au produit

Événement	Date effet	Formalisé	Montant PP	Montant PU	
Affaire Nouvelle	31/05/2011	<input checked="" type="checkbox"/>	10000	10000	<a>Afficher les détails <a>Supprimer
Avenant Augmentation	16/06/2020	<input type="checkbox"/>	12000 (+ 2000) (+ 20% par rapport à la PP précédente)	()	<a>Afficher les détails <a>Supprimer
Versement Unique	16/06/2020	<input type="checkbox"/>	(12000)	667	<a>Afficher les détails <a>Supprimer

Retour à la liste des solutions d'assurance

Fiche Contrat

2. Version papier du profil investisseur

FICHE DE RENSEIGNEMENTS
CONNAISSANCE ET EXPÉRIENCE EN MATIÈRE FINANCIÈRE
Directive 2004/39/CE du 30 avril 2004 sur les marchés d'instruments financiers, dite directive «MIF»
 En application des articles L 132-27-I du code des assurances et article L223-25-3 du code de la mutualité

DATE:

CLIENT:

COMPAGNIE:

NOM DOSSIER:

ÉVÉNEMENT: ☐ AFFAIRE NOUVELLE ☐ AVENANT ☐ RACHAT ☐ ARBITRAGE

CONNAISSANCES ET EXPÉRIENCES DE L'ÉPARGNANT

Etes-vous un professionnel de la finance ? ☐ OUI ☐ NON proFinance (0,1)

(nom)Conn (0,1)

Connaissez-vous ce support ?

<input type="checkbox"/> ACTION action (0,1)	<input type="checkbox"/> OUI <input type="checkbox"/> NON
<input type="checkbox"/> OBLIGATIONS obligations (0,1)	<input type="checkbox"/> OUI <input type="checkbox"/> NON
<input type="checkbox"/> OPCVM opcvm (0,1)	<input type="checkbox"/> OUI <input type="checkbox"/> NON
<input type="checkbox"/> FONDS STRUCTURES fondsStruct (0,1)	<input type="checkbox"/> OUI <input type="checkbox"/> NON

Si oui ≤ 1 ☐ 5 pts Si oui = 2 ☐ 20 pts Si 3 ≤ oui ≤ 4 ☐ 30 pts

(nom)Inv (0,1)

Avez-vous déjà investi dessus ?

<input type="checkbox"/> OUI <input type="checkbox"/> NON
<input type="checkbox"/> OUI <input type="checkbox"/> NON
<input type="checkbox"/> OUI <input type="checkbox"/> NON
<input type="checkbox"/> OUI <input type="checkbox"/> NON

Si oui = 0 ☐ 5 pts Si 1 ≤ oui ≤ 2 ☐ 10 pts Si oui ≥ 3 ☐ 15 pts

TOTAL

PROJET, MONTANT, PROPORTIONS ET DURÉE DE L'INVESTISSEMENT

Projet de l'investissement : ☐ RETRAITE ☐ ÉPARGNE ☐ TRANSMISSION

projetRetraite (0,1) projetEpargne(0,1) projetTrans (0,1)

Pour une prime périodique, quel est le montant d'investissement annuel envisagé ? invPP (4 occurrences)

<input type="checkbox"/> DE 0 À 1000€ <input type="checkbox"/> 5 pts	<input type="checkbox"/> DE + 1000€ À 7000€ <input type="checkbox"/> 10 pts	<input type="checkbox"/> DE + 7000€ À 20 000€ <input type="checkbox"/> 30 pts	<input type="checkbox"/> DE + 20 000€ <input type="checkbox"/> 40 pts	<input type="text" value="pts"/>
----------------------------------------------------------------------	-----------------------------------------------------------------------------	-------------------------------------------------------------------------------	-----------------------------------------------------------------------	----------------------------------

Durée qui vous sépare de la retraite ou durée de placement envisagée : dureePP (5 occurrences)

