



STAGE EN ENTREPRISE

3^{ÈME} BACHELIER EN INFORMATIQUE

Développement d'un outil de gestion de ressources (côté client)

Realdolmen

Avenue Victor Maistriau, 30
7000 Mons

Étudiant :

Alexandre DUCOBU

Responsable du stage :

Gianni TRICARICO

Maitre de stage :

Gino SPRIO



Campus
technique

Année académique 2017 - 2018



STAGE EN ENTREPRISE

3^{ÈME} BACHELIER EN INFORMATIQUE

Développement d'un outil de gestion de ressources (côté client)

Realdolmen

Avenue Victor Maistriau, 30
7000 Mons

Étudiant :

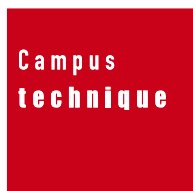
Alexandre DUCOBU

Responsable du stage :

Gianni TRICARICO

Maitre de stage :

Gino SPRIO



Année académique 2017 - 2018

Table des matières

Table des figures	5
Abstract	6
Version française	6
Version anglaise	7
1 Présentation	8
1.1 Introduction	8
1.2 Objectif d'un stage en entreprise	8
1.3 Choix du stage	8
1.4 But du stage	9
1.5 L'équipe de travail	9
1.6 Planning	10
2 Présentation de l'entreprise	11
2.1 L'entreprise	11
2.2 L'organisation de l'entreprise	12
2.3 Le maitre de stage	13
2.4 Autres membres de l'équipe	13
3 Présentation du projet	14
3.1 Le projet	14
3.2 Architecture et choix technologiques	15
3.3 Les outils	16
4 Analyse	18
4.1 La phase d'analyse	18
4.2 UML	18
4.3 Lexique et conventions	19
4.4 Cas d'utilisation	19
4.5 Exigences fonctionnelles et non-fonctionnelles	20
4.6 Profils et rôles	21
4.7 Interface	22
5 Conception	23
5.1 Organisation de la phase de développement	23
5.2 Premier sprint	24
5.3 Deuxième sprint	24
5.4 Troisième sprint	25
5.5 Quatrième sprint	25
5.6 Cinquième sprint	26
5.7 Sixième sprint	26
5.8 Résultat	27

6	Tests & bugs fixing	28
6.1	Organisation	28
6.2	Nouvelles fonctionnalités	29
6.3	Documentation	30
7	Réalisation	31
7.1	Développement du front-end	31
8	Le relationnel	35
8.1	Communication des informations	35
8.2	Difficultés	36
8.3	Intégration dans l'entreprise	37
9	Conclusion	38
9.1	Conclusion du projet	38
9.2	Apports du stage	41
9.3	Conclusion	41
	Lexique	42

Table des figures

Figure	Titre	Page
1	Aperçu du fichier Excel de base	9
2	Le planning du stage sur 15 semaines	10
3	Logo de Realdolmen	11
4	L'organigramme de Realdolmen Sud	12
5	Architecture et technologies de l'application	15
6	Logo de Spring	16
7	Logo de git	16
8	Logo d'IntelliJ IDEA	17
9	Logo d'Atom	17
10	Version finale de la vue de la liste des clients	22
11	Aperçu du fichier Excel contenant les bugs remontés	28
12	Barre d'actions simplifiée	32
13	Modale affichant un message d'erreur	33
14	Liste déroulante	38
15	Barre d'actions	39
16	Modale de confirmation	40

Abstract

Version française

En cette troisième et dernière année de bachelier en Informatique & systèmes, il nous a été demandé d'effectuer un stage en entreprise. Je l'ai accompli en compagnie de trois autres stagiaires : deux venants de la haute école Condorcet, et le troisième de la HEH.

Le but de ce stage était de développer un outil de gestion de ressources pour Real-dolmen Sud qui sera utilisé par son *Delivery Manager*.

Celui-ci a été développé autour de technologies web et d'une architecture multi-tiers standard.

Quant à la méthodologie de gestion du projet, nous avons suivi la méthode Agile.

La durée du stage était de treize semaines.

Les trois premières ont été dédiées à l'analyse du projet et de ses besoins. Ensuite, nous avons développé l'outil pendant six semaines, à raison d'un sprint par semaine. Les quatre dernières semaines nous ont servi à tester et à déboguer le code.

