



Espace Multimédia

www.doranco.fr

Projet OpusBeauté

BEAUDEAU FREDERIC

Mémoire présenté devant le jury
pour l'obtention du titre de

Concepteur Développeur Informatique

Paris, Doranco Nation, le 27/03/2019





Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Table des matières

Table des matières	2
Dédicace	4
Remerciements	4
Résumé	5
Abstract	5
Table des figures et des tableaux	6
Table des sigles et abréviations	7
Présentation	9
Introduction	9
Au détour d'une conversation, un projet	10
Recueil des besoins de l'interface	10
Le projet en Agilité	11
1. Choix de la méthode Agile	11
2. Définir les rôles et acteurs Agiles	11
3. Créer des personas	11
4. Dictionnaire Métier	12
5. Les User Stories	13
6. La priorisation des user stories	14
7. Estimation des user stories	14
8. Le Backlog du projet :	16
9. Les technologies retenues	17
10. Suivi des Stories dans le tableau des tâches SCRUM	18
Conception	20
1. Architecture de l'application	20
2. Modélisation via UML	20
3. Diagramme de cas d'utilisation	20
4. Diagramme d'activité	22
5. Diagramme de séquence	23
6. Diagramme de classe	24
7. Conception d'une base de données via la méthode Merise	26



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

8.	Dictionnaire des données	26
9.	Le Model Conceptuel de données MCD	27
10.	Le Modèle Logique des Données MLD	27
	Maquettage	29
1.	Le nom de l'application « OpusBeauté ».....	29
2.	Le logo	30
3.	Les maquettes	30
4.	Les rubriques métier :.....	31
	LE DEVELOPPEMENT	31
1.	Déroulement du développement.....	31
2.	Le cycle de développement :.....	32
3.	La classes Objets métiers.....	32
4.	La classes DAO	33
5.	Les classes Services.....	36
6.	Les classes WebServices	37
7.	Exécution automatique de méthodes	38
8.	Les tests	39
9.	Test Driven Development (TDD) ou développement piloté par les tests	40
10.	Développement IHM (l'Interface Homme Machine).....	40
11.	Burndown Chart de Sprint.....	43
12.	L'intégration continue	44
13.	Le déploiement continu.....	45
14.	Envoyer des sms	45
	Stratégie d'hébergement	46
	Résultats	47
	La continuité du projet	51
	Conclusion	51
	Bibliographie.....	53



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Dédicace

Je dédie ce mémoire à ma famille pour le temps familial investi dans ce projet. Je suis intimement convaincu que cette démarche est une pierre supplémentaire à la consolidation de notre futur.

Remerciements

Je remercie infiniment ma famille de m'avoir encouragé et soutenu tout au long des processus de formation, de stage et travail du soir. Je suis fier d'avoir été entouré dans ce projet qui se révèle être bien plus qu'un projet personnel.

Je remercie ma tutrice Audrey PILLON qui m'a offert la possibilité réaliser mon stage dans son institut de beauté « La Bonbonnière d'Audrey » sur un projet cohérent avec ma formation.

Je remercie aussi le centre de formation Doranco pour les moyens humains et d'infrastructure mis à disposition pour réaliser cet apprentissage. Je félicite Doranco de mettre à disposition un personnel intervenant de qualité pour animer les formations. Je remercie l'intervenant Julien FERREIRA d'avoir rendu les modules de formation interactifs. Sa passion et sa disponibilité pour des questions après les cours m'a donné goût à la pratique et une affection aux langages Java et Javascript.

Je tiens à remercier monsieur GUEMKAM SIMO Philippe Fabrice pour ses précieux conseils lors des derniers ajustements de mon mémoire.

Aussi je remercie le centre de mettre à disposition une riche logithèque via 'Imagine' de Microsoft accessible gratuitement le temps du cycle d'étude.

Je remercie aussi le groupe d'étudiant avec qui j'ai suivi le cursus de la formation CDI, pour, l'ambiance sérieuse en salle et d'avoir généré des échanges dynamiques avec les intervenants. Ce qui contribue fortement à un cadre favorable d'apprentissage.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Résumé

Ce mémoire présente une solution logicielle chargée de répondre à la problématique des rendez-vous oubliés ou annulés de la part des clients. Cette situation est occasionnelle dans l'activité de ma tutrice mais elle mérite d'être étudiée dans le but d'accroître la qualité de service avec les clients ainsi que pour l'organisation interne. A ce jour, un simple calendrier Google est en place et sans interaction avec sa clientèle. Je propose de développer un logiciel chargé d'envoyer un SMS ou Email de rappel de rendez-vous la veille au client et au praticien concernés.

Pour réaliser ce projet je vais m'appuyer sur les techniques de gestion de projet Agile (SCRUM), Merise, et développer une solution logicielle basée sur une architecture orientée service. Pour mettre en œuvre ce type d'architecture je réaliserais la conception de la base de données pour enregistrer les informations puis d'un middleware pour la gestion des services ainsi que d'une Interface Homme Machine pour l'interaction humaine avec le logiciel. Et finissant la réalisation par l'hébergement du projet dans un Cloud privé de Microsoft.

Ma tutrice est auto-entrepreneuse dans le secteur de l'esthétique et n'a pas de budget à ce jour pour financer une solution professionnelle pour répondre à ce besoin. Le projet n'a pas vocation à être en concurrence avec d'autres produits payants sur le marché. Ce projet sera simplement une étude de faisabilité, et si par la suite ma tutrice souhaite exploiter le logiciel, celui-ci sera ajusté pour être porté du stade de POC (Proof Of Concept) vers une mise en production. Ma tutrice aura le choix d'exploiter le logiciel ou alors l'expérience justifiera l'achat d'une solution pro du marché.

Abstract

This thesis presents a software solution responsible for answering the problem of forgotten or canceled appointments by clients. This situation is occasional in the activity of my tutor but it deserves to be studied in order to increase the quality of service with the customers as well as for the internal organization. Today, a simple Google calendar is in place and there's no interaction with its customers. I propose to create a software responsible for sending message text or email as reminder of appointments the day before to the client and the practitioner. To realize this project I will use some of the project management techniques like SCRUM(Agile) and Merise, and develop a software solution based on a service-oriented architecture. To implement this type of architecture I will realize the design of the database to record the informations, then a middleware for the management of the services and a Graphical User Interface for human's interaction with the software.

My tutor is a freelancer in the field of Beauty Institute activity and has no budget to fund a professional solution to meet this need. The project is not intended to compete with other paying products. This project will simply be a feasibility study, and if my tutor wishes to exploit the software, it will be adjusted to be upgraded from Proof Of Concept to a production release. My tutor will have the choice to exploit the software or the experience will justify the purchase of a pro solution.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Table des figures et des tableaux

Tableau 1 : Tableau des Personas	P.12
Figure 1 : Désignation des personas	P.12
Tableau 2 : Dictionnaire Métier	P.12
Figure 2 : User stories	P.13
Figure 3 : Priorisation des User Stories	P.14
Figure 4 : Tailles T-shirts Estimation des User stories	P.15
Figure 5 : Backlog du projet	P.16
Tableau 3 : Technologies retenues Stockage des données SGBDR	P.17
Tableau 4 : Technologies retenues briques du middleware	P.17
Tableau 5 : Technologies retenues briques IHM	P.17
Tableau 6 : Technologies retenues Outils supplémentaires	P.18
Tableau 7 : Technologies retenues Api Extérieure pour envoyer des SMS	P.18
Tableau 8 : Tableau des taches (Scrum Task Board)	P.19
Figure 6 : Architecture Orientée Services (SOA)	P.20
Figure 7 : Diagramme de cas d'utilisation Accéder aux rubriques du logiciel	P.21
Figure 8 : Diagramme de cas d'utilisation ajouter une Prestation	P.21
Figure 9 : Diagramme d'activité s'authentifier	P.22
Figure 10 : Diagramme de séquence envoyer des SMS de rappel de rendez-vous	P.23
Figure 11 : Diagramme de séquence ajouter un nouveau rendez-vous	P.24
Figure 12 : Diagramme de classe Rendez-vous	P.25
Tableau 9 : Dictionnaire des données	P.26
Figure 13 : Modèle conceptuel de données (MCD)	P.27
Figure 14 : Modèle Logique des données (MLD)	P.28
Figure 15 : Premier Maquettage	P.29
Figure 16 : Plaquette exemple de palette de couleur	P.29
Figure 17 : Logo de l'application	P.30
Figure 18 : Maquettes IHM	P.30
Figure 18 : Objet Métier / entité annotation @Entity, @Table, @Id	P.32
Figurer 19 : Surcharge méthodes .toString(), ;hashCode(), .equals()	P.33
Figurer 20 : DAO perstance.xml, @Stateless, @Transactionnal, @PersistanceContext,	P.34
Figure 21 : Méthode pour récupérer un rendez-vous	P.34
Figure 20 : Méthode pour récupérer une liste de rendez-vous	P.34
Figure 21 : Méthode multi critères en SQL	P.35
Figure 22 : Requête en Jpql	P.35
Figure 23 : Annotation @Stateless pour transformer la classe Service en JavaBean	P.36
Figure 24 : Annotation @Path sur la classe WebService pour ressource URL	P.37
Figure 25 : Annotation @Get sur la classe WebServcie précise le verbe à utiliser	P.37
Figure 26 : Interface graphique du logiciel Postman	P.38
Figure 27 : Annotation @Schedule exécution de méthodes programmée	P.39
Figure 28 : Exemple de méthode de test	P.39
Figure 29 : Exécution de test	P.40
Figure 30 : Mvc (modele, vue, contrôleur) Design patern MVC	P.41
Figure 31 : Classe Objet en TypeScript	P.42
Figure 32 : Classe contrôleur pour la logique de la vue	P.42



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Figure 33 : Fichier Html+ CSS composition de la vue	P.42
Figure 34 : Classe Service chargée de faire des opérations via web services	P.43
Figure 35 : Burndown Chart de Sprint	P.43
Figure 36 : Interface du logiciel Jenkins	P.45
Figurer 37 : Raspberry Pi avec clé 3G et abonnement Free	P.46
Tableau 10 : Décomposition de l'Url de l'Api RaspiSMS	P.46
Tableau 11 : Statistique coût d'une machine virtuelle Azure hébergeant ObusBeauté	P.47
Figure 38 : Landing page et menu de login	P.48
Figure 39 : Authentification	P.48
Figure 40 : Home page avec les rendez-vous du jour	P.48
Figurer 41 : Barre de menu (slide)	P.48
Figure 42 : Ajouter un rendez-vous	P.48
Figure 43 : Sélectionner une date	P.48
Figure 44 : Sélectionner l'heure de début du rendez-vous	P.49
Figure 45 : Sélectionner un client	P.49
Figure 46 : Sélectionner un soin	P.49
Figure 47 : Rechercher un rendez-vous	P.49
Figure 48 : Modifier un rendez-vous	P.49
Figure 49 : Forcer l'envoi de rappel de rendez-vous	P.49
Figure 50 : Email de rappel de rendez-vous client	P.50
Figure 51 : Email de rappel de rendez-vous praticien	P.50
Figure 52 : SMS de rappel de rendez-vous client	P.50
Figure 53 : SMS de rappel de rendez-vous praticien	P.50
Figure 54 : Bibliographie	P.53

Table des sigles et abréviations

1. POC : Proof Of concept (démonstration de faisabilité / réalisation expérimentale)
2. ESN : Entreprise de Service du Numérique
3. PO : Product Owner
4. MoSCoW : Must have this ..., Should have this ..., Could have this ..., Won't have this ...
5. UML : Unified Modeling Language
6. MCD : Modèle Conceptuel des Données
7. MLD : Modèle Logique des Données
8. DAO Data Access Object (Design Pattern)
9. ORM : Object Relational Mapping
10. API : Application Programming Interface
11. JPA : Java Persistance API
12. CRUD : Opération de base en persistance (Create, Read, Update, Delete)
13. ACID : (Atomicité, Cohérence, Isolation, Durabilité) propriétés pour garantir une transaction
14. JPQL : Java Persistance Query Langage
15. SQL : Structured Query Language, langage de requête structurée
16. IOC : Inversion Of control Inversion de contrôle / Injection de dépendance
17. EJB : Entreprise Java Bean
18. Http : HyperText Transfert Protocole
19. JSON : JavaScript Object Notation



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

20. TDD : Test Driven Developpment (Développement piloté par les Tests)
21. MVC : Model Vue Contrôleur (Design Pattern)
22. Iaas : Infrastructure As a Service
23. Saas : Software As a Servvcie
24. DaaS : Desktop As a Service
25. Paas : Platform As a Service
26. Adsl : Asymmetric Digital Subscriber Line
27. Wifi : Wireless Fidelity



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Présentation

Je suis Frédéric BEAUDEAU, j'ai 43 ans, en couple, avec deux jeunes enfants de 3 et 5 ans. Je viens du monde informatique et je mettais en œuvre mes compétences pour assurer le bon fonctionnement du Système d'Information d'une entreprise Pharmaceutique. J'ai occupé un poste d'administrateur Systèmes et Réseaux pendant onze ans. Ce métier requiert entre autres des compétences de scripting. Mon intérêt pour le développement a vu le jour dans le cadre d'un projet pour le service Assurance Qualité. Le projet était d'automatiser des fichiers de fabrication au format PDF. Ce projet, abouti par un succès, m'a permis de pratiquer une activité de développement de façon autodidacte.

A la suite d'une rupture conventionnelle, j'ai effectué une formation sur le BigData, et j'ai compris pendant le cursus qu'il était incontournable de maîtriser un langage objet. Ce domaine requiert des langages pour notamment automatiser des opérations d'extraction, de transformation... de conséquentes volumétries de données issues de différents métiers. Au terme de cette formation, il était clair que je devais ajouter les techniques développement de logiciel à mon panel de compétences. Pour réaliser cette démarche, je me suis appuyé sur le site « défi-métier » ou j'y ai trouvé une formation conventionnée dispensée par le centre de formation « Doranco », et pour aller jusqu'au bout de mon projet de formation, j'ai engagé mes congés personnels de formation (CPF) capitalisés ces onze dernières années et j'ai complété le restant dû par un financement personnel.

La démarche de faire le point sur mes acquis, d'identifier les besoins du marché et d'observer des axes de progrès me permet d'avoir une vue d'ensemble dans le vaste milieu de l'informatique. Cette démarche d'amélioration continue de carrière m'offre la possibilité de mettre en œuvre des compétences dans l'Infrastructure, la gestion de projets et la conception de logiciels. Mon nouveau Curriculum vitae a retenu l'attention d'une Entreprise de Service en informatique (ESN). L'ESN Informatique Nord2 procède à une réorganisation de son fonctionnement interne et dans ce remaniement mon profil correspondait à cette nouvelle organisation.

Introduction

Au détour d'une conversation entre Ingrid, ma concubine, et Audrey une professionnelle dans l'esthétique, Audrey exposait des réflexions pour développer son activité. Le sujet d'analyse ciblait l'amélioration de la qualité de service entre le client et la praticienne. Le sujet évoquait principalement les « ratés » dans la planification des rendez-vous. Ces ratés se manifestent généralement par une annulation au dernier moment, un report ou un oubli d'un rendez-vous par un client.

A l'écoute de la problématique, je me suis permis d'intervenir et de demander quels étaient le / les outils pour la gestion des rendez-vous. Audrey est auto-entrepreneuse et la gestion de ses rendez-vous est artisanale avec l'outil gratuit de Google pour la planification dans l'agenda. Google proposait l'envoi d'email / Sms au propriétaire de l'agenda (adresse email) mais pas pour le client. A ce jour, Google a supprimé le service d'envoi de sms au propriétaire de l'agenda.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Au détour d'une conversation, un projet

C'est alors que j'ai imaginé une solution personnalisée pour répondre à ce besoin métier :

- Saisir un rendez-vous pour une prestation entre un Client et le Praticien.
- Envoyer à J-1 un email et ou sms de rappel de rendez-vous aux deux parties.

Juste après, j'ai proposé mon idée de développement d'un logiciel personnalisé qui correspondrait en même temps à mon projet de soutenance pour valider ma formation. L'idée de la solution sur mesure pour répondre à ce problème fut favorablement reçue par Audrey.

C'est alors que le périmètre de ce projet a été délimité avec la contrainte que cela ne devait pas engendrer de frais pour son activité. Alors nous avons décidé de réaliser ce projet dans le cadre d'un POC (Proof Of Concept) sans obligation d'utilisation. Un POC est une démonstration de faisabilité, une réalisation expérimentale concrète et préliminaire, courte ou incomplète, illustrant une / des méthodes ou idées afin d'en démontrer la faisabilité.

Ce cadre étant compatible avec mon projet de fin de cycle d'apprentissage, j'ai pu réaliser mon stage à l'institut de beauté pour initier le cycle de conception, ainsi qu'une partie de l'application.

Recueil des besoins de l'interface

L'application se situe dans la catégorie des outils fonctionnels et sera réfléchie comme une application avec une apparence de type Intranet. L'interface est principalement destinée au praticien / utilisateur. La partie cosmétique de l'interface devra être en cohérence avec la charte graphique de l'institut. Le point départ de la charte graphique s'appuiera sur la palette de couleurs de la brochure tarifaire de l'institut.

L'interface sera pensée pour être le plus ergonomique possible, la navigation sera logique, simple et intuitive. L'apparence doit être personnalisée avec un ou des logos en rapport avec l'activité de l'esthétique.