<input type="checkbox"/> - DE 4 ANS <input type="checkbox"/> 5 pts	<input type="checkbox"/> DE 4 À 8 ANS <input type="checkbox"/> 15 pts	<input type="checkbox"/> DE 9 À 13 ANS <input type="checkbox"/> 25 pts	<input type="checkbox"/> DE 13 À 20 ANS <input type="checkbox"/> 35 pts	<input type="checkbox"/> PLUS DE 20 ANS <input type="checkbox"/> 50 pts	<input type="text" value="pts"/>
--------------------------------------------------------------------	-----------------------------------------------------------------------	------------------------------------------------------------------------	-------------------------------------------------------------------------	-------------------------------------------------------------------------	----------------------------------

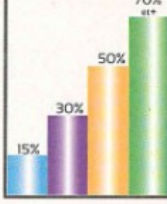
Pour une prime unique, quel est le montant d'investissement envisagé ? invPU (4 occurrences)

<input type="checkbox"/> DE 0 À 2500€ <input type="checkbox"/> 5 pts	<input type="checkbox"/> DE + 2500€ À 50 000€ <input type="checkbox"/> 10 pts	<input type="checkbox"/> DE + 50 000€ À 150 000€ <input type="checkbox"/> 30 pts	<input type="checkbox"/> DE + 150 000€ <input type="checkbox"/> 40 pts	<input type="text" value="pts"/>
----------------------------------------------------------------------	-------------------------------------------------------------------------------	----------------------------------------------------------------------------------	------------------------------------------------------------------------	----------------------------------

Durée minimale de Placement envisagée dans le cadre d'une prime unique : dureePU (5 occurrences)

<input type="checkbox"/> - DE 4 ANS <input type="checkbox"/> 5 pts	<input type="checkbox"/> DE 4 À 8 ANS <input type="checkbox"/> 15 pts	<input type="checkbox"/> DE 9 À 13 ANS <input type="checkbox"/> 25 pts	<input type="checkbox"/> DE 13 À 20 ANS <input type="checkbox"/> 35 pts	<input type="checkbox"/> PLUS DE 20 ANS <input type="checkbox"/> 50 pts	<input type="text" value="pts"/>
--------------------------------------------------------------------	-----------------------------------------------------------------------	------------------------------------------------------------------------	-------------------------------------------------------------------------	-------------------------------------------------------------------------	----------------------------------

Quelle proportion de votre patrimoine total représente ce montant ? proportionPatrimoine (4 occurrences)



<input type="checkbox"/> UN INVESTISSEMENT PRINCIPAL <input type="checkbox"/> 10 pts	<input type="text" value="pts"/>
<input type="checkbox"/> UNE PART IMPORTANTE <input type="checkbox"/> 20 pts	<input type="text" value="pts"/>
<input type="checkbox"/> UNE DIVERSIFICATION <input type="checkbox"/> 30 pts	<input type="text" value="pts"/>
<input type="checkbox"/> UNE PART PEU IMPORTANTE <input type="checkbox"/> 40 pts	<input type="text" value="pts"/>

partReservee (4 occurrences)

Dans votre épargne actuelle, quelle part est réservée aux actions ? ☐ 0% ☐ 40 pts ☐ DE 0% À 10% ☐ 30 pts ☐ DE 11% À 30% ☐ 20 pts ☐ + DE 30% ☐ 10 pts

Nous attirons votre attention sur les conséquences du non respect de la durée effective de votre placement, ce qui nous amènerait à modifier nos préconisations

SENSIBILITE AU RISQUE

Que faites vous si votre Epargne baisse ? **baisseEpargne (3 occurrences)**

☐ J'EN PROFITE POUR VERSER VERS LE FONDS EUROS OU DÉFENSIF 10 pts ☐ JE NE FAIS RIEN ET MAINTIENS MON OBJECTIF 20 pts ☐ J'EN PROFITE POUR VERSER SUR L'ÉPARGNE EN UC 30 pts

Quelle amplitude de variation vous semble acceptable sans faire d'arbitrage ? **amplitudeVar (5 occurrences)**