Version anglaise

In this third and last year of our Bachelor's degree in Computer Science & Systems, we were asked to do a work placement. I carried it out with three other trainees : two from Condorcet Higher School and the third from HEH.

The aim of this internship was to develop a resource management tool for Realdolmen South that will be used by its *Delivery Manager*.

It was developed around web technologies and a standard multi-tiers architecture. As for the project management methodology, we used the Agile software development.

The duration of the internship was thirteen weeks.

The first three weeks were dedicated to the analysis of the project and its needs. Then we developed the tool for six weeks, one sprint a week. The last four weeks have been used to test and debug the code.

1 Présentation

1.1 Introduction

En cette troisième et dernière année de bachelier en Informatique & systèmes, il nous a été demandé d'effectuer un stage en entreprise.

Celui-ci est l'aboutissement de nos trois années d'apprentissage. Il nous permettra de comprendre ce qu'est le travail en entreprise puisque nous serons nous-même acteurs au sein de celle-ci.

1.2 Objectif d'un stage en entreprise

L'objectif principal du stage est l'intégration professionnelle du stagiaire.

Il doit s'insérer dans une équipe de travail, et communiquer avec son maître de stage et les autres acteurs de l'entreprise.

Il doit, bien entendu, respecter les horaires et être présent régulièrement.

De plus, il doit comprendre la structure de l'entreprise et acquérir un esprit d'équipe.

Le stage permet de mettre en œuvre des technologies qui n'ont pas été envisagées dans les cours, d'acquérir un esprit d'initiative et un sens pratique (*et de la pratique*) ainsi que de suivre les instructions données.

En entreprise, il faut planifier les tâches dans le meilleur ordre, se montrer rentable et travailler tant en autonomie qu'en équipe.

Il faut aussi s'initier à la problématique de sécurité de l'entreprise et en respecter les normes.

Il faudra atteindre les objectifs demandés en respectant les échéances et en rendant un travail de qualité.

En outre, le stagiaire devra se montrer efficace et effectuer un maximum de tâches.

1.3 Choix du stage

J'ai choisi ce stage, car il proposait un projet basé sur des technologies web telles que celles que nous avons pu apprendre lors de notre cursus.

Il me demanderait aussi d'apprendre pour, par exemple, importer/exporter des données dans l'application, gérer différents profils pour la sécurité,...

De plus, la méthode Agile, abordée au cours de M. Tricarico, serait au centre de la méthodologie de gestion de projet.

Finalement, la proximité avec l'entreprise était aussi intéressante en cas de grèves ou de mauvais temps.

1.4 But du stage

Le but du stage est de développer une première version utilisable d'un outil de gestion de ressources pour Realdolmen Sud afin de faciliter le travail de monsieur Sprio, mon maitre de stage, qui, pour l'instant, utilise un fichier Excel.

De fait, l'extraction d'informations importantes est fastidieuse avec Excel. La gestion des données est lourde, il n'y a pas de données de références ce qui conduit à des informations réencodées, n'ayant pas toujours la même dénomination.

De plus, il n'y a aucun contrôle à l'encodage ce qui multiplie les risques d'erreurs.

Cet outil sera une application web et permettra d'afficher une liste des consultants, leur occupation, leurs projets et clients, etc.

Les données seront modifiables par le Delivery Manager et l'application sera en anglais pour une accessibilité maximale.

[illegible]

FIGURE 1 – Aperçu du fichier Excel de base

Le but de la figure ci-dessus n'est pas de vous montrer les données du fichier, mais de vous montrer le manque d'ergonomie et de lisibilité qu'offre le fichier Excel utilisé pour la gestion des ressources.

Remarquons que le fichier contient plus de 500 lignes identiques à celles visibles dans cet aperçu.

1.5 L'équipe de travail

L'équipe de développement est composée de quatre stagiaires :

- Arnaud Urbain et moi venons du campus technique de la HEH ;
- Les deux autres stagiaires, Kevin Chyzy et Pierre Duvivier, viennent de la haute école Condorcet. Leur stage leur sert de TFE et dure quinze semaines. Ils ont commencé le même jour que nous.

1.6 Planning

Le planning est découpé en trois phases :

- la phase d'analyse : les trois premières semaines du stage ;
- la phase de développement, découpée en six sprints d'une semaine ;
- la phase de tests et de corrections de bugs, se déroulant les quatre dernières semaines du stage.