En général, la prise de nouveaux rendez-vous se fait en présence de la clientèle à la fin d'une prestation. L'utilisation de l'application sera principalement réalisée depuis un ordinateur portable mais pour les situations de mobilité, l'accès à une version mobile serait un plus.

Les besoins orientés métier :

- **Le praticien / utilisateur doit pouvoir s'identifier** (il est préférable de protéger l'accès à l'application aux personnes accréditées)
- **Ajouter un Rendez-vous** (à minima un Rdv comporte les informations de base comme la date et l'heure. Il faudra ajouter des informations supplémentaires : le choix d'un client(e), d'un(e) praticien(ne), d'une prestation, d'un lieu de prestation ainsi que la durée)
- **Mettre à disposition les prestations existantes** (les prestations existantes seront importées)



- **Ajouter une Prestation** (le périmètre d'activité n'est pas figé, Audrey peut ajouter des nouvelles Prestations)
- **Ajouter un Client** (avec le développement de son activité, Audrey est en mesure d'ajouter de nouveaux clients)
- **Ajouter un Praticien** (si Audrey fait évoluer son personnel, il lui sera possible d'intégrer de nouveaux praticiens)
- **Ajouter un Utilisateur** (Audrey peut donner accès à l'application à de nouveaux utilisateurs)

Le projet en Agilité

1. Choix de la méthode Agile

Bien qu'à ce jour des articles inondent le web, j'ai découvert la méthode Agile au cours d'un module prévu dans le cursus de la formation. Une partie de ce module m'a perturbé, car les projets que j'ai réalisés dans le système et réseaux me permettaient de mettre en œuvre les phases de tests avec le Service Qualité qu'en fin de réalisation. La partie de gestion de projet Agile qui m'a surpris est la Sprint Review (livrable) une démonstration d'un produit en cours de développement. Celle-ci a bousculé mes habitudes de gestion de projets en cascade (cycle en V). Mais ce flottement n'a pas duré longtemps car l'aspect du FeedBack utilisateur m'a beaucoup plu. J'ai découvert que le feedback est un élément qualitatif et précieux pour se rapprocher au mieux de ce qu'attendent les utilisateurs. En cas de trop gros écart entre le Sprint attendu par le client et le Sprint réalisé, il sera possible de réintroduire le Sprint dans le BackLog.

Parmi le vaste panel de méthodes Agiles disponibles comme l'Extrem programming, kanban, Lean Développement, Feature Driven Development Développement et enfin la méthode SCRUM, je vais m'appuyer sur cette dernière pour organiser le déroulement de mon projet. Cela me permettra de mettre en œuvre la méthode abordée en formation. L'objectif sera de se familiariser avec ce cadre de travail fourni par la méthode et d'observer la satisfaction utilisateur, procéder par itération en livrant de nouvelles fonctionnalités et de récupérer le retour utilisateur/client.

2. Définir les rôles et acteurs Agiles

Le fait d'être seul dans la réalisation de ce projet de développement, me permettra de jouer tour à tour les rôles de PO (Product Owner), Scrum Master et de l'équipe de développement. Pour l'occasion, il faudra essayer de cibler les actions, les tâches et responsabilités de chacun des rôles. Ma cliente jouera le rôle de Stakeholder : il est évident que c'est la personne la mieux placée pour occuper cette place.

3. Créer des personas

La méthode préconise de commencer par la création de personas, ce sont des acteurs virtuels chargés de représenter des personnes ou systèmes afin à décrire des interactions. En tant que Product Owner, je procède à la création de mes personas : seulement trois suffiront à décrire les situations.

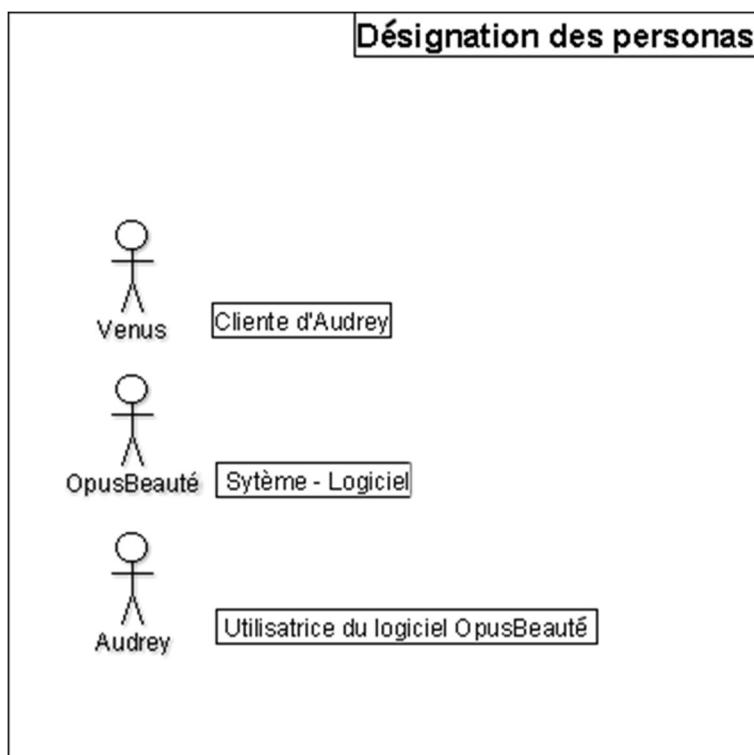


Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Ma cliente a souhaité nommer le système OpusBeauté qui est un clin d'œil à son enfance de musicienne et de sa profession actuelle.

Nom	Définition
Audrey	Audrey est auto entrepreneuse, l'institut de beauté est son activité, elle sera utilisatrice et praticienne.
Venus	Venus sera une cliente de l'institut de beauté.
Opus beauté	OpusBeauté sera le système



4. Dictionnaire Métier

Voici une tape déterminante dans la modélisation de l'application : la définition du dictionnaire métier. Avec ma tutrice, nous avons listé les mots métiers de son activité, ce qui a permis de réaliser la convention de nommage des concepts métiers. L'avantage de ce procédé est d'assurer une cohérence sur l'ensemble de l'application entre les parties IHM, middleware, base de données.

Nom	Définition
Rendez-vous	Planification d'une date et une heure entre un client, un praticien et un lieu pour la réalisation d'une prestation et de sa durée.
Lieu Rendez-vous	C'est l'adresse du lieu de rendez-vous
Rappel de rdv	Email ou sms pour rappeler le rendez-vous planifié.
Client	Personne qui bénéficie d'une prestation.
Adresse Client	Est l'adresse du client
Genre	Genre humain homme, femme.
Praticien	Personne qui pratique le soin



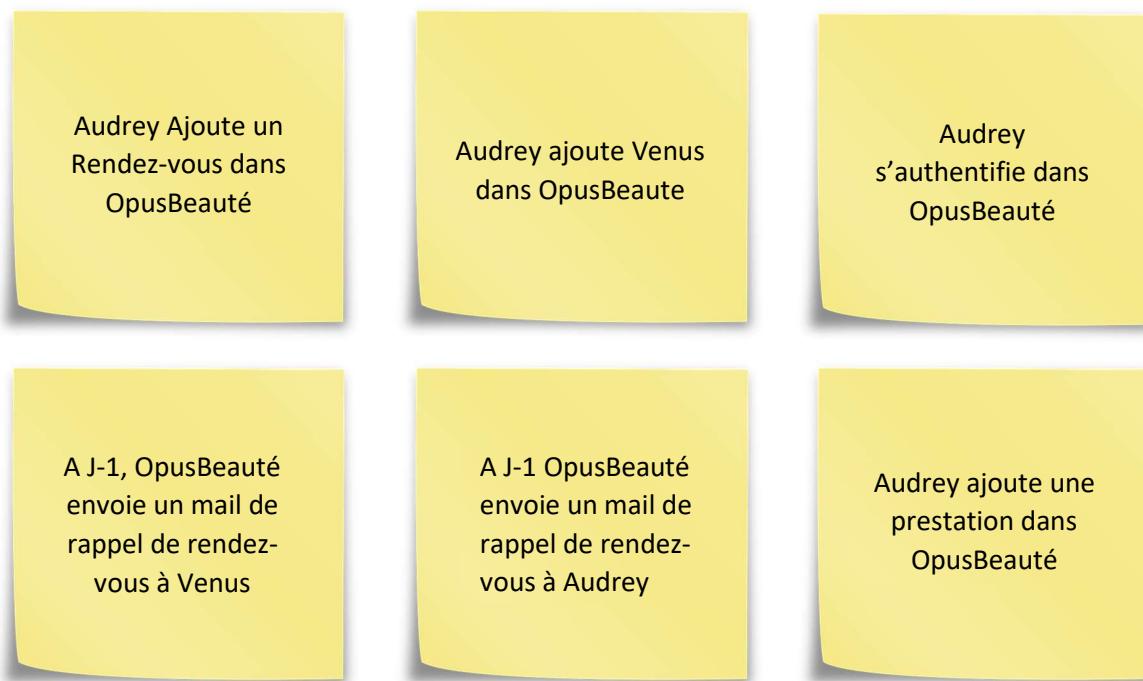
Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Utilisateur	Personne qui utilise l'interface du logiciel
Prestation	Soin qui sera prodigué par un praticien
Activité	Catégorie pour classer les soins
Soin	Technique qui sera prodiguée à la clientèle
Forfait	Le soin fait-il l'objet d'un forfait de plusieurs séances ?

5. Les User Stories

En tant que Product Owner et avec ma tutrice, nous avons procédé à la création des User Stories pour verbaliser les fonctionnalités nécessaires dans le logiciel. A l'aide du dictionnaire métier, des Personas et de la liste des besoins nous avons les supports nécessaires pour être en mesure de les créer. Pour la réalisation de cette activité, la convention recommande de rédiger une User Story sous la forme suivante : « En tant que » :qui (persona) « je voudrais » : quoi (fonctionnalité) « afin de » : objectif. Pendant le module de formation, nous avons pratiqué cette activité en la modifiant légèrement afin de rendre la User Story moins verbeuse. Je vais reproduire la forme syntaxique de ce que l'on a vu en cours. La User Story simplifiée sera rédigée de la façon suivante : Qui (persona) + verbe d'action + complément.



Quelques exemples de user stories



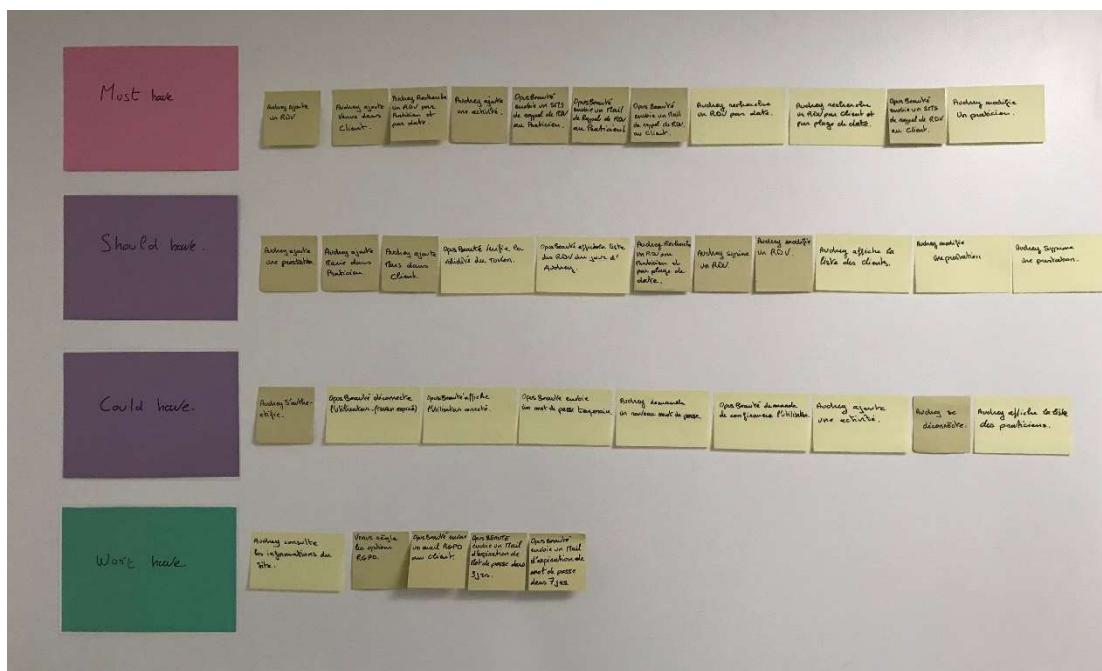
Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

6. La priorisation des user stories

L'objectif de cette activité est d'ordonnancer les stories afin de les prendre en charge selon une estimation de priorité. La priorité est dans un premier temps évaluée par ma tutrice selon son prisme métier. Pour mettre en place cette activité, nous avons utilisé la méthode **MOSCOW**, cette méthode utilise les lettres en anglais, celles-ci représentent des abréviations de phrase liées à la priorisation :

- M pour Must have (il faut entendre une priorité Obligatoire).
- S pour Should have (il s'agit plutôt d'une priorité Recommandée)
- C pour Could have (là nous sommes sur une priorité de Suggestion)
- W pour Won't have (et enfin cela relève du Souhait)



7. Estimation des user stories

Cette activité sert à estimer la complexité d'une User Story. L'évaluation n'est pas évidente et requiert beaucoup d'attentions car il y a plusieurs facteurs à observer.

Il y a la nature de la story, elle peut être d'une grande technicité et avoir besoin d'intégrer beaucoup de mécanismes pointus nécessitant un travail conséquent de documentation. Il faut aussi prendre en compte le niveau de compétences techniques de l'équipe de développement.

L'évaluation des User Stories n'est pas une chose aisée dans les premières expériences car il est difficile d'évaluer la réelle complexité de celles-ci. Seule l'expérience, la répétition me permettra d'affiner les évaluations vers plus de justesse.

Pour réaliser cette étape, je me suis servi d'un système de taille de t-shirt : cette méthode a aussi été présentée en formation. Je vais donc m'en servir pour évaluer la taille de mes User stories. La technique d'évaluation sera celle des T-shirts et comportera 4 tailles : S, M, L et XL.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation



Cartes Scrum Game agilbee

Application des tailles à des niveaux de difficulté :

- Lettre S : Pour une User Story estimée être facile et rapide.
- Lettre M : Pour une User Story estimée être moyenne en temps et simple techniquement.
- Lettre L : Pour une User Story estimée être longue et/ou complexe mais disposant d'un code existant similaire déjà développé.
- Lettre XL : Pour une User Story estimée être longue et complexe et nécessite surtout de la documentation.

L'attribution de la taille aux différentes User Stories sera faite comme en formation. Je cherche dans mon volume de User Stories celles qui sont susceptibles de correspondre à une taille M. Cette approche me permet d'avoir un référentiel pour évaluer les autres User Stories. Un exemple d'attribution : pour la User Story : « Audrey ajoute un client », celle-ci aura une taille M. La User Story « Audrey Ajoute un praticien » aura quant à elle une taille S, moins grande, car j'ai déjà développé une partie du code, ne nécessitant qu'une adaptation. Dans mon projet, une bonne partie des User Stories sont en XL car ce projet est mon premier développement et bien sûr, je n'avais pas encore de code existant. Les User Stories nécessitant des mécanismes d'authentification, les tokens, les intercepteurs l'envoi de mails, de sms, les Services Timers ont requis documentation approfondie.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

8. Le Backlog du projet :

Audrey ajoute Venus dans client.	Audrey recherche un RDV par client et par plage de date.	OpusBeauté affiche l'utilisation correcte.	OpusBeauté demande de configurer l'utilisation.
Audrey ajoute une prestation.	Audrey recherche un RDV par date.	OpusBeauté vérifie la validité du token.	OpusBeauté déconnecte l'utilisateur. (token expiré)
Audrey ajoute Plans dans Client.	Audrey Recherche Un RDV pour Praticien et par plage de date.	Audrey ajoute une activité.	Audrey demande un nouveau mot de passe.
Audrey ajoute un RDV	Audrey modifie Un RDV.	Audrey modifie Un praticien.	OpusBeauté envoie un mot de passe temporaire.
OpusBeauté envoie un mail de rappel de RDV au Client.	Audrey supprime un RDV.	Audrey Supprime une prestation.	OpusBeauté envoie un mail d'expiration de mot de passe dans 7Jrs.
OpusBeauté envoie un SFS de rappel de RDV au Praticien.	Audrey ajoute Planie dans Praticien.	Audrey modifie une prestation	OpusBEAUTE envoie un mail d'expiration de Mot de passe dans 3Jrs.
OpusBeauté envoie un mail de Rappel de RDV au Praticien.	OpusBeauté affiche la liste des RDV du jour d' Audrey.	Audrey affiche la liste des clients.	Audrey consulte les informations du site.
OpusBeauté envoie un SFS de rappel de RDV au Client.	Audrey s'authenti- fifie.	Audrey ajoute une activité.	OpusBeauté envoie un mail RGPD au client.
Audrey Recherche Un RDV par Praticien et par date.	Audrey se déconnecte.	Audrey affiche la liste des praticiens.	Venus régule les options RGPD.