☐ 5% 5 pts ☐ 10% 15 pts ☐ 20% 25 pts ☐ 30% 35 pts ☐ 40% 45 pts

Pour être plus précis, quelle variation ou volatilité êtes-vous prêt à accepter ? **volatiliteAccept (6 occurrences)**
(La volatilité permet de mesurer le degré de dépendance aux fluctuations du marché)

☐ PAS RISQUÉ/ VOLATILITÉ DE 0.5% À 2% 10 pts ☐ PEU RISQUÉ/ VOLATILITÉ DE 2% À 5% 20 pts ☐ MOYENNEMENT RISQUÉ/ VOLATILITÉ DE 5% À 10% 30 pts ☐ ASSEZ RISQUÉ/ VOLATILITÉ DE 10% À 15% 40 pts ☐ FORTEMENT RISQUÉ/ VOLATILITÉ DE 15% À 25% 50 pts ☐ TRÈS FORTEMENT RISQUÉ/ VOLATILITÉ + DE 25% 60 pts

NBR DE POINTS TOTAL

Quelle attitude souhaitez vous adopter sur votre épargne ? **attitude (3 occurrences)**

☐ JE CHERCHE UNE PERFORMANCE SUPÉRIEURE ☐ JE CHERCHE UNE PERFORMANCE MODÉRÉE ☐ JE CHERCHE UNE PROGRESSION CONTINUE ET RÉGULIÈRE

REPORT DE LETTRE

PRINCIPE DE GESTION

Quelle Gestion souhaitez-vous idéalement adopter pour vos investissements ?

☐ EFFECTUER LIBREMENT LA GESTION DE CE DOSSIER (GESTION LIBRE) ☐ DÉLÉGUER LA GESTION EN FONCTION DE VOTRE PROFIL DE RISQUE ET VOTRE SENSIBILITÉ AU RISQUE (GESTION PROFILÉE OU SOUS-MANDAT) ☐ SOUS TRAITER POUR UNE GESTION DÉGRESSIVE EN FONCTION DE L'HORIZON DE PLACEMENT (GESTION PILOTÉE)

CHOIX DE REPARTITION DE VOTRE INVESTISSEMENT

Sur la base des informations que vous nous avez indiquées, nous vous suggérons de répartir votre investissement de la façon suivante :

SUGGESTION

☐ LIBRE ☐ PROFILÉE ☐ PILOTÉE **suggestionGestion (3 occurrences)**

☐ Approche sans Risque ☐ Approche Prudente ☐ Approche Équilibrée ☐ Approche Dynamique ☐ Approche Offensive **gestionApproche (5 occurrences)**

CHOIX DU CLIENT

☐ LIBRE ☐ PROFILÉE ☐ PILOTÉE **choixGestion (3 occurrences)**

☐ Approche sans Risque ☐ Approche Prudente ☐ Approche Équilibrée ☐ Approche Dynamique ☐ Approche Offensive **choixApproche (5 occurrences)**

Tous les produits pouvant vous être proposés sont soumis aux fluctuations des marchés et aucune garantie ne peut être donnée quant à leur évolution future. Il est rappelé que les performances passées ne préjugent pas des performances futures, ne sont pas constantes dans le temps et ne constituent en aucun cas une garantie future de performance ou de capital.

Le client reconnaît avoir pris connaissance de contenu du présent document préalablement à la signature du ou des contrats d'assurance proposé(s) ci-dessus et en accepter les termes. Il reconnaît également avoir reçu les Conditions générales de ces mêmes contrats, et après lecture attentive, en accepter les termes.

Fait le _____ en deux exemplaires dont un pour _____ qui reconnaît l'avoir reçu

Le candidat à l'assurance L'intermédiaire en assurance

Signature : Signature :

Ce document ne se substitue en aucun cas aux devoirs et obligations professionnelles de l'intermédiaire. Il ne constitue qu'une aide pédagogique permettant la mise en place d'un arbre de décision approprié, sous la forme d'un questionnaire logique et proportionné.