Cette dernière durera encore 2 semaines lorsque Arnaud et moi aurons terminé, puisque la durée du stage des élèves de Condorcet est de 15 semaines. Les deux dernières semaines serviront principalement à la rédaction de la documentation technique et à la mise en production de l'application.

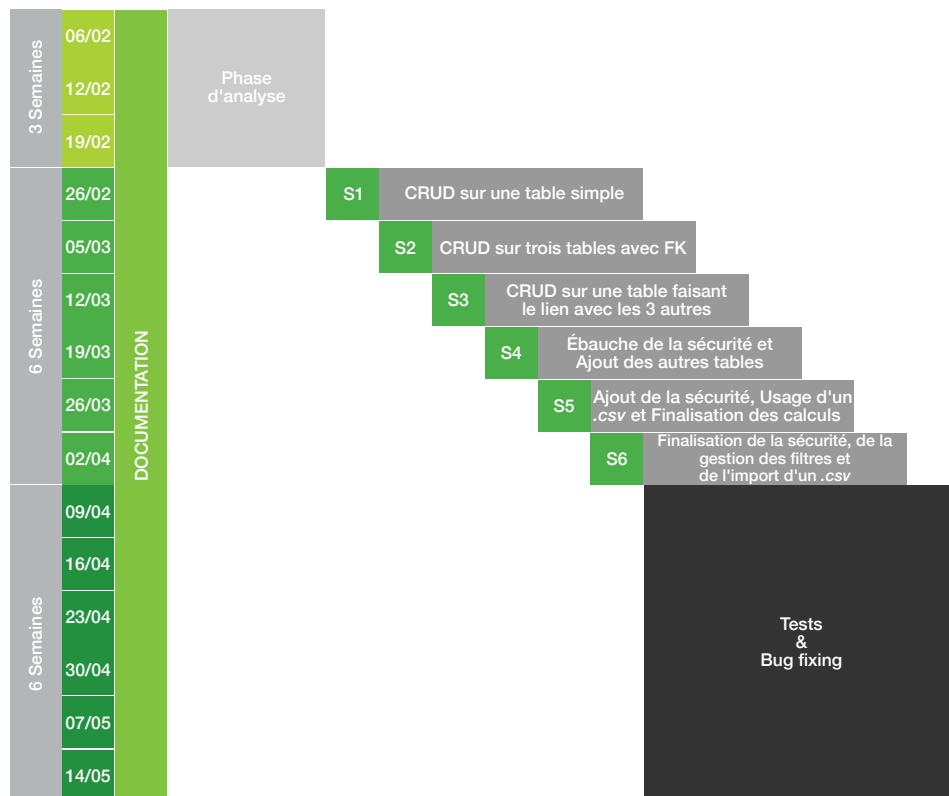


FIGURE 2 – Le planning du stage sur 15 semaines

2 Présentation de l'entreprise

2.1 L'entreprise

J'effectue mon stage chez Realdolmen, expert ICT indépendant.

Cette entreprise de consultance informatique belge, l'une des plus importantes, accompagne ses clients tout au long de leurs projets ICT. Ses principaux segments d'activité sont l'*IT & business consulting services* et l'*IT business support*, avec une position particulièrement forte auprès des entreprises de tailles moyennes supérieures.

La société compte près de 1 250 professionnels hautement qualifiés qui offrent leurs services informatiques à plus de 1 000 clients dans le Benelux, tant sur le plan stratégique, tactique, qu'opérationnel.

Comme son nom l'indique, Realdolmen est une Société Anonyme.

L'entreprise est principalement localisée en Flandres.

Elle est divisée en 8 établissements dont un en Wallonie, *Realdolmen Sud*, et un au Grand-Duché de Luxembourg.

Le siège de Realdolmen se trouve à Huizingen. Son adresse est Avenue Vaucampslaan 42, à Beersel. On peut les joindre par téléphone à ce numéro +32 2 801 55 55 et par mail à *info@realdolmen.com*.

Realdolmen Sud se trouve à Mons, et plus précisément Avenue Victor Maistriau, n°30. On peut les joindre par téléphone à ce numéro +32 2 801 63 00.

L'équipe de travail à laquelle j'appartiens possède un bureau avec une baie vitrée et un petit balcon. Les bureaux sont bien aérés et communiquent entre-eux.