Sens de lecture : 1^{ère} colonne de haut en bas et de gauche à droite



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

9. Les technologies retenues

Pour ce projet d'application web n-tiers, j'ai opté pour les technologies suivantes :

Stockage des données	
SGBDR	
 PostgreSQL	J'ai retenu ce gestionnaire de base de données relationnelle pour deux raisons : PostgreSQL mais est fondé sur une communauté mondiale de développeurs et d'entreprises, ce qui fait qu'on y trouve une très bonne documentation et des forums actifs.
 pgAdmin Version 4 Management Tools for PostgreSQL	En plus de l'interface de ligne de commande fourni par PostgreSQL, pgAdmin est une interface graphique pour l'administration du SGBDR. Un super outil pour tester facilement ses requêtes et bien d'autres actions.

Ensemble briques qui composent le middleware	
	Wildfly est un des serveurs d'application qui prend en charge les spécifications Java notamment, offre le support Java EE, Hibernate, JPA ...
 HIBERNATE	Outils incontournable pour mettre en œuvre la persistance d'objets avec une base de donnée.
	Cet outil sert à automatiser la compilation des dépendances et code source du projet.
	J'ai choisi ce langage Objet car c'est un langage fortement typé, il est populaire, propose une bonne documentation et beaucoup d'articles sont présents sur la toile.
	Ide dont je me sers pour développer en JavaEE.

Ensemble d'outils pour la partie IHM	
	C'est un Framework côté client basé sur le langage TypeScript. J'ai choisi Angular pour plusieurs raisons. J'ai vu ce Framework en formation, mais le concept de gérer les vues de façon éclatées dans trois fichiers m'a paru plus facile dans mon apprentissage du développement d'IHM.
	C'est un langage fortement typé comme pour Java. TypeScript est un langage transcompilé en Javascript afin d'être exécuté sur la majorité des navigateurs.
	AngularCli est un outil très pratique afin de fournir une « Command Line Interface » pour interagir par des lignes de commandes sur diverses opérations de création de vue, de service, de directives, compilation du projet...
	Moteur bas niveau servant à exécuter du code JavaScript côté serveur. Une multitude de Frameworks sont disponibles, il me servira pour Angular. Il possède plusieurs particularités dont le fait d'être asynchrone.

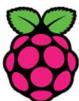


Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

 Angular Material	Bibliothèque mise à disposition gratuitement par Google. Il faut l'intégrer dans son projet Angular pour disposer d'objets visuel prêts à l'emploi.
	Visual Studio Code est l'IDE de Microsoft gratuit et très pratique pour développer en TypeScript. Une multitude de plugins gratuits sont à disposition.

Outils supplémentaires	
	GitHub est un service de repository et de versionning. Je m'en sers pour déposer mon code source et cela gère aussi le versionning.
	Outils très pratique servant à tester les Web Services lors de développement de middleware. On peut s'en servir pour tester des APIs (Application Programming Interface) tiers exposant des Web Services que l'on souhaite intégrer.
	Outil pour l'intégration continue, il est couplé avec mon compte GitHub. A chaque nouvelle version que j'envoie (push de mon code sur GitHub), Jenkins est déclenché, il récupère la nouvelle version, la compile en passant en revue les tests et valide ou non la Build.

API extérieure pour envoyer des SMS	
API RaspisMS	Distribution linux en partie paramétrée dont la particularité est de mettre à disposition une API pour envoyer des sms.
	Mini ordinateur au format carte de crédit. Très apprécié par la communauté de makers.
	J'ai acheté pour mon projet une carte usb 3G pour la connecter au RaspberryPi. Celle-ci me permet de faire la passerelle et d'accéder au réseau 3G pour envoyer des sms.
	Etant abonné à free, je dispose d'une carte sim à 0 euros pour envoyer gratuitement des sms.

10. Suivi des Stories dans le tableau des tâches SCRUM

L'état d'avancement des User Stories est suivi par le tableau des tâches SCRUM, défini pour organiser visuellement les sprints qui vont être déplacés horizontalement de colonne en colonne.

Le cheminement des Sprints sera organisé dans un tableau de 4 colonnes :

- 1 : « A faire » c'est l'emplacement d'une story sur le starting block,
- 2 : « Analyse » c'est le moment de réflexion de documentation pour préparer la ou les techniques à utiliser dans cette User Story,
- 3 : « Développement » c'est la partie où la User Story est convertie en code,



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

- 4 : « Fini » c'est l'état d'une Story User complètement terminée dans son intégralité middleware + IHM (passage à la suivante).

Je me suis permis d'ajouter un code couleur associé au développeur, et, sur le post-it de la User Story, je colle une gommette de la couleur du développeur qui est en charge de la réaliser. Cela permet de visualiser très rapidement par qui la User Story est prise en charge. Ceci est valable dans une équipe de plusieurs développeurs.

Il est important de bien stipuler l'état « terminé » d'une User Story. Il s'agit de préciser l'état de (Définition Of Done), pour ce projet, je vais définir l'état de Terminé (DONE) d'une User Story qui aura fait l'objet d'un processus de réflexion, de développement de WebServices, du développement de l'IHM. Le « DONE » sera valide à condition que les tests confirment que la nouvelle fonctionnalité est fonctionnelle dans l'IHM et surtout qu'il n'y a pas de comportements anormaux.

J'ai aussi ajouté une contrainte sur le processus de prise en charge des sprints, le Work In Progress (Wip). Cette sorte de verrou va m'éviter d'avoir un grand nombre de sprint en cours de traitement sans aller jusqu'au DONE. J'ai décidé de fixer le WIP à 2 sprints pour les colonnes en « réflexion » et « développement » et la colonne test n'autorisera qu'un seul sprint. En réalité, je n'ai pas eu besoin de ce type de régulateur car j'ai veillé à n'avoir que deux sprints en simultané.

Tableau des tâches (Scrum Task Board)

A faire	Analyse	Dev	Wip L 2	Wip L 1
				Audrey ajoute un RDV
			Audrey ajoute une prestation.	
Audrey ajoute Venus deux Client.				
	Opus BeauVé envoie un mail de rappel de RDV au Client.			
		Opus BeauVé envoie un SETS de rappel de RDV au Practiseur.		
			Opus BeauVé envoie un mail de rappel de RDV au Practiseur.	

Scrum board



Projet OpusBeauté

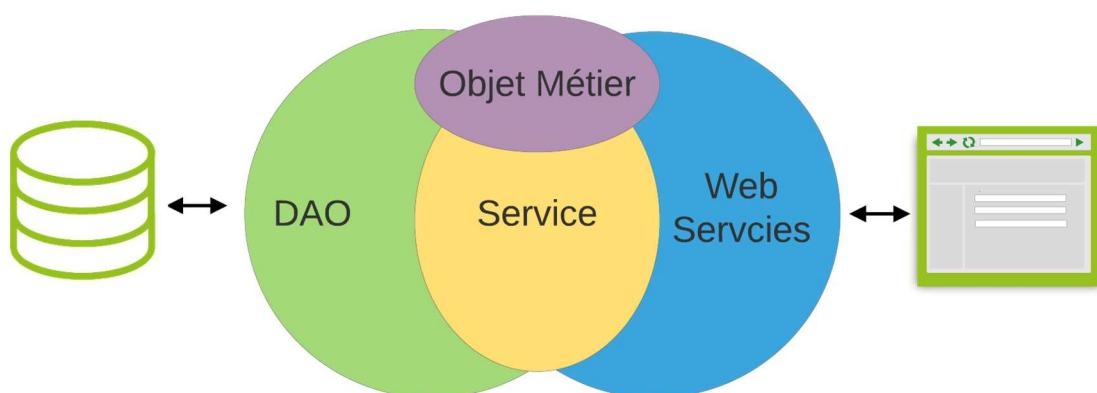
Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Conception

1. Architecture de l'application

J'ai choisi de structurer mon application sur une architecture de type Service SOA (Services Oriented Architecture). La conception devra répondre à la séparation des principales parties du logiciel pour qu'elles ne soient pas liées. L'Interface homme-machine, le middleware et la base de données seront trois parties indépendantes. La communication entre les parties sera possible à travers le réseau via le protocole HTTP. Les étapes réalisées en amont avec le Framework Agile sont cruciales dans la modélisation des aspects métiers et des données à faire transiter. La répartition en « couche » de l'application va permettre un isolement des trois parties. Le choix des technologies pour réaliser cette abstraction est important pour être en phase avec ce type d'architecture.

L'idée est de mettre le middleware au cœur de la solution entre l'IHM et la base de données. L'IHM communiquera via le réseau avec le middleware à travers la consommation de WebServices RESTfull afin d'exploiter les informations. Et à son tour le middleware réalisera des actions en fonction des règles métier définies et sera chargé, si besoin, de communiquer avec la base de données via le réseau. Le mécanisme pour accéder aux données est réalisé à l'aide du service DAO (design pattern Data Access Object). Ainsi on peut assurer une indépendance des trois parties et d'assurer la maîtrise des données accessibles depuis l'IHM. La description de conception avec la séparation des couches démontre la mise en place d'un schéma d'une architecture n-tiers.



2. Modélisation via UML

Cette activité met en œuvre la modélisation de l'application. Pour réaliser cette étape, je vais utiliser UML (Unified Modeling Language). Via ce langage de modélisation graphique, je vais réaliser des visuels de conception du système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

3. Diagramme de cas d'utilisation

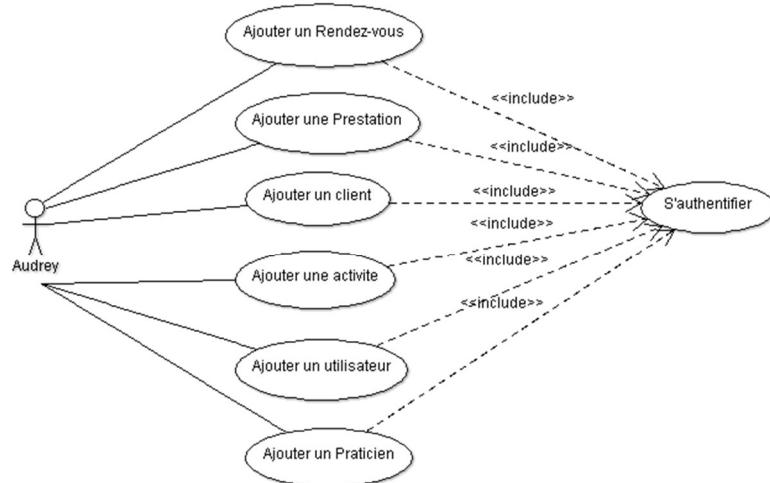
Ce type de visuels mettent en scène l'interaction fonctionnelle des acteurs avec le système.



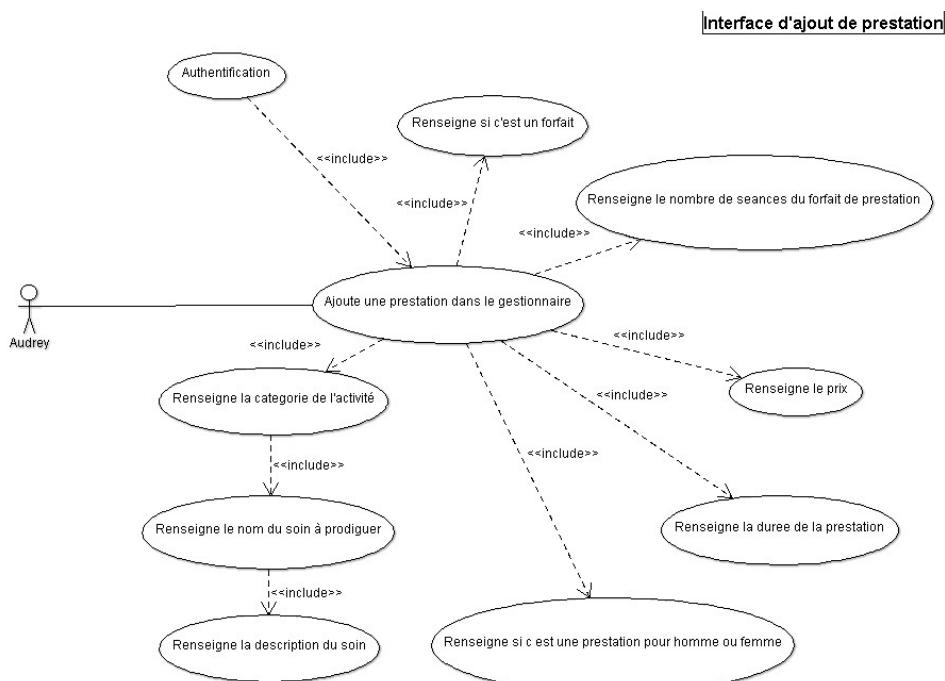
Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Accéder aux rubriques du logiciel OpusBeauté



Ce cas d'utilisation permet la description des actions possibles par Audrey.



Ce cas d'utilisation décrit les actions possibles sur l'interface d'ajout d'une Prestation

Ces deux cas d'utilisation me permettent d'avoir une idée assez claire des fonctionnalités que je dois faire apparaître dans l'interface utilisateur.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

4. Diagramme d'activité

Le diagramme d'activité permet de visualiser le déroulement d'un processus et met à plat une logique d'évènements.

Diagramme d'activité « s'authentifier »

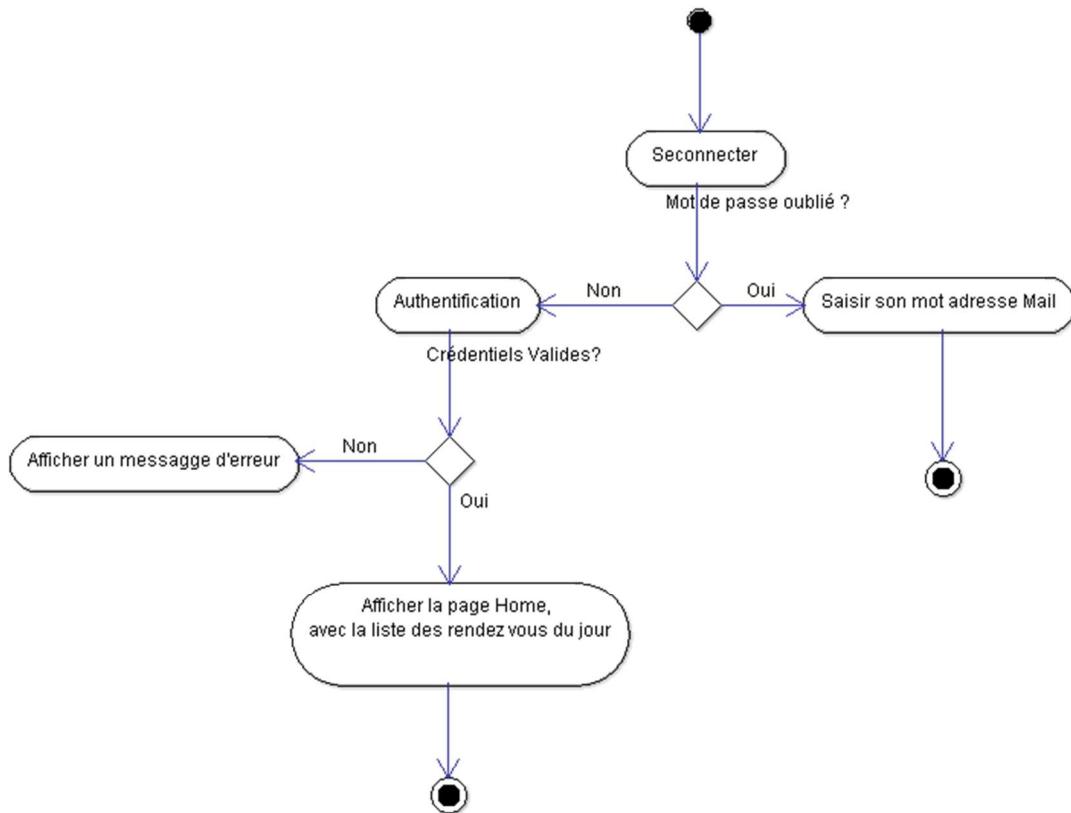


Diagramme d'activité décrivant mon processus de login dans l'application OpusBeauté

Ce diagramme décrit le processus logique imaginé lors de l'authentification dans l'application OpusBeauté.



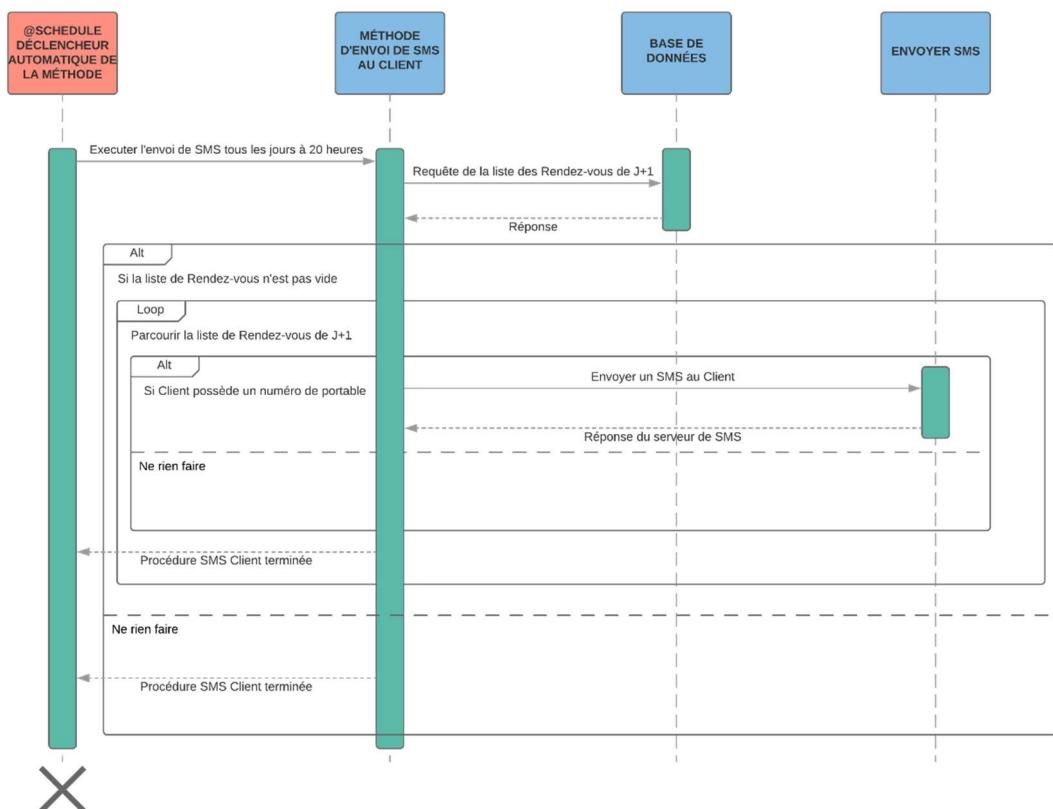
Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

5. Diagramme de séquence

Le diagramme de séquence me permet de réaliser une représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique et de vérifier la logique imaginée pour la séquence d'authentification.