GRILLE DE SCORING DIDACTIQUE

Nombre de points		Moins de 140 points			Entre 140 et 169 points			Entre 170 et 210 points			plus de 210 points		
Sensibilité		A	B	C	A	B	C	A	B	C	A	B	C
Gestion Profilée	Gestion	Gestion Libre	Gestion Libre	Gestion Profilée	Gestion Libre	Gestion Profilée	Gestion Profilée	Gestion Libre	Gestion Profilée	Gestion Profilée	Gestion Profilée	Gestion Profilée	Gestion Profilée
	Approche	1	2	3	2	3	4	4	5	6	5	6	7
Gestion Pilotée	Gestion	Gestion Libre	Gestion Libre	Gestion Pilotée	Gestion Libre	Gestion Pilotée	Gestion Pilotée	Gestion Libre	Gestion Pilotée	Gestion Pilotée	Gestion Pilotée	Gestion Pilotée	Gestion Pilotée
	Approche	1	2		2			4					
Gestion Libre	Gestion	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre	Gestion Libre
	Approche	1	2	3	2	3	4	4	5	6	5	6	7

Lexique

1 Sans Risque 2 à 3 Prudent 3 à 4 Équilibré 4 à 6 Dynamique 6 à 7 Offensive

3. Exemple de fichier .ics

```
BEGIN:VCALENDAR
CALSCALE:GREGORIAN
X-WR-TIMEZONE:Europe/Paris
VERSION:2.0
NAME:Cercle SPA - Alain GAZONI
X-WR-CALNAME:Cercle SPA - Alain GAZONI
REFRESH-INTERVAL;VALUE=DURATION:PT1H
X-PUBLISHED-TTL:PT1H
METHOD:PUBLISH
PRODID:-//CercleSPA//CalendarExport//FR
BEGIN:VEVENT
DTSTART:20200729T144500
DTEND:20200729T154500
DTSTAMP:20200729T135257
CREATED:20200729T135143
SUMMARY:Adhésion : Carbonnier
DESCRIPTION:
UID:d2dde18f00665ce8623e36bd4e3c7c5@cerclespa.fr
END:VEVENT
BEGIN:VEVENT
DTSTART:20200731T140000
DTEND:20200731T143000
DTSTAMP:20200729T135257
CREATED:20200729T135203
SUMMARY:RV Téléphonique : Carbonnier
DESCRIPTION:
UID:ad61ab143223efbc24c7d2583be69251@cerclespa.fr
END:VEVENT
BEGIN:VEVENT
DTSTART:20200803T103000
DTEND:20200803T113000
DTSTAMP:20200729T135257
CREATED:20200729T135238
SUMMARY:Visite : Carbonnier
DESCRIPTION: Présentation des contrats
UID:d09bf41544a3365a46c9077ebb5e35c3@cerclespa.fr
END:VEVENT
END:VCALENDAR
```

4. Nouvelles tables

pass(id, annee, pass, created_at, updated_at)

profil_investisseurs(id, *client_id*, proFinance, action, actionConn, actionInvt, obligations, obligationsConn, obligationsInvt, opcv, opcvConn, opcvInvt, fondsStruct, fondsStructConn, fondsStrucInvt, projetRetraite, projetEpargne, projetTrans, invtPP, dureePP, invtPU, dureePU, proportionPatrimoine, partReservee, baisseEpargne, amplitudeVar, volatilitAccept, attitude, principeGestion, score, suggestionGestion, suggestionApproche, choixGestion, choixApproche, created_at, updated_at)

profil_investisseur_questions(id, etape, ordre, attribut, question)

profil_investisseur_occurences(id, *question*, ordre, occurrence, points, lettre, occurrenceCourt)

conseiller_documents(id, title, description, file, *conseiller_id*, *dossier_id*, *onglet_id*, created_at, updated_at)

dossier_acpr(id, ordre, name, created_at, updated_at)

onglet_acpr(id, *dossier_id*, ordre, name, created_at, updated_at)

docu_signs(id, *conseiller_id*, *client_id*, envoi, type, created_at, updated_at)

B. Glossaire

AJAX (Asynchronous JavaScript And XML) : méthode de communication asynchrone avec le serveur, afin de pouvoir modifier le contenu d'une page sans avoir à l'actualiser