FIGURE 3 – Logo de Realdolmen

À la fin de la troisième semaine de stage, le groupe français *Gfi Informatique* a lancé une OPA amicale sur Realdolmen qui l'a acceptée.

Le but de l'opération est de renforcer l'implantation du groupe Gfi en Belgique et au Luxembourg. Gfi Informatique s'appuiera sur le management et sur les salariés de Realdolmen pour continuer à développer leur plateforme au Benelux^{1 2}.

1. <http://www.realdolmen.com/fr/news/projet-doffre-publique-dachat-amicale-de-realdolmen-par-gfi-informatique>

2. <https://www.lecho.be/entreprises/services/gfi-lance-une-opa-sur-realdolmen/9985577.html>

2.2 L'organisation de l'entreprise

Le directeur de l'exploitation (*COO*) de Realdolmen est Marc De Keersmaecker et le secrétaire-général est Thierry de Vries.

Depuis 2009, Paul De Schrijver en est le directeur administratif et financier (*CFO*).

En novembre 2012, c'est Johnny Smets qui est devenu directeur général.

En 2014, William De Plecker est devenu directeur des ressources humaines (*CPO*).

Enfin, c'est Tim Claes qui, depuis mars 2015, est le directeur des ventes et du marketing.

Du côté de Realdolmen Sud, c'est Eddy Volcher qui est directeur général.

Nous l'avons ainsi souvent rencontré.

Voici l'organigramme de Realdolmen Sud :

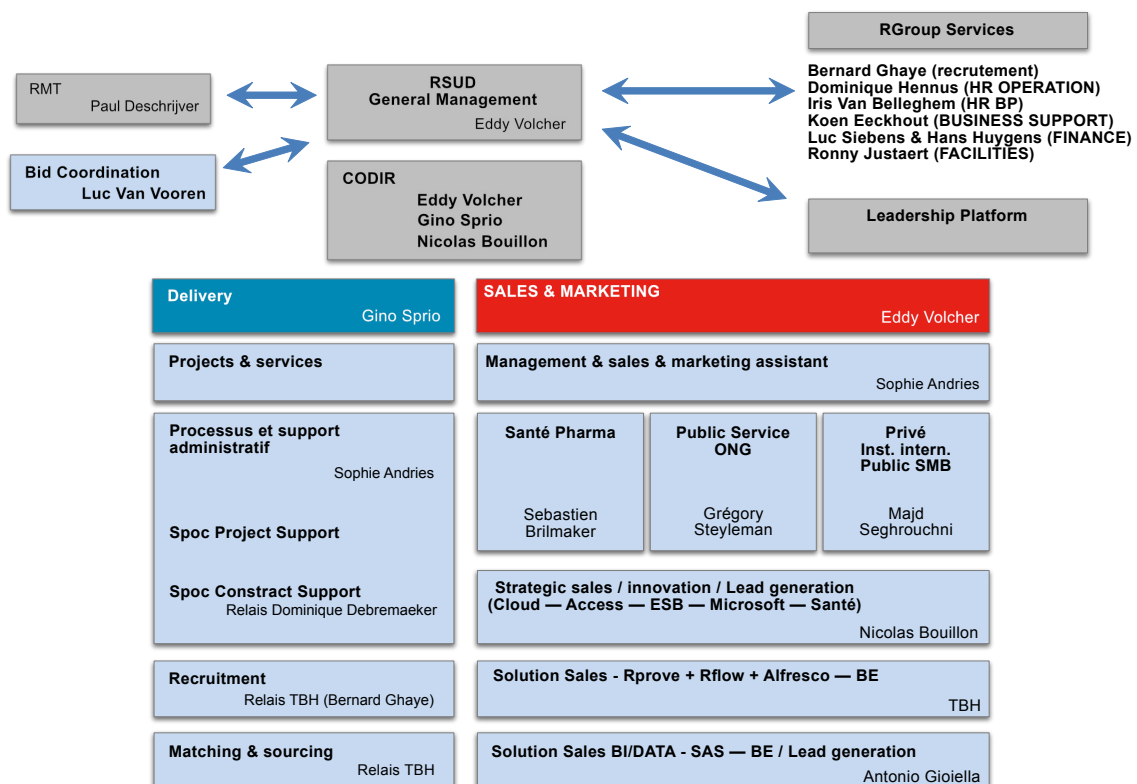


FIGURE 4 – L'organigramme de Realdolmen Sud

2.3 Le maitre de stage

Notre maitre de stage, monsieur Gino Sprio, est le Delivery Manager de Realdolmen Sud (*Rsud*).