ENVOYER DES SMS DE RAPPELS DE RENDEZ-VOUS A J+1 AU CLIENT



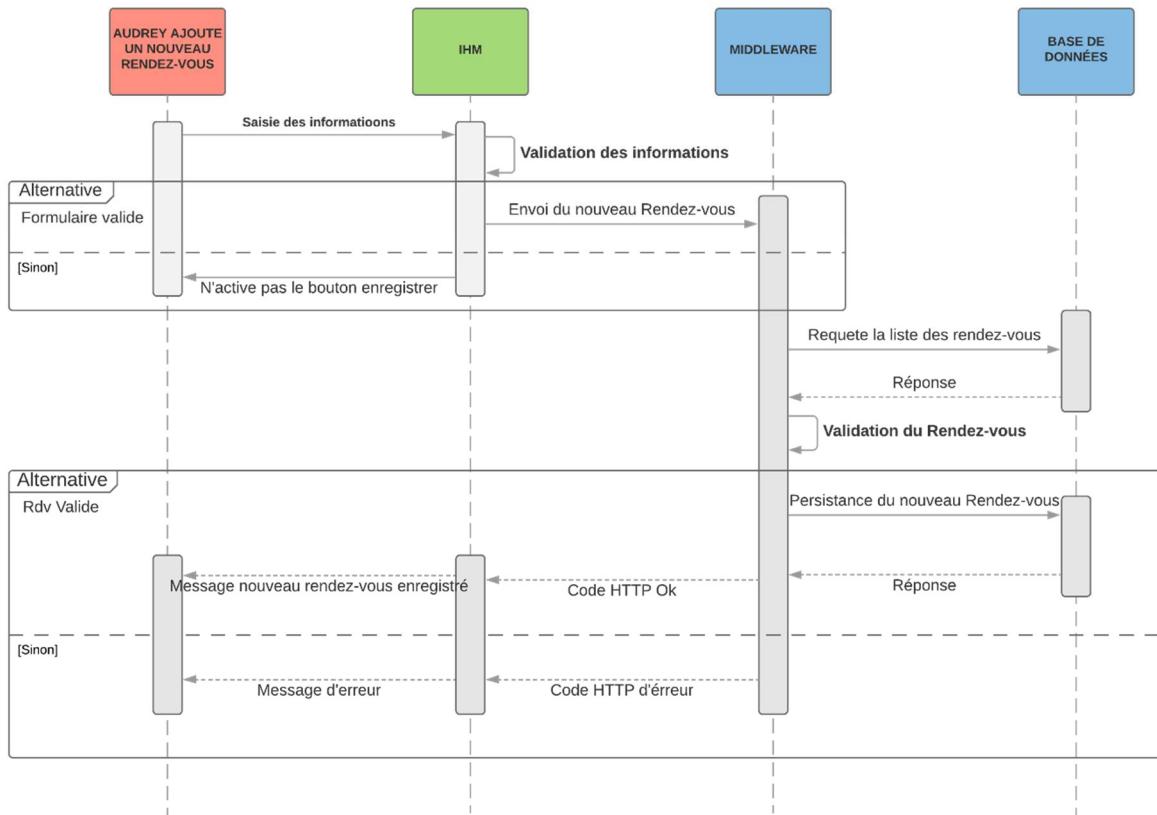
Ce diagramme de séquence est chargé de montrer les échanges lors de la procédure d'envoi de messages.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

DIAGRAMME DE SEQUENCE AJOUTER UN NOUVEAU RENDEZ-VOUS



Ce diagramme me permet de vérifier les échanges de messages entre les différents composants de l'application.

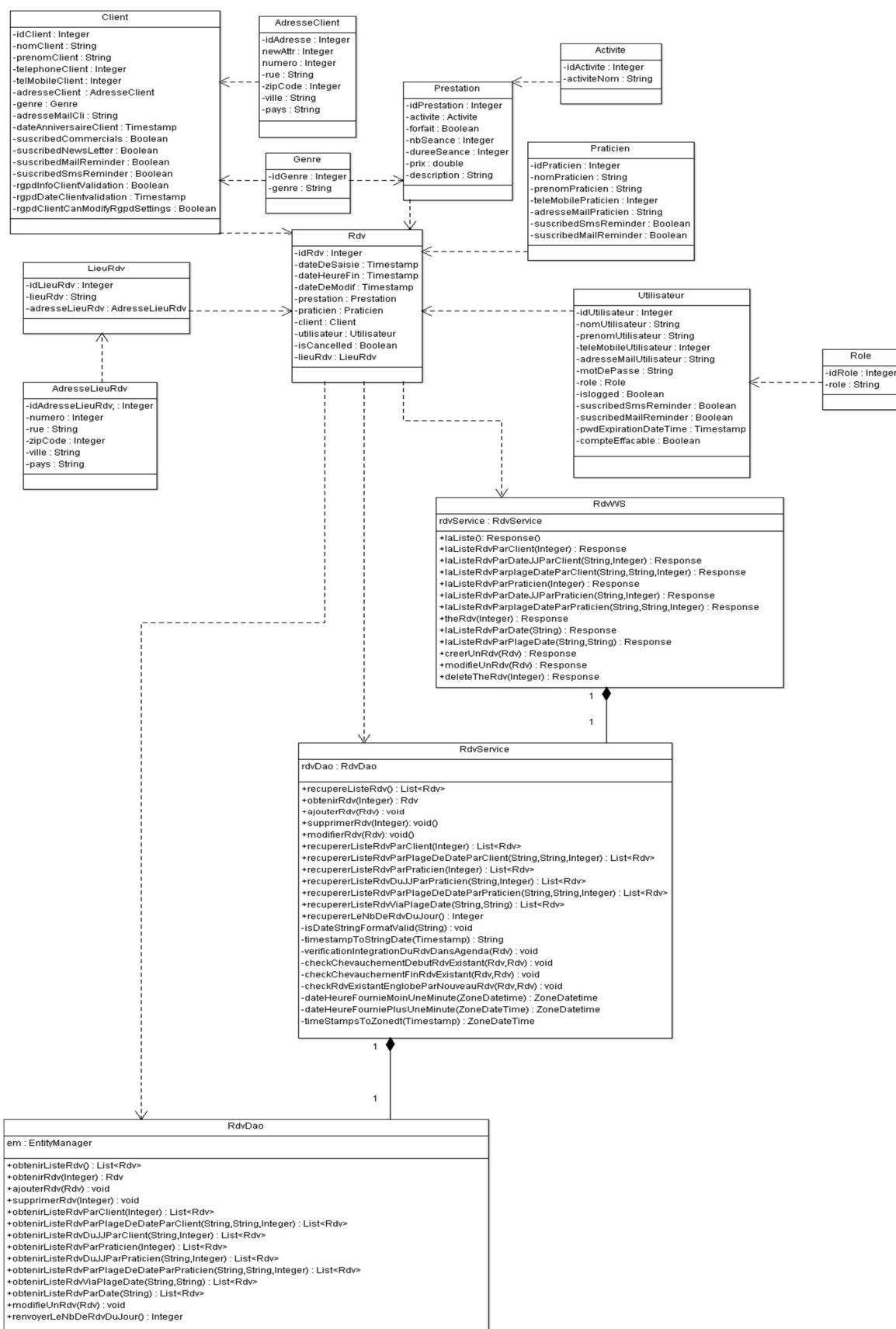
6. Diagramme de classe

Le diagramme de classe est une étape fondamentale dans la conception orientée objet, ce sont les visuels permettant une représentation des éléments qui composent un système et leurs relations. A la phase de développement d'un sprint, je serai en mesure de m'appuyer sur ce genre de visuel pour la modélisation de l'application ainsi que des fonctionnalités.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation



Ce diagramme montre l'interaction entre les classes pour les cas d'utilisations dédiés au Rendez-Vous.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Cette illustration met en évidence l'architecture orientée service par la séparation des rôles de la persistance des données (RdvDao), la gestion des règles métier (RdvService) et des WebServices (RdvWS) chargés consommer et produire du contenu à travers le protocole Http à destination de l'IHM.

7. Conception d'une base de données via la méthode Merise.

La méthode Merise existe depuis les années 70. C'est une méthodologie de modélisation à usage général dans le domaine du développement de systèmes d'information. Je vais faire l'usage d'une méthode Merise basée sur le modèle entité-relation. En utilisant Merise, je serai en mesure de concevoir des tables avec des relations pour créer une base de données relationnelle.

8. Dictionnaire des données

Le dictionnaire des données est un document qui regroupe toutes les données à conserver dans la base (et qui figureront donc dans le MCD). Pour chaque donnée, il indique :

- Le code mnémonique : il s'agit d'un libellé désignant une donnée.
- La désignation : il s'agit d'une mention décrivant ce à quoi la donnée correspond (par exemple « Nom client »).
- Le type de donnée : Alphabétique, Numérique, Alphanumérique, Date, Booléen ...
- La taille : elle s'exprime en nombre de caractères ou de chiffres.
- Remarque : peut indiquer les contraintes.

Code Mnémonique	Désignation	Type	Taille	Remarque
Client_idClient	Identifiant Client	Auto_increment		Not null
Client_nomClient	nom du Client	Varchar	30	
Client_prenomClient	prénom du Client	Varchar	30	
Client_adresse	Adresse du Client	Varchar	30	
Client_telephone	Téléphone du Client	Integer	10	
Client_dateAnniversaire	date Anniversaire du Client	Date		format aaaa-mm-jj
Client_subscribedCommercials	Si le Client est abonné à la newsLetter	Boolean		
Client_subscribedSmsReminder	Si le client est abonné au rappel de Rdv par Sms	Boolean		
Client_subscribedMailReminder	Si le client est abonné au rappel de Rdv par Mai	Boolean		
Client_subscribedNewsLetter	Si le client est abonné à la NewsLetter	Boolean		
Adresse_IdAdresse	Identifiant Adresse	Auto_increment		Not null
Adresse_numero	Numéro de la rue	Integer	3	
Adresse_rue	Rue	Varchar	30	
Adresse_ville	Ville	Varchar	30	
Adresse_zipCode	code postale	Integer	5	
Adresse_pays	Pays	Varchar	20	
Genre_idGenre	Identifiant Genre	Auto_increment		Not null
Genre_genreHum	Genre Humain	Varchar	5	
Rdv_idRdv	Identifiant Rendez-vous	Auto_increment		Not null
Rdv_dateDeSaise	Date de saisie du Rdv	Date		format aaaa-mm-jj
Rdv_dhDebut	Date et heure du début du Rdv	TimeStamp		
Rdv_dhFin	Date et heure de la fin du Rdv	TimeStamp		

Dictionnaire des données (partiel)

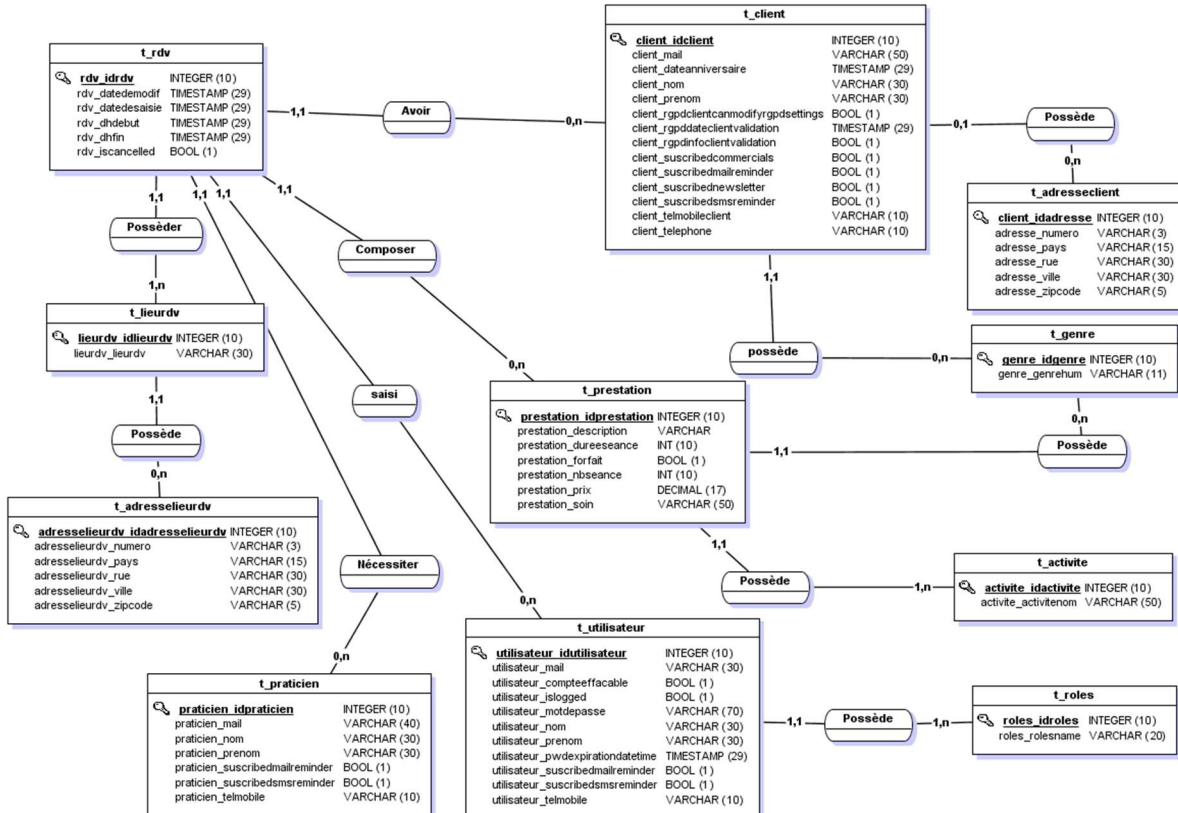


Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

9. Le Model Conceptuel de données MCD

L'objectif du MCD est de formaliser les données à utiliser dans le système d'information pour obtenir une représentation des données humainement compréhensible. Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur les concepts « entité-association ».



Le MCD me permet de visualiser les données et leur organisation via des relations.

Ci-dessus, le modèle de conception de mon application. Je vais décrire quelques relations avec des entités, en prenant quelques exemples :

- Relation entre « Client » et « genre », j'ai conçu cette représentation de la façon suivante : Un client possède un genre, j'entends par cette relation qu'un client ne peut posséder qu'un seul genre ce qui va me donner une cardinalité de 1,1. Dans le sens inverse un « Genre » possède un ou plusieurs « Client », ce qui nous donne une cardinalité de 1,N.
- La relation entre « Rdv » et « LieuRdv », j'ai souhaité que qu'un « Rdv » ne Mentionne qu'un seul « LieuRdv » donnant une cardinalité de 1,1 et pour la relation inverse j'ai souhaité qu'un « LieuRdv » soit mentionné dans plusieurs « Rdv » ce qui fait une cardinalité de 1,N.