API REST (Representational State Transfer) : architecture d'API offrant réactivité et fiabilité

CDN (Content Delivery Network) : réseau d'ordinateurs diffusant du contenu tel qu'un script ou une feuille de style permettant d'accélérer le chargement des pages ou des applications web

CMS (Content Management System) : logiciel destiné à la conception et à la mise à jour dynamique de sites ou applications web

CRON : diminutif de chrono table, table de planification, il s'agit d'un logiciel fonctionnant en arrière plan et qui permet de lancer des tâches planifiées

CSV (Comma-Separated Values) : format texte représentant les données d'une table sous forme de valeurs séparées par une virgule

framework : ensemble de composants logiciels servant de fondation pour le développement d'une application ou d'un logiciel

URI/URL (Uniform Resource Identifier/Locator) : chaîne de caractère identifiant une ressource réseau, l'URL fournit des moyens d'agir sur la ressource

widget : élément de l'interface affichant du contenu dynamique et interactif sur une page web

C. Bibliographie

AntheDesign – Le php oui, mais avec Laravel : <https://www.anthedesign.fr/developpement-web/php-avec-laravel/> (dernière consultation le 17/08/2020)

Easy Partner – Pourquoi choisir Laravel ? : <https://www.easypartner.fr/blog/pourquoi-choisir-le-framework-laravel/> (dernière consultation le 17/08/2020)

Documentation Laravel 5.8 : <https://laravel.com/docs/5.8/> (dernière consultation le 17/08/2020)

ITANÉA – Laravel et l'incontournable Laragon : <https://itanea.fr/apprendre-le-developpement-web/laravel-et-lincontournable-laragon/> (dernière consultation le 17/08/2020)

Documentation Carbon : <https://carbon.nesbot.com/docs/> (dernière consultation le 17/08/2020)

Laravel Phone : <https://github.com/Propaganistas/Laravel-Phone> (dernière consultation le 17/08/2020)

Twilio – How to Create an iCal Calendar Feed : <https://www.twilio.com/blog/how-create-ical-calendar-feed-php-laravel-lumen> (dernière consultation le 17/08/2020)

iCalendar Validator : <https://icalendar.org/validator.html> (dernière consultation le 17/08/2020)

Documentation Chart.js : <https://www.chartjs.org/docs/latest/> (dernière consultation le 17/08/2020)

Documentation jQuery Steps : <https://github.com/rstaib/jquery-steps/wiki> (dernière consultation le 17/08/2020)

Documentation jQuery Validation : <https://jqueryvalidation.org/documentation/> (dernière consultation le 17/08/2020)

Laravel MediaLibrary : <https://github.com/spatie/laravel-medialibrary> (dernière consultation le 17/08/2020)

DropzoneJS : <https://www.dropzonejs.com/> (dernière consultation le 17/08/2020)

Multiple File Upload with DropzoneJS and Laravel MediaLibrary :

<https://laraveldaily.com/multiple-file-upload-with-dropzone-js-and-laravel-medialibrary-package/> (dernière consultation le 17/08/2020)

DocuSign : <https://www.docusign.fr/> (dernière consultation le 17/08/2020)

DocuSign – Quickstart : Requesting a Signature via Email : <https://developers.docusign.com/esign-rest-api/code-examples/quickstart-request-signature-email> (dernière consultation le 17/08/2020)

Laravel DocuSign : <https://github.com/Tucker-Eric/Laravel-Docusign> (dernière consultation le 17/08/2020)

DocuSign Rest Client : <https://github.com/Tucker-Eric/docusign-rest-client> (dernière consultation le 17/08/2020)

The Official DocuSign PHP Client : <https://github.com/docusign/docusign-php-client> (dernière consultation le 17/08/2020)

DocuSign – PHP Launcher Code Examples : <https://github.com/docusign/code-examples-php> (dernière consultation le 17/08/2020)

Institut des sciences du Digital Management & Cognition
Pôle Herbert Simon
13 Rue Michel Ney
54000 Nancy

Stratégies & Perspectives d'Avenir
13 Boulevard de Baudricourt
54600 Villers-lès-Nancy