Un Delivery Manager est responsable du suivi des prestations réalisées chez les clients qui lui sont confiés.

C'est donc lui qui s'occupe de la gestion des ressources de Rsud (*consultants, projets, plannings,...*).

Monsieur Sprio est donc la personne qui nous a donné les objectifs du stage et c'est à lui que nous présentons notre avancement.

2.4 Autres membres de l'équipe

Nous partageons notre bureau avec un troisième stagiaire de la haute école Condorcet, Diego Rodriguez. Il travaille sur une application mobile développée avec Xamarin.

Nous sommes donc encadrés par notre maitre de stage, monsieur Sprio, mais aussi par monsieur Bouillon, *Head of Strategic Sales and Innovation* de Rsud.

Lors de la phase d'analyse, nous avons, *en moyenne*, une réunion par jour avec eux. Monsieur Bouillon passe souvent dans notre bureau afin de voir notre avancement. Il en profite aussi pour nous donner de précieux conseils que ce soit dans le fonctionnement ou l'affichage du projet.

Pour finir, nous étions en contact (*via Slack*) avec monsieur Boucher, architecte applicatif et support technique ponctuel.

Il est aussi venu nous expliquer et nous montrer comment bien démarrer le projet avec les différentes technologies choisies.

3 Présentation du projet

3.1 Le projet

Le projet consiste à développer un outil de gestion de ressources pour Rsud. Celui-ci remplacera l'utilisation d'un fichier Excel dont la gestion est lourde. En effet, celui-ci ne permet pas l'extraction d'informations importantes de manière simple et efficace.

Le nombre de consultants et de projets augmentant, l'ergonomie diminue de manière exponentielle étant donné qu'un consultant peut travailler simultanément sur plusieurs projets. Ce manque d'ergonomie entraîne une maintenabilité difficile.

Pour finir, l'utilisation d'Excel ne permet pas de contrôles à l'encodage et ne possède pas de données de références, ce qui induit des données mal formatées ou formatées de différentes manières.

Tout cela fait perdre un temps précieux à M. Sprio.

Notre projet propose une meilleure productivité ainsi qu'une meilleure maîtrise des données grâce à l'utilisation de formulaires permettant la vérification des données ainsi que l'enregistrement de données de référence au sein de l'application.

Celui-ci affiche les données dans des vues ergonomiques telles que des tableaux listant les consultants, clients ou projets, ou des vues personnelles pour chacun de ces objets. Celles-ci sont différentes d'après le rôle de l'utilisateur : administrateur, directeur des ressources (*resources manager*) ou commercial (*account manager*).

Nous proposons aussi une vue prévisionnelle, c'est-à-dire que, dans la liste des consultants et sur la page d'accueil, l'utilisateur peut retrouver une prévision de l'occupation des consultants pour les douze mois à venir.

Les données peuvent être facilement filtrées, importées au format `.csv` et exportées au format Excel (`.xlsx`).

Finalement, l'avantage de la technologie web est que, quel que soit le matériel de l'utilisateur (*ordinateur, tablette ou smartphone*), celui-ci aura accès à l'application, aux données et aux mêmes fonctionnalités.

L'outil doit permettre de :

- donner une vue du planning d'occupation des consultants ;
- rentrer/éditer les occupations des consultants par projet/mission ;
- importer/éditer la liste des projets, consultants et clients ;
- gérer un minimum de 4 profils utilisateurs (*sécurité et vues personnalisées*) ;
- gérer la notion de planning définitif et prévisionnel ;
- exporter un certain nombre d'informations en Excel, CSV ou tout autre format utilisable avec Office.

3.2 Architecture et choix technologiques

Avec l'aide de monsieur Boucher, support technique ponctuel et architecte applicatif, nous avons choisi une architecture multi-tiers en cinq couches.

C'est l'architecture standard utilisée depuis quelques années pour ce genre de projets. Elle est donc documentée et éprouvée par de nombreux développeurs dans le monde.