10. Le Modèle Logique des Données MLD

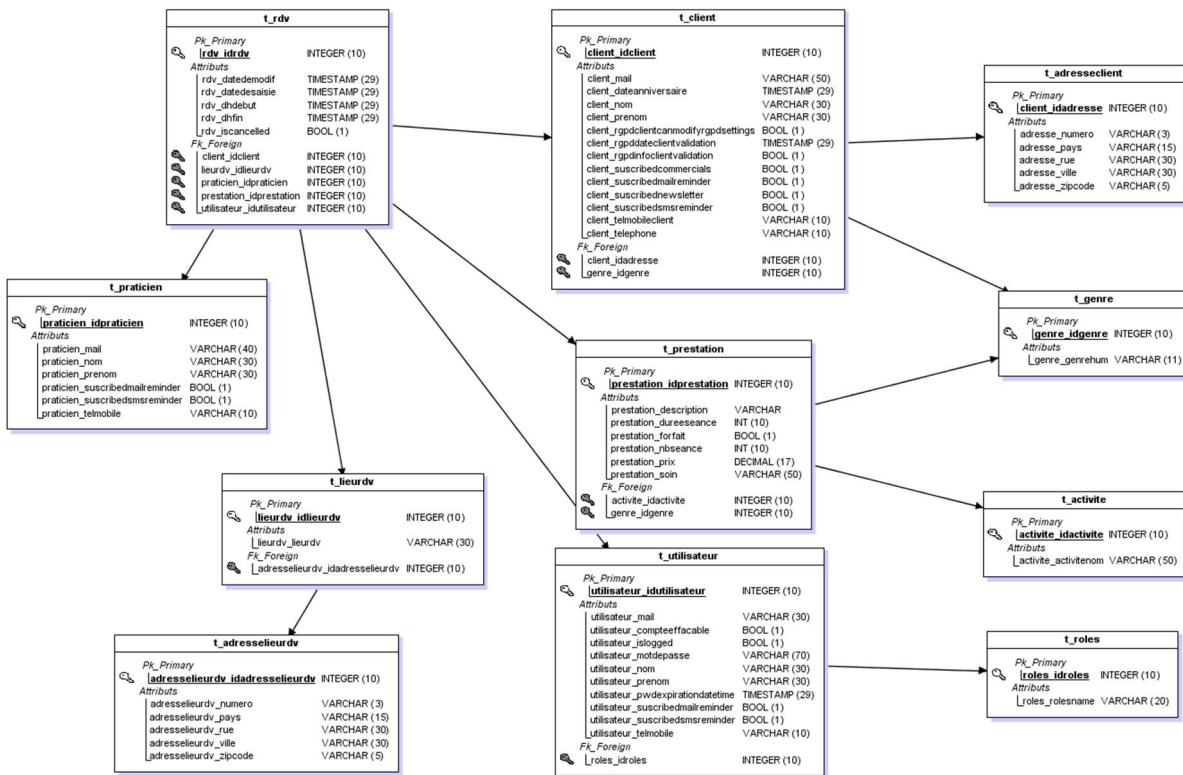
A partir du MCD je procède à sa conversion pour obtenir un Model Logique des Données (MLD). La conversion va permettre d'obtenir une structure de base de données exploitable en transformant les entités en tables et les associations cardinales en relations. Le processus de conversion peut créer des



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

tables de jointures et / ou des tables jointes via le principe de clé primaire (PK Primary key) importée dans une autre table sous la forme de clé étrangère (FK Foreign Key).



Le MLD représente ma base de donnée relationnelle

Après conversion, je retrouve mes tables devenues relationnelles via la mise en place de clés primaires et des jointures via l'ajout de clés étrangères. On y retrouve aussi les attributs, les types, les contraintes, ...

Dans la modélisation de mes données, je n'ai pas eu recourt à des associations pour créer des tables de jointures. Maintenant, via mon MLD, je suis en mesure de concevoir mes couches DAO liées aux objets métier.

Lors de la rédaction de ma soutenance, j'ai eu un petit souci avec les captures et la sauvegarde de mon activité de modélisation de ma base de données. Pour pallier à ce problème, je me suis appuyé sur ma base de données construite lors du développement des mes DAO / Objets métiers à partir de mon MLD. J'ai réalisé une action de « reverse engineering » sur ma base de données via mon outil de modélisation Merise. Le logiciel « Jmerise » m'a permis de me connecter à ma base de données et de récupérer la structure de la base et de pouvoir recréer un MCD et MLD. Il faut savoir que depuis le début de mon projet, ma base de données a fait l'objet de quelques évolutions et va sûrement encore évoluer dans le futur avec des nouveaux user stories à ajouter dans le BackLog.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

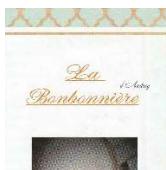
Maquettage

J'ai réalisé le wireframe (maquette fonctionnelle) de l'application. Le maquettage consiste concrètement en un croquis, un collage papier ou un schéma numérique. Via ce principe, j'ai présenté le design, les fonctionnalités de l'application à ma tutrice. J'ai fait un premier maquettage sans informations de charte graphique et de couleur, et cela a donné le résultat ci-après :



Premier essai de maquettage

Dès le début de la présentation de la maquette, ma tutrice m'a informé que la navigation horizontale dans le header et la couleur ne lui convenaient pas. Alors j'ai utilisé sa brochure tarifaire, qui pour l'instant représente sa charte graphique.



La plaquette me fournit une palette de 3 couleurs le vert, du doré et du blanc. Ainsi qu'une police de titre.

Je me suis appuyé sur différents éléments pour réaliser le nouveau maquettage :

1. Le nom de l'application « OpusBeauté »

J'ai proposé de trouver un nom pour l'application afin d'intituler le projet, et surtout cela me semble être un début de personnalisation. Ma tutrice a très rapidement proposé « OpusBeauté » et je lui ai demandé la signification de celui-ci. Audrey m'a expliqué que c'était un rappel entre son passé de musicienne avec son activité professionnelle à ce jour. Cela permet au projet d'avoir un nom d'application personnalisé et en relation avec son activité dans l'esthétisme.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

2. Le logo

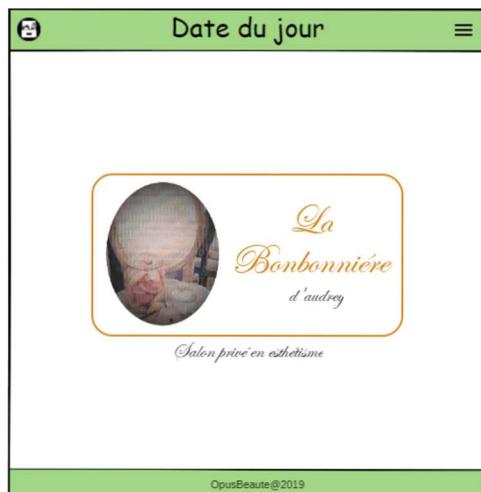
J'ai réalisé un logo pour le menu avec un outil vectoriel afin d'en générer un étirable et léger.



Pour créer le logo de l'application, je me suis inspiré d'un modèle existant sur un site payant spécialisé dans les photos et logos professionnels. Le logo est composé de trois matériaux que l'on trouve dans l'esthétique (un pinceau à maquillage, un miroir et une mallette).

3. Les maquettes

Pour la création du nouveau maquettage je vais utiliser la charte graphique à partir des couleurs de la brochure tarifaire. La barre de navigation dans le haut de la page ne lui plaisait pas alors j'ai proposé de passer par un menu coulissant sur la gauche, et de laisser la barre du haut uniquement pour les boutons de menus / login, ainsi que la date au milieu.



La landing page



Le menu bouton de login



Page Authentification

Page Home après Login



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Avec la pratique du maquettage, je suis en mesure de proposer un design (charte de couleurs, disposition des rubriques, ...). Ma tutrice a accepté cette version beaucoup plus en harmonie avec l'identité visuelle de son activité professionnelle. Et d'un point de vue fonctionnel, on distingue clairement la disposition des actions de navigation pour se connecter, et parcourir les rubriques métier.

4. Les rubriques métier :

Voici l'énumération de la liste des rubriques métiers disponibles depuis la barre de menu à gauche de la page web. Le menu coulisse depuis le bord afin d'afficher les rubriques.

- Rdv : donne accès à la création et liste / recherche de rendez-vous.
- Client : permet d'accéder à la création et liste / recherche des clients.
- Prestation : permet d'accéder à la création et liste / recherche des prestations.
- Praticien : permet d'accéder à la création et liste / recherche des praticiens.
- Utilisateur : permet d'accéder à la création et liste / recherche des utilisateurs.
- Activité : permet d'accéder à la création et liste / recherche des activités.

LE DEVELOPPEMENT

Sur le point d'initier le développement, il me reste à définir le temps moyen d'un Sprint. Etant au début d'un tout premier projet en autonomie, je vais fixer le temps d'un Sprint à quatre semaines pour le réaliser. Mon Sprint Zéro est terminé j'ai validé les étapes préliminaires au lancement du développement :

- Priorisation des User Stories (Backlog Produit).
- Définition du DONE.
- Durée moyenne d'un sprint.
- Le poste de travail avec les outils.
- L'équipe (moi).

1. Déroulement du développement

Je vais suivre le processus comme réalisé en formation. J'en profite pour revenir sur la Définition du Done et rappeler que le sprint est terminé quand celui-ci a parcouru toutes les étapes (de « analyse » puis « développement » et pour finir la case « test »). Si un sprint est terminé avant les 4 semaines, j'en commence un nouveau. Dans le cas d'un retard, je fais le point, je communique pour repousser la présentation et le présenter dès que possible. Le sprint pris en charge fait l'objet d'un découpage pour créer une multitude de tâches afin de détailler son anatomie (les points techniquement complexes, faciles ...). Le découpage de sprint donne naissance au 'Backlog de sprint', celui-ci est une liste détaillée des composants qui constituent la User Story. Le Backlog de sprint sert à répartir / organiser les tâches de développements au sein de l'équipe.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

2. Le cycle de développement :

1. Développement des classes d'objet métier et si besoin des entités.
2. Développement si besoin de la classe DAO en lien avec l'objet métier/entité développée en point 1 afin de réaliser les opérations du CRUD à travers les méthodes de la classe DAO.
3. Développer la classe Services qui intégrera les règles métier en lien avec la User Story.
4. Développer la classe WS afin d'exposer les WebServices REST via URL.
5. Chaque classe intégrera la gestion des exceptions afin de ne pas bloquer l'exécution du programme lors d'un incident et surtout de subtiliser l'erreur pour y glisser et remonter mes propres messages d'erreur.
6. Réaliser les tests des WebServices avec le logiciel Postman.
7. Développer la fonctionnalité dans l'IHM pour consommer les WebServices exposés par le middleware.

3. La classes Objets métiers

Je crée ma classe objet en fonction de la User Story, qui va s'appuyer sur un mécanisme de gestion de persistance afin de mapper mes objets dans la base de données. Le mécanisme couramment utilisé en JavaEE est l'api JPA (Java Persistance API) qui permet de faire ce mapping. JPA est un ORM (Object-Relationnal Mapping) et « Hibernate » en tire parti. A l'aide d'annotations propres à la persistance, il sera possible de transformer mon objet métier en entité. Par cette action, il sera possible de persister cette entité dans la base de données PostgreSQL sur du long terme car une entité a une courte durée de vie. J'ai paramétré mon fichier de configuration de persistance pour qu'à chaque démarrage du serveur d'application la base de donnée est recréée avec les tables correspondantes. Avec le fichier de persistance, j'importe automatiquement un fichier pour insérer un jeu de données pour peupler ma base. Cette option devra être changée pour la mise en production du logiciel.

Pour devenir une entité, l'objet métier doit répondre à des prérequis :

- La classe doit avoir un constructeur vide, des variables membre privé accessibles via des méthodes dites accesseurs/mutateurs, equals() et hashCode(), ne peut être final et doit être sérialisable, ces contraintes relèvent de la convention javaBean JSRxxx.
- Il faut définir une clé primaire.
- Il faut définir une ou des colonnes ;
- Je surcharge la méthode `.toString()` héritée de la classe Objet pour maîtriser le contenu des variables.
- J'ajoute l'annotation `@XmlRootElement` pour la transformation de l'objet en XML/JSON.

Voici en code :

<pre>38 @XmlRootElement 39 @Entity 40 @Table(name = "T_RDV") 41 public class Rdv implements Serializable { 42 43 private static final long serialVersionUID = ... 44 }</pre>	<p><code>@Entity</code> est l'annotation principale pour informer que l'on souhaite transfromer l'objet en entité. <code>@Table</code> sert à nommer la table « T_RDV ».</p>
<pre> @Id @SequenceGenerator(name="seq",sequenceName="RDV_SEQ", initialValue = 30) @GeneratedValue(strategy = GenerationType.SEQUENCE, generator="seq") @Column(name = "RDV_IDRDV", nullable = false, length = 8) public Integer getIdRdv() { return idRdv; }</pre>	<p>L'annotation <code>@Id</code> indique que la colonne « RDV_IDRDV » est la clé primaire. Les annotations <code>@SequenceGenerator</code> et <code>@GeneratedValue</code> servent à générer une séquence dans la base de donnée pour une incrementation automatique et à partir de la position 30.</p>



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

```
@Override  
public String toString() {  
    return new ToStringBuilder(this, ToStringStyle.JSON_STYLE)  
        .append("idRdv", this.idRdv)  
        .append("dateHeureDebut", this.dateHeureDebut)  
        .append("dateHeureFin", this.dateHeureFin)  
        .append("dateDeSaisie", this.dateDeSaisie)  
        .append("dateDeModif", this.dateDeModif)  
        .append(prestation)  
        .append(praticien)  
        .append(client)  
        .append(lieuRdv)  
        .append(isCancelled)  
        .build();  
}
```

Je surcharge ma méthode `.toString()` avec mes variables objet.

```
@Override  
public int hashCode() {  
    final int prime = 31;  
    int result = 1;  
    result = prime * result + ((idRdv == null) ? 0 : idRdv.hashCode());  
    result = prime * result + ((dateHeureDebut == null) ? 0 : dateHeureDebut.hashCode());  
    result = prime * result + ((dateHeureFin == null) ? 0 : dateHeureFin.hashCode());  
    result = prime * result + ((dateDeSaisie == null) ? 0 : dateDeSaisie.hashCode());  
    result = prime * result + ((dateDeModif == null) ? 0 : dateDeModif.hashCode());  
    result = prime * result + ((prestation == null) ? 0 : prestation.hashCode());  
    result = prime * result + ((praticien == null) ? 0 : praticien.hashCode());  
    result = prime * result + ((client == null) ? 0 : client.hashCode());  
    result = prime * result + ((lieuRdv == null) ? 0 : lieuRdv.hashCode());  
    result = prime * result + ((isCancelled == null) ? 0 : isCancelled.hashCode());  
    return result;  
}
```

Je surcharge ma méthode `.hashCode()` héritée de la classe Objet afin de générer un hash de l'objet selon la valeurs de ses variables.

```
@Override  
public boolean equals(Object candidat) {  
    if (candidat == this)  
        return true;  
    if (candidat == null)  
        return false;  
    if (!(candidat instanceof Rdv))  
        return false;  
    final Rdv autre = (Rdv) candidat;  
    return new EqualsBuilder()  
        .append(this.idRdv, autre.idRdv)  
        .append(this.dateHeureDebut, autre.dateHeureDebut)  
        .append(this.dateDeSaisie, autre.dateDeSaisie)  
        .append(this.dateHeureFin, autre.dateHeureFin)  
        .append(this.dateDeModif, autre.dateDeModif)  
        .append(this.prestation, autre.prestation)  
        .append(this.praticien, autre.praticien)  
        .append(this.lieuRdv, autre.lieuRdv)  
        .append(this.client, autre.client)  
        .append(this.isCancelled, autre.isCancelled)  
        .build();  
}
```

Je surcharge ma méthode `.equals()` pour que la comparaison d'objet s'appuie sur les variables .

Ma classe Objet métier est prête pour devenir des entités.

4. La classes DAO

A présent, j'ai besoin de créer une classe chargée de dialoguer avec ma base de données. Une classe DAO (Data Access Object) est un Design Pattern chargé de réaliser une abstraction entre le l'Objet Métier (entité) et les opérations d'accès à la base de données. L'Api JPA (Java Persistance API) se charge de réaliser la séparation des accès aux données (BDD) et objets métiers (POJO). JPA permet de réaliser les actions de créer, lecture, mise à jour et effacer => le CRUD (Create, Read, Update, Delete) à partir de JavaBean(EJB). Les EJB (Enterprise Java Bean) proposent des Beans de type entités pour assurer la persistance des objets.

Pour fonctionner, l'API JPA repose sur :

- Un fichier Persistance en xml pour déclarer l'unité de persistance nécessaire au contexte de Persistance.
- Un contexte de Persistance.
- Un gestionnaire de persistance (EntityManager) qui assure la gestion des entités persistantes.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

```
<persistence-unit name="opusBeautePU" transaction-type="JTA">
    <!-- persistence-unit name="dobyPU" transaction-type="RESOURCE_LOCAL" -->
    <!-- Le choix de l'outil JPA à utiliser -->
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
```

Déclaration de l'unité de persistance depuis le fichier Persistance.xml ;

```
@Stateless
@Transactional
public class RdvDao {
```

L'annotation @Stateless pour indiquer que c'est un Bean de session sans état.

```
@PersistenceContext(unitName = "opusBeautePU")
private EntityManager em;
```

L'annotation de persistance contexte nécessaire au gestionnaire d'entité.