La première couche est la couche **Presentation** (*front-end, ou partie client*), c'est moi qui en suis le responsable. C'est par elle que l'utilisateur a accès et utilise l'application.

J'ai choisi d'utiliser le trio HTML5, CSS et JavaScript (*épaulé par jQuery*).

Côté design et responsive design, nous nous sommes basés sur Bootstrap et, plus particulièrement, sur AdminLTE³. Celui-ci est basé sur la version 3 de Bootstrap et est l'un des templates le plus utilisé pour la partie d'administration des sites.

Ensuite, vient le back-end, ou partie serveur.

La couche **REST**, dont Kevin Chyzy est responsable, fait le lien entre le front-end et le back-end.

En effet, c'est par elle que le front-end reçoit et renvoie les données au back-end.

Ici encore, nous utilisons une technologie standard : une API REST.

Le back-end comprend encore trois couches **Service**, **Repository** et **Database**.

La première est développée en Java à l'aide de Spring et de Maven. C'est Pierre Duvivier qui en est responsable.

Elle est principalement responsable des calculs et opérations du back-end.

Les deux suivantes sont liées : la couche **Database**, comprenant une base de données, est créée et rendue accessible par la couche **Repository** en utilisant Hibernate.

Côté sécurité, nous utilisons OAuth 2.0 et passons par l'authentification LDAP de Realdolmen.

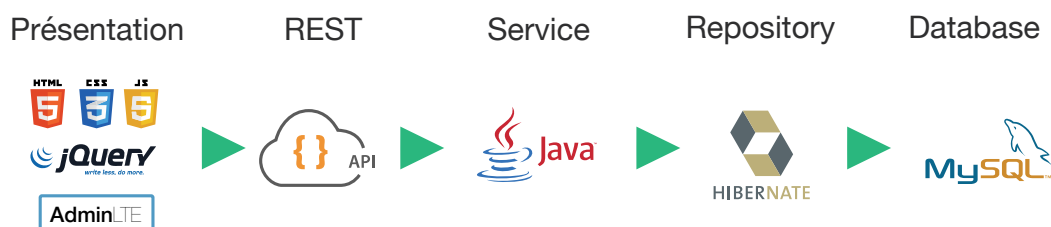


FIGURE 5 – Architecture et technologies de l'application

3. Lien vers AdminLTE : <https://adminlte.io>

3.3 Les outils

Nous avons commencé par choisir **Spring Boot** pour le développement du back-end. Ce framework⁴ libre permet de construire et de définir l'infrastructure d'une application Java (back-end), dont il facilite le développement et les tests. Il s'agit d'une solution modulaire, ce qui en fait sa force et explique, avec sa richesse et son efficacité, l'engouement des développeurs pour ce framework.

Le Java permet d'utiliser la programmation orientée objet. Du côté de la gestion des dépendances et de la compilation du site, nous avons utilisé **Apache Maven** ou, comme il est plus communément nommé, *Maven*.

Afin de tester les appels REST sans passer par le front-end, nous avons utilisé **Swagger**. Il affiche les paramètres à insérer et permet de retrouver l'URL à utiliser et les données retournées par chaque appel.

De plus, Kevin pouvait tester différents appels (*ajout, listage, modification,*) et voir s'ils fonctionnaient correctement avant de les mettre à disposition de l'équipe.

Pour ce qui est de la base de données, nous avons utilisé **H2** qui est open source et qui est créée en mémoire vive. Cela nous a permis de faire de nombreux tests en local sans avoir de répercussions sur l'état de la base de données vu qu'elle était réinitialisée à chaque compilation de l'application.

Sur les serveurs de tests et de production, nous utilisons une base de données MySQL. Au cours du développement, nous avons utilisé deux logiciels de gestion de versions (*VCS*) différents.

Premièrement, **Bitbucket** qui a été écrit en Python et permet d'utiliser *git* et *Mercurial*. C'est M. Boucher qui l'a initialisé pour nous. Il a été utilisé toute la durée du développement et pendant la majorité de la phase de bug fixing.

Dans un second temps, nous avons utilisé **Gogs** (*Go Git Service*) qui est basé sur *git* et a été écrit en Go. Nous avons commencé à l'utiliser lorsque l'un des consultants nous en a parlé.

En effet, Realdolmen Sud possède un serveur *Gogs* autohébergé, ce qui apporte une sécurité et une rapidité de connexion accrues.