Méthode pour récupérer un rendez-vous :

```
/**
 * Recupere un Rdv par son Id
 *
 * @param idRdv Integer
 * @return rdv Rdv
 * @throws RdvInexistantException Exception
 */
public Rdv obtenirRdv(final Integer idRdv) throws RdvInexistantException {
    logger.info("RdvDao log : Demande a la bdd le Rdv id : " + idRdv);
    Rdv rdv = null;
    rdv = em.find(Rdv.class, idRdv);

    if (Objects.isNull(rdv)) {
        logger.error("RdvDao log : Le rdv : " + idRdv + " demande est introuvable");
        throw new RdvInexistantException("RdvDao Exception : L'Id : " + idRdv + " est introuvable dans la base");
    }

    logger.info("RdvDao log : Rdv " + idRdv + " trouve, envoie de l objet Rdv a RdvService");
    return rdv;
}
```

Méthode de récupération d'une liste de Rendez-vous via l'API Criteria :

```
/**
 * Recuperer la liste de Rdv
 *
 * @return List
 * @throws DaoException Exception si pas d eliste trouvée
 */
public List<Rdv> obtenirListeRdv() throws DaoException {
    final CriteriaBuilder cb = em.getCriteriaBuilder();
    final CriteriaQuery<Rdv> reqCriteria = cb.createQuery(Rdv.class);
    reqCriteria.from(Rdv.class);
    final TypedQuery<Rdv> requete = em.createQuery(reqCriteria);
    List<Rdv> listeRdvs;

    try {
        logger.info("RdvDao log : Demande a la bdd la liste des Rdv");
        listeRdvs = requete.getResultList();

    } catch (Exception message) {
        logger.error("RdvDao Exception : Probleme de la bdd pas de liste de rdv trouvée.");
        throw new DaoException("RdvDao Exception : Probleme de la bdd pas de liste de rdv trouvée.");
    }
    logger.info("RdvDao log : Envoi de la liste de Rdv");
    return listeRdvs;
}
```



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Une méthode multi critères en SQL :

```
/**  
 * Recupere la liste de rdv par Date et  
 * Par Client  
 *  
 * @param dateJJ String  
 * @param IdClient Integer  
 * @return List  
 * @throws DaoException Exception  
 */  
public List<Rdv> obtenirListeRdvDuJJParClient(final String dateJJ, final Integer IdClient) throws DaoException {  
    try {  
        Logger.info("RdvDao log : Demande a la Bdd de la liste des Rdv's selon les criteres "  
            + "de date selectionnee et par Client");  
  
        String requete = "SELECT c FROM Rdv c" + " WHERE rdv_dhdebut >= '" + dateJJ + " 00:00:00'"  
            + " AND rdv_dhdebut <= '" + dateJJ + " 23:59:00'"  
            + " AND RDV_IDCLIENT_fk = " + IdClient  
            + " ORDER BY rdv_dhdebut";  
  
        TypedQuery<Rdv> query = em.createQuery(requete, Rdv.class);  
        List<Rdv> listeRdvDateJClient = query.getResultList();  
        Logger.info("RdvDao log : Transmission de la Liste des Rdv's par date selectionnee et par client.");  
        return listeRdvDateJClient;  
    } catch (Exception message) {  
        Logger.error("RdvDao log : probleme sur le format de la date ou id_Client.");  
        throw new DaoException("RdvDao Exception : Date non existante dans calendrier.");  
    }  
}
```

Ici, un exemple de requête en JPQL :

```
/**  
 * Recupere la liste de rdv par client  
 *  
 * @param idClient Integer  
 * @return List  
 * @throws DaoException Exception  
 */  
public List<Rdv> obtenirListeRdvParClient(final Integer idClient) throws DaoException{  
    try {  
        Logger.info("RdvDao log : Demande a la Bdd de la liste des Rdv's selon les criteres "  
            + "de date selectionnee et par Client");  
        String requete = "SELECT c FROM Rdv c" + " WHERE client_idClient = " + idClient;  
  
        TypedQuery<Rdv> query = em.createQuery(requete, Rdv.class);  
        List<Rdv> listeRdvClient = query.getResultList();  
        Logger.info("RdvDao log : Transmission de la Liste des Rdv's par date selectionnee et par Client.");  
        return listeRdvClient;  
    } catch (Exception message) {  
        Logger.error("RdvDao log : probleme sur le format de la date ou id_Client.");  
        throw new DaoException("RdvDao Exception : Date non existante dans calendrier.");  
    }  
}
```

Dans ces portions de code on constate que l'on peut requêter la base de données selon différentes techniques. Les requêtes peuvent être natives en SQL, en JPQL / Critéria. Les classes DAO définies dans mon projet sont sans état (Stateless). Lorsque le serveur d'applications extrait un bean session sans état du pool d'objets, il sait qu'il peut être utilisé pour satisfaire toute requête, car il n'a aucun état conversationnel. La classe DAO doit aussi être transactionnelle pour pouvoir réaliser le *rollback()* en cas de problème de la requête. Cela permettra de garantir que la base de données relationnelle soit ACID :

Il s'agit des propriétés nécessaires pour garantir une transaction informatique. Très utiles dans les bases de données lors d'opérations critiques, les propriétés ACID se matérialisent par les fonctions de transactions. Toutes les bases de données ne mettent pas en place ces propriétés.

- **Atomicité** : Tout ou rien. Soit l'opération se fait en entier, soit elle ne se fait pas du tout. La notion d'atomicité sous-entend la possibilité de défaire une opération avortée.
- **Cohérence** : L'opération doit assurer que la base de données sera dans un état valide après l'opération.
- **Isolation** : L'opération doit se faire en toute autonomie sans dépendance à une autre opération.
- **Durabilité** : En cas de problème important (coupure d'électricité), les modifications apportées sont bien enregistrées.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

5. Les classes Services

A présent, je créé la classe Services qui est au cœur du middleware. Elle est en charge du traitement des données en respectant les règles métier. Ma classe Services est entre les classes DAO et WebServices, et contient des méthodes de traitement. Les traitements sont chargés de réaliser sur les données différentes actions de modification, de validation, de comparaison, de vérification, ...

Quelques exemples de traitements métier :

- Vérifier que la donnée est de type string.
- Vérifier que la donnée n'est pas nulle ou vide.
- Formater l'intégralité la chaîne de caractères du Nom Client en majuscule.
- Formater la chaîne du Prénom Client avec la première lettre en majuscule puis le reste de la chaîne en minuscule.
- Vérifier que la fin du nouveau rendez-vous ne chevauche pas le début d'un existant le même jour.
- Vérifier que le mot de passe fourni est identique à celui inscrit dans la base de données.

Les données font donc l'objet de traitements pour obtenir des résultats attendus, mais aussi retourner des exceptions en cas de non-conformité ou de problème. Les classes Services dans le middleware permettent à l'IHM ou tout autre forme de client de s'affranchir en partie ou complètement de la responsabilité du traitement des données.

Les Classes Services sont, elles aussi, annotées pour être sans état et transactionnelles et rendra disponibles les méthodes pour une autre classe via le mécanisme d'injection de dépendance IoC (Inversion de Contrôle). Cette fonctionnalité est possible via les EJB (Entreprise Java Bean dans JavaEE) par l'annotation @EJB. L'injection permet de mettre à disposition les méthodes publiques d'une classe annotée Stateless (Bean de session sans état). La classe se verra être encapsulée pour être mise à disposition par le conteneur sans que le développeur n'ait à gérer le cycle de vie, la sécurité, la gestion de l'état, la gestion des transactions (par nature les EJB sont transactionnels)... Le déploiement des EJB est réalisé dès le démarrage de l'application depuis le serveur d'application, les EJB exécutés dans des conteneurs, sont gérés, et au besoin les méthodes seront disponibles via le pool. Les patrons Inversion de Control (IOC) et Pool (Pool) font partie d'une vingtaine de design pattern utilisés dans le développement objet, les patrons sont les meilleures réponses à des problèmes récurrents.

```
@Stateless
public class ClientService {
    static final Logger logger = LogManager.getLogger(ClientService.class);

    @EJB
    private ClientDao clientdao;
    @EJB
    private AdresseClientDao adressedadao;
    @EJB
    private GenreDao genredao;
```

Voici l'annotation @Stateless pour transformer la classe en Bean de session sans état afin d'être injectée dans une autre classe.

Plus bas, les annotations @EJB pour injecter les classe annotée @Stateless.

Via ce mécanisme les méthodes des classes injectées seront disponibles dans cette classe.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

6. Les classes WebServices

Ce sont les dernières classes du middleware de mon projet, les classes WebServices ont pour rôle de mettre à disposition des Services Web via des méthodes. Les méthodes peuvent faire transiter des informations au format clé/valeur (JSON) ou XML.

Les services sont consommés ou produits avec toute sorte de client (IHM) via le protocole HTTP (Hyper Text Transfert Protocole), qui met à disposition des verbes GET, POST, PUT, DELETE, ...

C'est encore les annotations qui permettent de réaliser les fonctionnalités dont j'ai besoin. Il faut annoter les classes et / ou les méthodes publiques avec (@Path) pour créer le / les chemins URL (Uniform Ressource Locator) afin accéder aux ressources. J'ai également utilisé les annotations associées aux verbes du protocole HTTP (@GET, @POST, @PUT, @DELETE). Cela requiert que le client ayant besoin d'accéder à la ressource doit faire la requête avec le même verbe HTTP.

Les ressources sont chargées de faire transiter les données selon différents types (application/json, text/plain, text/html, image/jpeg ...). Aussi, j'ai utilisé les annotations facultatives (@CONSUMES et / ou @PRODUCES), ces deux annotations permettent de déclarer les types MIME (identifiant de format de données sur internet).

J'ai utilisé certaines annotations mises à disposition par l'API java JAX-RS comme (@PathParam) afin d'associer un morceau de l'URL de requête à un champ ou un paramètre. (@QueryParam et @FormParam) permettent d'associer un paramètre de la requête à un champ ou un paramètre d'une méthode des classes web services.

```
@SecuApp
@Stateless
@Path("/prestation")
@DefineUserRole({"ALLOWALL"})
public class PrestationWs {
```

L'annotation @Path au niveau de la classe permet de mettre à disposition l'URL de la ressource principale.

```
@DefineUserRole({"ALLOWALL"})
@GET
@Path("/{idRdv: \d+}") // fonctionne bien 11/07
@Produces(MediaType.APPLICATION_JSON)
public Response chercheUnRdv(
    @PathParam("idRdv") final Integer idRdv) throws RdvInexistantException {
```

Ici l'annotation @GET précise le verbe HTTP à utiliser pour accéder à la méthode publique.
L'annotation @PATH sur la méthode permet de définir la suite de l'url pour accéder à la ressource.
Pour finir, @PATHPARAM sert à indiquer l'information à récupérer depuis l'URL.

Maintenant, la dernière partie du middleware est capable de jouer son rôle pour répondre aux différentes formes de clients qui ont le besoin de consommer les ressources misent à disposition. A chaque méthode de classes WebServices, je me suis servi de l'application POSTMAN pour réaliser des tests afin de m'assurer du bon fonctionnement de mes ressources.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Interface graphique du logiciel PostMan, je peux tester mes WebService et les sauvegarder pour les répéter au besoin.

Le WebServices est chargé de répondre au client et le protocole HTTP accompagne la réponse avec des codes d'état pour spécifier une des cinq catégories de réponse (informations, succès, redirection, erreur client et erreur serveur).

- 200 : succès de la requête ;
- 301 et 302 : redirection, respectivement permanente et temporaire ;
- 401 : utilisateur non authentifié ;
- 403 : accès refusé ;
- 404 : page non trouvée ;
- 500 et 503 : erreur serveur ;
- 504 : le serveur n'a pas répondu.

Cela permet d'informer le client sur le déroulement de la requête lors de la consommation du WebServices.

J'ai créé des classes d'exceptions pour les situations exceptionnelles. A chaque niveau de classes (Dao, Service et WebServcies), j'ai utilisé au besoin les clauses (Try) et (Catch) pour intercepter et prendre en charge un éventuel problème dans le déroulement du code. En cas de problème, une exception est jetée (throw), celle-ci remonte jusqu'au WebServices et mentionne un message d'erreur ainsi que l'état d'erreur grâce au Protocole HTTP. La gestion d'erreur permet à l'application de ne pas s'arrêter en cas de problème grâce aux exceptions gérées, et, du côté d'une IHM, la remontée d'exception permet d'être en mesure d'afficher un message d'information à destination de l'utilisateur.

7. Exécution automatique de méthodes

Mon middleware réalise les tâches d'envoi de mail et de sms automatiquement sans action humaine. Ces tâches sont exécutées à vingt heures tous les jours. Cette action automatisée est réalisée grâce au Service Timer de JavaEE.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

```
/**  
 * Envoie un Sms automatiquement  
 * chaque jours à 20H00  
 *  
 * @throws Exception  
 */  
@Schedule(dayOfWeek = "*", hour = "20", minute = "00")  
public void sendSmsClientScheduled() throws Exception {
```

L'annotation `@Schedule` indique dans ce cas d'exécuter la méthode tous les jours de la semaine à vingt heures.

8. Les tests

L'objectif de réaliser des tests est de vérifier que ce qui a été fabriqué est conforme à ce qui a été demandé. Il faut créer une classe de tests par classe à tester. Dans chaque classe de tests, il y a une méthode par méthode à tester. Pour chaque classe du logiciel, on va avoir sa sœur pour le test.

Définition : un cas de tests est un ensemble composé de trois objets.

- Un état (ou contexte) de départ ;
- Un état (ou contexte) d'arrivée ;
- Un oracle, c'est à dire un outil qui va prédire l'état d'arrivée en fonction de l'état de départ et comparer le résultat théorique et le résultat pratique.

Un cas de test peut donc s'appliquer à plusieurs méthodes, par exemple plusieurs classes implémentant la même interface.

L'ensemble des classes d'un projet doivent être encadrées par des tests. Dans le cadre de mon premier projet, j'ai procédé à une mise en place partielle de tests. Je me suis appuyé sur le Framework Junit pour faciliter la mise en place de ceux-ci. J'ai créé une classe de tests dans le répertoire test de l'arborescence de mon projet Java. Une annotation suffit pour exécuter une méthode à tester.

Junit propose un ensemble d'annotations (`@Test`, `@Before`, `@After...`) et de méthodes (`.assertEquals()`, `.assertFalse()`, `.assertTrue()`, `.assertNull()`, `.asssertNotnull()`) pour la réalisation de tests et la vérification d'un résultat attendu.

Exemple d'une méthode de tests :

```
@Test  
public void validFormatAdresseMail() {  
  
    String adresseMail = "test@test.com";  
  
    Assert.assertEquals("L'adresse mail doit valider : ", "test@test.com", adresseMail);  
    String ePattern = "[a-zA-Z0-9.!#$%&'*+=?^_-{}~-]+@[([0-9]{1,3}\\.){1,3}[0-9]{1,3}\\.){1,3}[0-9]{1,3}@[a-zA-Z]{2,3})$";  
    java.util.regex.Pattern p = java.util.regex.Pattern.compile(ePattern);  
    java.util.regex.Matcher m = p.matcher(adresseMail);  
    Assert.assertTrue("Le format doit revoyer true : ", m.matches());  
}
```

L'annotation `@Test` sert à indiquer que la méthode de classe est à exécuter pour faire un test. La méthode `.assertEquals()` permet de saisir la valeur de la donnée à tester, l'objectif est de vérifier par un test d'expression régulière la validité de la structure du mail. La méthode `.assertTrue()` sert à vérifier que le résultat du traitement correspond à (True) avec la valeur de départ.

L'enchainement de tests sur l'ensemble d'un projet permet d'éviter les risques régressions lors de l'ajout de nouvelles fonctionnalités (User stories).



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

```
Windows PowerShell
-----
T E S T S
-----
Running fr.labonbonniere.opusbeaute.CheckNbCharGenreUTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.188 sec - in fr.labonbonniere.opusbeaute.CheckNbCharGenreUTest
Running fr.labonbonniere.opusbeaute.CheckSpaceUTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec - in fr.labonbonniere.opusbeaute.CheckSpaceUTest
Running fr.labonbonniere.opusbeaute.SupprimeLesCaracteresSpeciauxUTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 sec - in fr.labonbonniere.opusbeaute.SupprimeLesCaracteresSpeciauxUTest
Running fr.labonbonniere.opusbeaute.ValidAdresseMailFormatUTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.016 sec - in fr.labonbonniere.opusbeaute.ValidAdresseMailFormatUTest

Results :

Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
```

Pour lancer l'exécution de tests unitaires, il suffit d'exécuter la commande avant la compilation avec la commande (maven) mvn test. La commande aura pour action de procéder à l'ensemble des tests de l'application et de donner le résultat de la batterie de tests réalisés avec succès et/ou échec. Les tests peuvent être réalisés automatiquement à la compilation du code via (maven) mvn clean install.

9. Test Driven Development (TDD) ou développement piloté par les tests

Pour ce premier projet, j'ai mobilisé les efforts pour assimiler les aspects de persistance, de Services et de WebService et d'IHM. C'est pour cela que je n'ai pas adopté cette technique de développement pilotée par l'écriture de tests avant le code métier. Je préfère en avoir la connaissance théorique et réservé l'apprentissage de cette méthode dans les prochains projets.

Pour l'instant, voici les préconisations pour le cycle de développement TDD en cinq étapes :

- Ecriture d'un premier test.
- Exécution du test et vérifier qu'il échoue (car le code qu'il teste n'a pas encore été implémenté).
- Ecriture de l'implémentation pour faire passer le test (il existe différentes manières de corriger ce code).
- Exécuter les tests afin de contrôler que ceux-ci passent et que l'implémentation respectera les règles fonctionnelles des tests unitaires.
- Remaniement (Refactor) du code afin d'en améliorer la qualité mais en conservant les mêmes fonctionnalités.

10. Développement IHM (l'Interface Homme Machine)

Voici la dernière étape du développement d'une User Story : la partie orientée vers l'utilisateur afin qu'il puisse interagir avec le logiciel. J'ai choisi de développer l'IHM en Angular version 4.

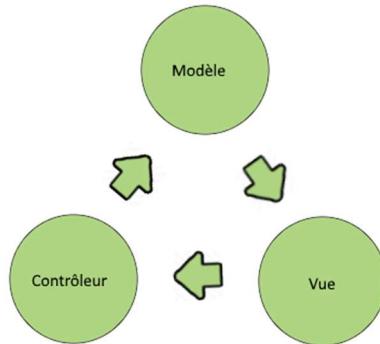
Le Framework gratuit proposé par Google a été utilisé pendant un des modules de formation pour nous faire réaliser une première interface graphique. J'ai continué d'utiliser ce Framework, car c'est celui avec lequel j'ai eu ma première initiation. J'ai apprécié que les fichiers avec leur rôle soient séparés : pour un débutant, l'appréhension s'en retrouve facilitée. Angular est basé sur du Typscript de Microsoft afin d'offrir un typage statique du code contrairement à javascript qui est basé sur un typage dynamique. Maintenant j'ai l'habitude de typer mon code et cela permet de se prémunir d'un changement de type d'une variable par erreur. J'ai migré mon projet Angular de la version 4 pour arriver à la version 6 puis la version 7 via les outils et procédure à disposition sur le net par Google.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Angular répond au design Pattern MVC (Model-Vue-Contrôleur). il s'agit d'un patron de conception qui permet d'avoir une stricte séparation entre la vue (ce que l'utilisateur voit), le modèle (les données) et le contrôleur (les actions possibles).



MVC est utilisé dans de nombreux langages de programmation pour apporter une structure/architecture à une application. C'est le nom donné à une manière d'organiser son code et une façon d'appliquer le principe de séparation des responsabilités, en l'occurrence celles du traitement de l'information et de sa mise en forme.

En voici les composants :

- Modèle :
Structure de données représentant une entité de l'application, généralement transmise en JSON depuis l'IHM vers le middleware et inversement.
- Vue :
La vue est rendue par la combinaison des fichiers HTML/SCSS. Les données sont issues du Modèle pour mettre la Vue à jour et afficher les bonnes informations dans le fichier HTML.
- Contrôleur :
Comme son nom l'indique, cette couche contrôle des données. Les contrôleurs permettent de communiquer avec la Vue. C'est via le contrôleur que je peux mettre les données à jour de façon dynamique entre le serveur et le client. Le contrôleur s'appuie sur une classe service qui me sert à récupérer ou envoyer des données.

Google Angular met à disposition un Framework puissant pour réaliser l'interface graphique avec des notions de directives. Les directives structurelles comme *ngIf, *ngFor, sont utilisées lorsque l'on souhaite modifier ou transformer le DOM (Document Object Model).

Google propose Material, un module d'intégration qui permet d'obtenir facilement une interface responsive harmonieuse et unie dans le style 'flat' de Google. C'est une série d'objets quasi prêts à l'emploi comme des 'Inputs' pour saisir du texte, des listes déroulantes, des sliders, des boutons, des Checkbox, ...

J'ai également utilisé Flexbox pour faciliter la répartition des objets afin de remplir au mieux l'espace disponible, et de s'adapter à tous les terminaux et toutes les tailles d'écrans. Un container flexible permet aux items de s'étendre pour occuper la place disponible ou au contraire les réduit pour leur éviter de déborder. A mon prochain projet, je m'appuierai sur le système de 'grid' afin de comparer les deux approches.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

L'équipe Angular a mis en place un utilitaire pour assister le développement. Il se nomme Angular Cli. Ce dernier met à disposition des commandes pour automatiser la création de composants, des services, ... de tester et de construire l'application afin de gagner en qualité et en temps sur les tâches répétitives.

Ces commandes permettent donc de créer des composants pour générer l'environnement dans lequel j'intègre mes objets (inputs, modal, zone de texte, boutons,...) depuis Material design. Le cumul de ces objets dans une page constitue l'interface permettant l'interaction avec l'utilisateur. Ces composants sont le fruit de la combinaison de plusieurs fichiers dont le rôle leur est propre.

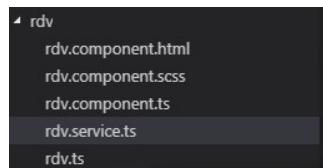
- Classe Objet représente la définition du même objet dans le middleware.

```
rdv.ts      x
1  import { Prestation } from "../prestation/prestation";
2  import { Praticien } from "../praticien/praticien";
3  import { Client } from "../client/client";
4  import { Utilisateur } from "../utilisateur/utilisateur";
5  import { LieuRdv } from "../lieuRdv/lieuRdv";
6
7  export class Rdv {
8
9      public idRdv: number;
10     public dateHeureDebut: number;
11     public dateDeSaisie: number;
12     public dateHeureFin: number;
13     public dateDeModif: number;
14     public prestation: Prestation;
15     public praticien: Praticien;
16     public client: Client;
17     public lieuRdv: LieuRdv;
18     public utilisateur: Utilisateur;
19     public isCancelled: boolean;
20 }
```

- Classe contrôleur contient des méthodes diverses pour la logique de la vue, pour traiter les données et de récupérer / créer / mettre à jour des données, traitements métier si besoin.

```
/*
 * Active / Desactive Bouton Sauvegarder
 */
public toggleSaveButtonStatus() {
    this.logger.info("RdvAddComponent log : Methode activee");
    if (this.save_button_state == true) {
        this.save_button_state = !this.save_button_state;
    }
}
```

- Fichiers HTML + CSS (Cascading Style Sheets) : l'association de ces deux fichiers permet de mettre en forme la vue avec les données selon le maquettage réalisé en amont dans le projet avec le StakeHolder.



- Classe Service possède des méthodes qui permettent à la demande du contrôleur de récupérer / ajouter / mettre à jour / effacer des données via les WebServices REST du Middleware en JSON.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

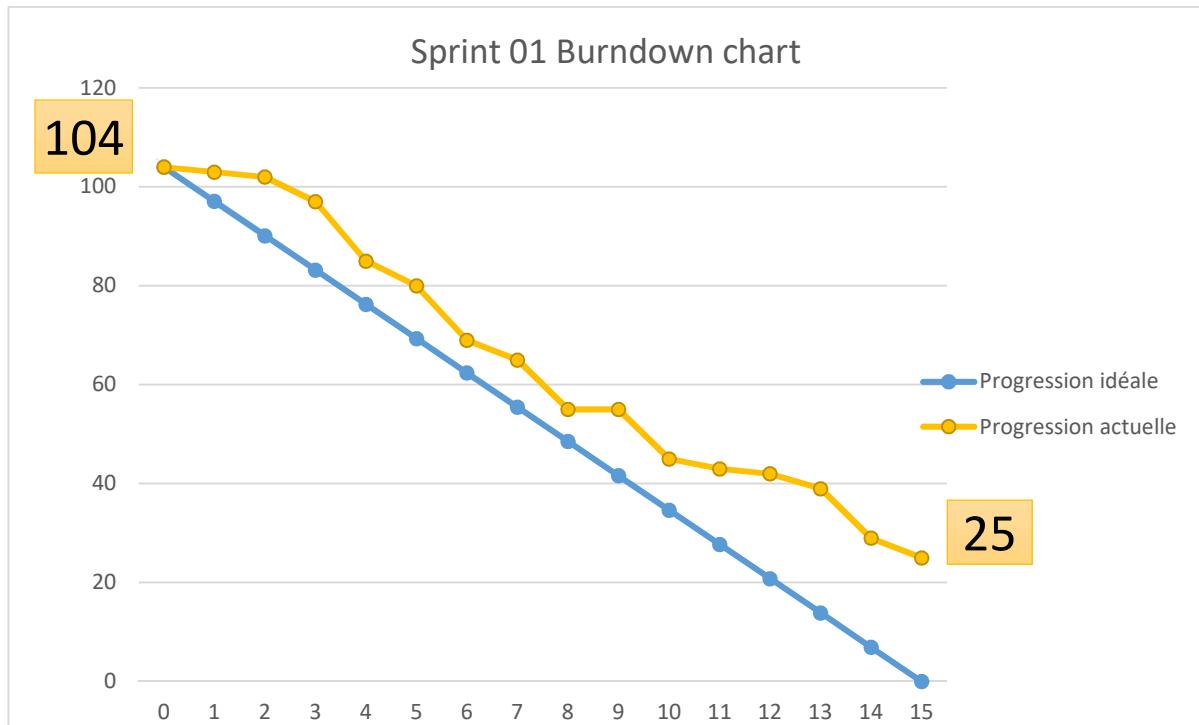
```
/**  
 * recupere la liste totale de rdv  
 */  
public Observable<Rdv[]> getRdvListTotale() {  
  
    this.logger.info("RdvService Log : Recupere la liste totale de Rdv");  
  
    return this.httpClient  
        .get<Rdv[]>(this.url + '/liste')  
        .pipe(map(res => res));  
}
```

Le sprint est terminé après avoir testé la nouvelle fonctionnalité développée. Celle-ci doit être opérationnelle depuis le middleware jusqu'à l'IHM. Pour être validée à 'Done', la nouvelle 'feature' doit correspondre à la User Story et à la maquette.

11. Burndown Chart de Sprint

La réalisation du Burndown Chart de sprint s'appuie sur le rituel de la réunion quotidienne (la mélée). La mélée va permettre et de faire le point sur l'état d'avancement des tâches prises en charge en mentionnant en toute transparence ce qui a été fait, les situations problématiques et ce qui va être pris en charge.

A partir de ces échanges, le Burndown Chart de sprint prend de la consistance. Le graphique a l'avantage d'être visuellement éloquent et permet d'ajuster la dynamique de la réalisation des tâches suivantes du Backlog de sprint.



BurnDown chart du premier sprint met en évidence le retard de l'itération par rapport aux estimations.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

J'ai réalisé dans le tableur Excel les courbes d'indications mettant en relation la progression idéale (estimée) et la progression réelle. Je me suis appuyé sur différentes informations dont l'évaluation du temps théorique pour réaliser le sprint, le temps réel de développement, le nombre de développeurs, le nombre d'heures travaillées par jour. La courbe bleue (progression estimée idéale) indique la progression entre le temps alloué pour les différentes tâches du sprint et le temps alloué à la complexité de la User Story. La courbe de la progression réelle est le résultat du temps passé pour la réalisation de la tâche et le nombre de jours alloués au développement de la User Story.

Le graphique démontre visuellement que mon premier sprint est à peu près 20% en retard par rapport à l'estimation initiale du temps de développement prévu. Etant seul dans ce projet, je suis à la fois Scrum Master et développeur. En tant que développeur, je suis en mesure d'identifier les difficultés rencontrées et de les transmettre lors de la mêlée quotidienne. En tant que Scrum Master, je suis sensibilisé aux problématiques énoncées et j'essaye de mettre en place des solutions facilitatrices pour améliorer le déroulement des prochaines tâches.

Le premier point justifiant le retard est situé sur l'expérience lors de l'évaluation de la technicité des User Stories. En tant que développeur, j'ai surévalué ma capacité à traiter les tâches dans le temps donné pour réaliser la User Story. Je suis dans un contexte d'apprentissage et je vais en profiter pour me mettre dans la peau du Scrum Master et de nouveau évaluer la User Story suivante. Il faut prendre en compte le fait que des points d'expérience ont été gagnés lors du premier sprint et ce capital d'expérience augmentera au fil des sprints suivants.

Les informations saisies dans mon fichier Excel pour tracer mon BurnDown Chart de Sprint me permettent d'analyser plus finement le tracé du premier sprint. Je constate que j'ai eu besoin de plus de temps pour le développement de l'IHM. Le temps dédié à cette partie a été conséquent car j'ai eu besoin de créer la structure de base : le header, la page principale (body), le pied de page, le menu (slide) des rubriques et les boutons.

Je compte sur l'amélioration de mon expérience lors des sprints suivants afin de me rapprocher de la courbe de progression estimée.

12. L'intégration continue

L'intégration continue est une pratique consistant à intégrer de façon continue les changements apportés à un projet. Chaque intégration est vérifiée et testée par un Build automatisé, afin de détecter les erreurs d'intégration le plus rapidement possible. Un autre avantage de cette pratique est qu'elle permet d'améliorer la qualité du code. Les développeurs, n'ayant plus à se soucier des problèmes, peuvent davantage se concentrer sur le bon fonctionnement du code. La qualité du produit fini s'en trouve améliorée.

Pour utiliser l'intégration continue, j'ai installé le logiciel Jenkins en local sur mon poste de développement. J'y ai associé mon compte Github pour qu'à chaque Commit de mon code Java du middleware ou en TypeScript de mon IHM se déclenche un Hook. Ceci afin de transférer le nouveau code source depuis Github vers mon Jenkins. A réception du code, Jenkins lance les tests et compile le projet. Si le code ne comporte pas d'erreur en exécutant les tests unitaires, la Build est réalisée avec succès sinon c'est en échec et il faut modifier le code récemment ajouté. Le résultat de la compilation se manifeste par des icônes.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

S	W	Name	Last Success	Last Failure	Last Duration
●	●	OpusBeauté_Ihm	5 min 10 sec - #25	2 mo 21 days - #3	4 min 27 sec
●	●	OpusBeauté_Mw	2 days 15 hr - #21	2 mo 21 days - #3	1 min 21 sec

Capture de l'interface du logiciel d'intégration continue du projet OpusBeauté.

13. Le déploiement continu

Jenkins permet aussi de mettre en place le déploiement continu (Continuous Deployment) et la livraison continue (Continuous Delivery). Pour mon premier projet, je n'ai pas été jusqu'à la mise en place de ce niveau de services, mais dans le futur proche, cela fera l'objet d'une intégration afin de mettre en pratique la théorie autour de ces deux techniques.

Le déploiement continu et la livraison continue sont des pratiques directement liées à l'intégration continue. La livraison continue sert à automatiser le processus de « relaxe » des changements apportés au logiciel et de les mettre à disposition.

Le déploiement continu va encore plus loin, et consiste à livrer chaque changement apporté au logiciel. Dans ce cas de figure, il n'y a pas d'intervention humaine. Les seuls changements qui ne sont pas déployés sont ceux qui échouent à un test. Cette pratique permet d'accélérer la boucle de feedback.

14. Envoyer des sms

Dans le cadre du projet au même titre que les mails, j'ai besoin d'envoyer des sms pour informer le client de son rendez-vous programmé le lendemain ainsi que d'envoyer un autre sms au praticien avec la liste de ses rendez-vous planifiés pour le lendemain.

L'envoi de sms depuis l'application est possible depuis le middleware via un module chargé de consommer un WebServices extérieur, celui-ci est mis à disposition par une API (Application Programming Interface).

Il existe des plateformes de services payantes sur Internet pour envoyer des sms. Dans le cadre de mon projet, je vais assumer le service d'envoi de sms via des outils gratuits à disposition sur Internet. Je m'appuie sur un mini-ordinateur destiné au grand public : le 'Raspberry Pi 3', installé avec un système d'exploitation linux 'Debian' et une API dédiée à l'envoi de sms via le réseau 3g/4G depuis le protocole HTTP. Pour envoyer des sms, il faut y connecter une clé usb 3G/4G pour accéder au réseau GSM. Pour



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

ce projet, je vais utiliser un abonnement gratuit fourni par l'opérateur Free. La combinaison de ces solutions me permet d'envoyer des sms depuis un web services personnalisé avec un coût réduit.



Le micro ordinateur au format de carte de crédit avec la clé 3G/4G connectée en Usb pour envoyer des sms depuis l' API RaspiSms. La documentation et les sources de RaspiSms sont accessibles depuis Github : <https://github.com/RaspbianFrance/RaspiSMS>

Voici la décomposition colorée de l'url fournie par RaspiSms :

```
http://URLCIBLE/smsAPI/?email=test@test.com\&password=LeMotDePasse\&numbers=0611223  
344\&text=Texte%20du%20SMS%20à%20envoyer
```

La partie de l'url en vert correspond à l'adresse de l'API, la partie en bleu correspond à l'authentification, le numéro de portable de destination est en marron et en bleu clair. C'est le message à envoyer au format. Mon middleware consomme cette ressource de type WebService à chaque besoin d'envoyer un SMS.

Stratégie d'hébergement

Au moment où je rédige ma soutenance, la partie base de données, le middleware et l'IHM sont hébergés dans le cloud Azure. J'ai réservé un nom de domaine www.opusbeauté.fr pour une année chez le fournisseur français OVH.

J'ai choisi une infrastructure en tant que Service (IaaS), une forme de Cloud Computing offrant des ressources informatiques au sein d'un environnement virtualisé (le Cloud) par le biais d'Internet ou d'une autre connexion. L'IaaS est l'une des quatre principales catégories de services Cloud Computing, au même titre que le Software as a Service (SaaS), le Desktop as a Service (DaaS), et la Platform as a Service (PaaS). C'est la première fois que j'utilise un service Cloud de type IaaS alors pour l'occasion, je voulais tester l'utilisation des services de Microsoft. J'ai aimé la simplicité pour mettre en œuvre une machine virtuelle et d'y installer les ressources dont j'ai besoin pour mon projet.

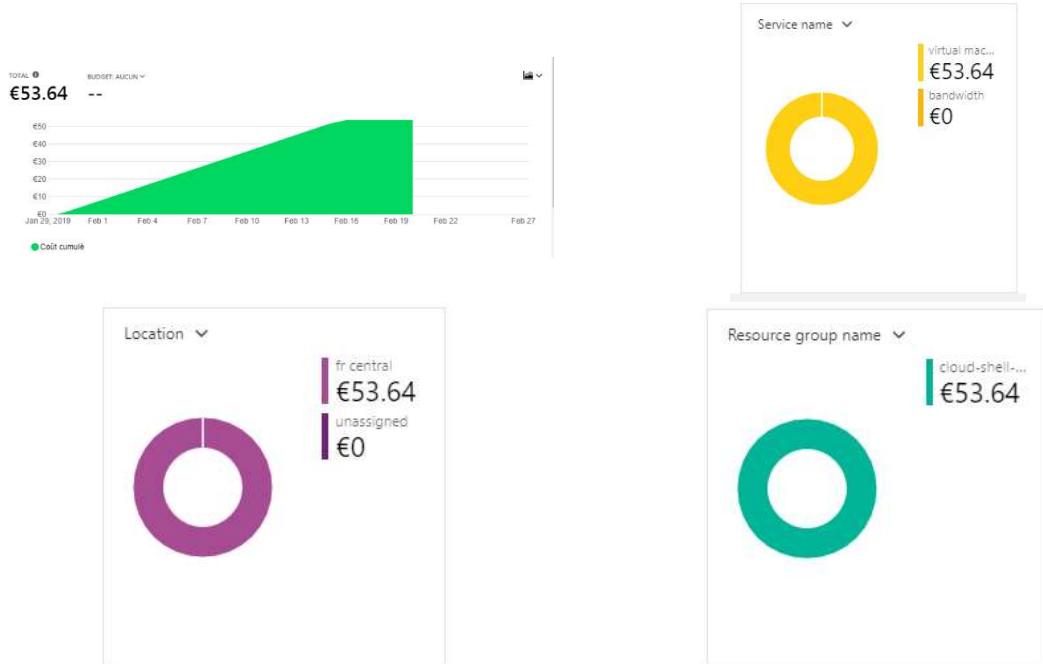


Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Les plateformes IaaS offrent des ressources hautement scalables, pouvant être ajustées sur demande en fonction des besoins de l'utilisateur. Les IaaS sont donc optimales pour les charges de travail temporaires, expérimentales ou soumises à des changements inattendus.

Ci-dessous les visuels des coûts d'une machine virtuelle avec de bonnes performances pendant vingt jours, à un coût de revient d'approximativement de 70 Euros par mois.



Statistiques d'utilisation et de coût d'une machine virtuelle hébergeant les services de l'application opusBeauté.

Je vais laisser la machine virtuelle fonctionner dans le cloud Azure le temps du projet jusqu'à la soutenance. Ensuite je remettrai mon projet sur une machine virtuelle hébergée chez moi en 'SelfHosting' pour continuer le développement. J'ai profité des accords de licences gratuits à disposition par le centre de formation pour monter un serveur Windows 2012 et de virtualiser sur ce serveur un système d'exploitation en Windows 10 comme station de développement, ce qui me permet faire du bureau à distance depuis une machine sans performances élevée. Cet accès au poste de développement déporté est possible depuis tout endroit pourvu d'un accès à Internet (adsl, hotspot wifi, partage de connexion 4G...).

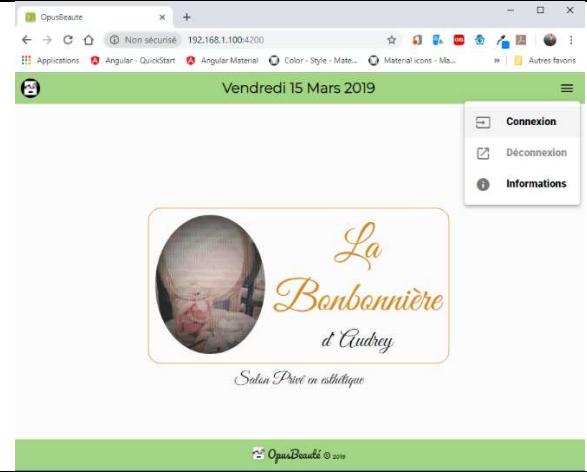
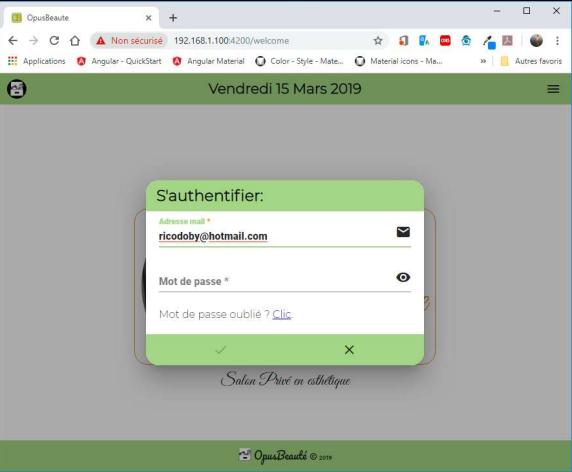
Résultats

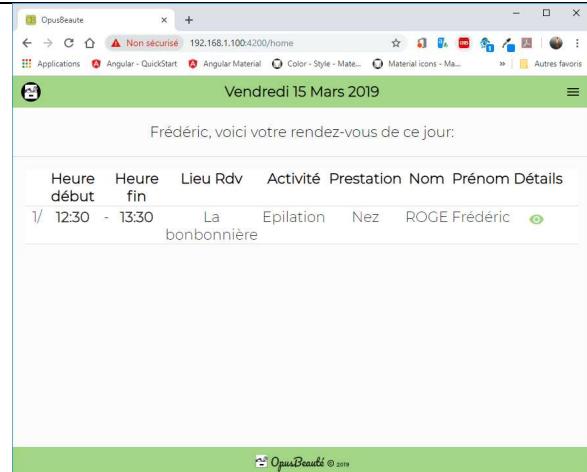
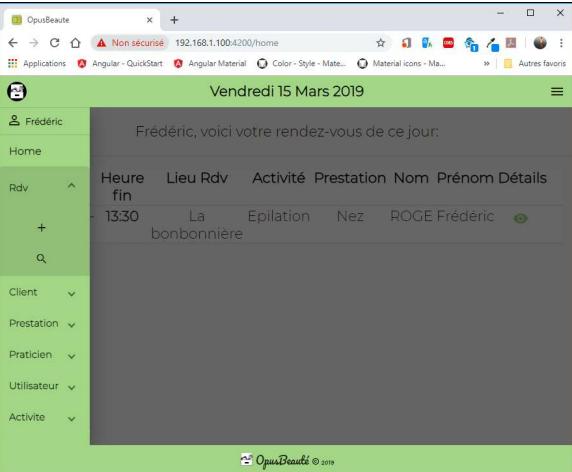
J'ai réalisé la majeure partie des User Stories de mon projet, en ce qui concerne la partie IHM il reste à produire les parties « Modifier » un utilisateur, client, une prestation, un praticien ainsi qu'activité. Je prévois de reprendre le développement de la partie modification à l'issue de la présentation de mon mémoire. A ce stade mon application est suffisamment fonctionnelle pour assurer la démonstration de l'envoi de rappel de rendez-vous via SMS ou Email au client et au praticien.

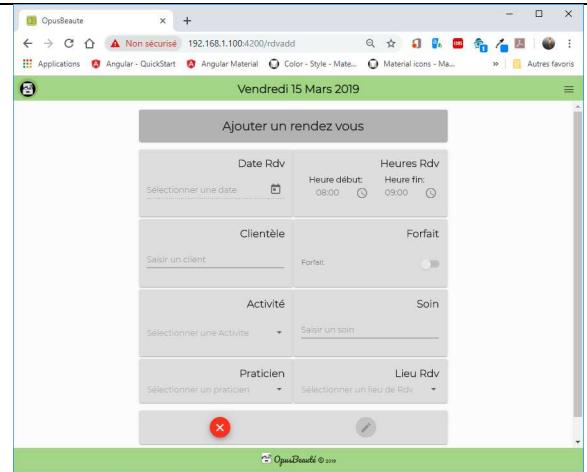
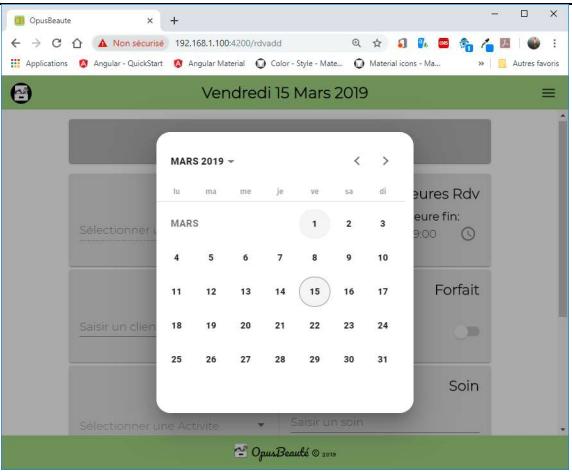


Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

	
<p>Landing page et menu de login</p>	<p>Authentification</p>

	
<p>Home page avec les rendez-vous du jour</p>	<p>Barre de menu (slide)</p>

	
<p>Ajouter un rendez-vous</p>	<p>Sélectionner une date</p>



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

<p>Sélectionner l'heure de début du rendez-vous</p>	<p>Sélectionner un client</p>

<p>Sélectionner un soin</p>	<p>Rechercher un rendez-vous</p>

<p>Modifier un rendez-vous</p>	<p>Forcer l'envoi de rappel de rendez-vous</p>



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

<p>La Bonbonnière d'Audrey : Rappel de Rdv</p> <p>D dobylov@gmail.com Jeu 14/03/2019 23:39 Vous ↗</p> <p>Bonjour Frédéric, vous avez rendez-vous demain à 08:30. Pour votre séance : <i>EPILATION_Sourcils</i>. Votre soin aura lieu à l' adresse suivante : <i>La bonbonnière</i> : 6 Rue de la pointe 60440 Nanteuil le Haudouin</p> <p>Cordialement, La Bonbonnière d'Audrey</p> <p>Se désinscrire au rappel de rendez-vous ici.</p>	<p>La Bonbonnière d'Audrey : Rappel de Rdv du 16-03-2019</p> <p>D dobylov@gmail.com Ven 15/03/2019 20:00 Vous ↗</p> <p>Bonjour fred, Vos rendez-vous de demain : 08:00 EPILACTION_Jambe Demi. Clientèle: PILLON Ingrid Lieu: La bonbonnière 6 Rue de la pointe 60440 Nanteuil le Haudouin</p> <p>17:00 EPILATION_Sourcils. Clientèle: PILLON Ingrid Lieu: Le Salon 23 Avenue du chemin de fer 93420 Villepinte</p> <p>20:00 FAUX ONGLES_1 Faux Ongle. Clientèle: PILLON Ingrid Lieu: Le Salon 23 Avenue du chemin de fer 93420 Villepinte</p> <p>Cordialement, La Bonbonnière d'Audrey</p>
<p>Email de rappel de rendez-vous client</p>	<p>Email de rappel de rendez-vous praticien</p>

<p>Bonjour Ingrid,</p> <p>La Bonbonnière d'Audrey vous rappelle votre rendez-vous de demain à 20:00. Pour le soin : FAUX ONGLES_1 Faux Ongle</p> <p>A l'adresse suivante : Le Salon 23 Avenue du chemin de fer 93420 Villepinte</p> <p>Cordialement, La Bonbonnière.</p> <p> <input type="button" value="Message"/> </p>	<p>Bonjour Fred, Vos rendez-vous de demain : 08:00 EPILACTION_Jambe Demi. Clientèle : PILLON Ingrid. Lieu : La Bonbonnière 6 Rue de la pointe 60440 Nanteuil le Haudouin</p> <p>17:00 EPILATION_Sourcils. Clientèle : PILLON Ingrid. Lieu : Le Salon 23 Avenue du chemin de fer 93420 Villepinte</p> <p>20:00 FAUX ONGLES_1 Faux Ongle. Clientèle : PILLON Ingrid. Lieu : Le Salon 23 Avenue du chemin de fer 93420 Villepinte</p> <p>Cordialement, Opus Beauté</p> <p> <input type="button" value="Message"/> </p>
<p>SMS de rappel de rendez-vous client</p>	<p>SMS de rappel de rendez-vous praticien</p>



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

La continuité du projet

Ce premier projet va faire l'objet de modifications afin de le faire évoluer vers d'autres technologies. Il sera très intéressant de modifier la vue de l'affichage d'un nouveau rendez-vous et d'y intégrer un calendrier pour un visuel global et précis mais surtout plus ergonomique.

Je vais migrer mon middleware développé en Java 8, en modifiant mon code pour être conforme à Java11. Depuis Java 9, java implémente de nouvelles fonctionnalités dont Jigsaw pour rendre l'application modulaire. Le but premier de Jigsaw est de réduire la taille des applications en n'embarquant que les dépendances nécessaires, afin de déployer le logiciel sur des environnements où l'espace de stockage peut coûter cher, comme l'IoT ou le Cloud.

Le format de la base de données est à ce jour en postgresSQL et je vais essayer de remplacer celle-ci par GraphQL, ce qui me permettra de cultiver d'autres méthodes innovantes pour requêter une base de données. J'ai envie de continuer avec le langage Java et d'ajouter la compétence du Framework Spring, très utilisé en entreprise. Plus tard, je me pencherai sur d'autres Frameworks Javascript pour développer une autre IHM en React Js afin de faire une prise de contact avec entre autres la gestion du DOM virtuel.

J'ai aussi l'idée de découper mon application en de multiples parties autonomes pour créer des MicroServices et de les intégrer dans des containers docker.

J'aime mon premier projet de développement et j'éprouve l'envie de continuer à le développer, car c'est une base qui me paraît suffisante pour réaliser diverses améliorations en terme de code, de Refactoring et de technologies à intégrer. Il me semble très important de faire de la veille sur les différentes techniques pour être en mesure d'avoir un panel de solutions afin de mettre en place les réponses les plus adaptées face aux problématiques.

Conclusion

Mon idée initiale était d'étendre mes compétences dans l'informatique avec l'apprentissage d'un langage de programmation orienté objet. En choisissant la formation de Concepteur Développeur Informatique, j'ai pu assimiler des techniques qui m'ont permis d'aller bien au-delà de mon idée de départ et d'intégrer des méthodes de gestion de projet dédiées au monde du logiciel. Ce projet m'a donné l'occasion de mettre en pratique les très connues méthodes Agile et Merise et UML. Avec ce projet, j'ai constaté qu'il était crucial de privilégier l'écoute du client afin de produire un logiciel conforme aux attentes et être en phase pour synthétiser les aspects métier de son activité professionnelle. Et pour aider ma cliente à se représenter les actions réalisables sur le logiciel et du design de l'interface, j'ai modélisé des documents visuels. Le déroulement du projet gravite constamment autour du client / utilisateur pendant le développement afin de recueillir son « feedback » au moment d'une présentation d'un livrable.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

J'ai aussi compris que la combinaison de ces méthodes de gestion de projet m'ont permis de découper ce qui semblait initialement être un gros projet en différentes petites parties techniquement abordables.

Pour conclure ce mémoire, je suis satisfait de ce projet dont j'ai abordé la conception jusqu'au développement. Celui-ci présente les principales fonctionnalités attendues dans le cadre de la méthode Agile et je suis en mesure de continuer l'implémentation de nouvelles fonctionnalités pour le plaisir.

Le développement de logiciel nécessite une perpétuelle veille technologique afin d'être en phase avec les nombreuses technologies existantes et émergeantes, ce qui à mon sens donne des perspectives passionnantes, riches et intéressantes à ce métier.

A ce jour, je suis aux prémices de la maîtrise des technologies de développement. Je suis conscient que c'est l'enchaînement de projets qui constitueront mon capital d'expertise dans ce domaine. Et de ce fait, grâce à ces technologies qui évoluent, je suis enthousiaste à l'idée d'avoir des possibilités d'apprentissage sur le long terme.

Je suis fier d'avoir entrepris cette démarche de formation dans ma carrière professionnelle. Mes nouvelles compétences, jumelées avec mon expertise dans le système et réseaux, ont suscité l'intérêt d'Entreprises de Service du Numérique et cela a abouti une proposition d'embauche. En effet, je suis actuellement en poste en qualité de responsable de site à Villepinte (93) pour le compte d'un Gestionnaire d'activités économiques internationales.



Projet OpusBeauté

Soutenance CDI,
le 27/03/2019 à
DORANCO Nation

Bibliographie

Dans ce projet je me suis fortement appuyé sur de la documentation à disposition sur Internet. Les sources étant soit des articles ou de la documentation en anglais et ou en français. Je me suis aussi documenté via des ouvrages recommandés en formation mais aussi par découverte personnelle.

Titre	Auteur(s)	Edition	Année
Programmer en Java 9 ^e édition	Claude Delannoy	Eyrolles	2017
Java EE6 et GlassFish3	Antonio Goncalves	Pearson	2009
Beginning Java EE7	Antonio Goncalves	Apress	2013
JPA et Java Hibernate – Apprenez le mapping Objet-relationnel (ORM) avec Java	Martial Banon	Edition ENI	2017
Développer des services REST en Java - Echanger des données au format JSON	Aurélie Sobrero	Editions ENI	2014
UML 2 en action de l'analyse des besoins à la conception 4 ^e édition	Pascal Roques, Franck Vallée	Eyrolles	2007
Design Patterns les 23 modèles de conception illustrés en UML2 et Java	Laurent Debrauwer	Editions ENI	2007
Apprendre à développer avec JavaScript 2 ^e édition	Christian Vigouroux	Editions ENI	2016
La méthode Merise Tome 1, Principes et outils	Hubert Tardieu, Arnold Rochefield, René Colletti	Les éditions d'organisation	1983
La méthode Merise Tome 2, Démarche et pratiques	Hubert Tardieu, Arnold Rochefield, René Colletti, Georges Panet, Gérard Vahée	Les éditions d'organisation	1985