FIGURE 6 – Logo de Spring



FIGURE 7 – Logo de git

4. Voir *Framework* dans le **Lexique**.

Du côté des logiciels, vu que le back-end est développé à l'aide de Java, nous avons utilisé **IntelliJ IDEA** (*version Community*) qui est l'un des IDE Java les plus utilisés. De plus, son intégration avec git a facilité le travail de nos collègues qui n'avaient pas l'habitude d'utiliser les logiciels de gestion de versions.

Pour le développement du front-end, l'outil important peu, j'ai décidé d'utiliser **Atom**, un éditeur de texte libre et multi-plateforme. Il a la particularité d'être modulable et de pouvoir se transformer en IDE.

Vu que j'emploie dans la majorité de mes projets, il était plus intéressant que je l'utilise plutôt que d'apprendre à utiliser un nouvel outil.

Pour utiliser notre application web, nous avons le choix du navigateur. Notre maitre de stage nous a parlé d'*Internet Explorer*, mais, pour une compatibilité maximale, nous avons convenu de nous baser sur **Google Chrome**, installé par défaut sur les machines de Realdolmen.

Lors du développement du front-end, j'ai utilisé plusieurs navigateurs afin de garantir la meilleure compatibilité possible : *Google Chrome*, *Firefox*, *Opera* et *Safari*.

J'ai aussi utilisé, un moment, *Microsoft Edge*, mais celui-ci comptait bien souvent des incompatibilités. Par exemple, la page de connexion se résume à un formulaire sur Edge, alors que les autres navigateurs affiche une image de fond et le design de l'application.



FIGURE 8 – Logo d'IntelliJ IDEA



FIGURE 9 – Logo d'Atom

4 Analyse

4.1 La phase d'analyse

Notre analyse a duré trois semaines et a débuté le premier jour du stage.

Dès ce jour, nous nous sommes mis d'accord sur les technologies et l'architecture que nous utiliserions, d'après les informations données par monsieur Boucher.

C'est ainsi que l'architecture multi-tiers en cinq couches a été choisie.

La répartition des tâches, un stagiaire par couche, s'est faite d'après les préférences et connaissances de chacun.

Ensuite, dès le second jour, l'analyse en tant que telle a commencé.

La première chose à laquelle nous nous sommes attaqués était l'UML⁵ de la base de données.

4.2 UML

En effet, l'UML⁶ est très important pour l'application et représente notre compréhension du projet.

C'est ainsi que, le jour même, nous avons eu une réunion avec messieurs Sprio et Bouillon afin d'avoir rapidement des retours et de ne pas nous enfoncer dans de mauvais raisonnements.

Il s'est avéré que nous étions bien partis et, pour continuer de la sorte, nous avons eu deux *points* journaliers avec eux.

Tout au long de la phase d'analyse, nous avons amélioré l'UML au fur et à mesure que nous mettions de nouvelles fonctionnalités ou besoins à jour, ou que des manques étaient pointés par nos référents.

Nous sommes ainsi passés d'un UML simple, principalement constitué de tables vides, à un UML plus complexe, possédant de nombreux champs dans ses tables, une table servant à l'historique des assignements et à la mise en évidence des clés étrangères.

5. Voir *UML* dans le **Lexique**.

6. Voir la page 4 dans les **Annexes**

4.3 Lexique et conventions

Parallèlement à la création de l'UML de la base de données, j'ai rédigé un lexique des termes utilisés que ce soit pour des termes utilisés par l'entreprise, ou pour nos propres termes.

Par exemple, le terme **AKP**, pour *Aankopen Prijs*, signifie *Prix d'achat* du consultant.

Bien entendu, ce lexique nous a mené à définir des conventions.

Le code de l'application, ainsi que sa documentation, sont écrits en anglais.

De plus, le code suit la convention de dénomination ci-dessous :

- le nom des tables de la base de données commence par une majuscule et est au singulier ;
- le nom des classes Java est lié à la couche logicielle à laquelle elles appartiennent. Par exemple, celles faisant le lien entre les couches **Presentation** et **REST** sont au format **xxxDto**.
- en ce qui concerne le format de nommage des différentes tables, classes, méthodes et variables du projet, nous avons choisi la convention chameau (*ou camel case*).

4.4 Cas d'utilisation

Les principaux utilisateurs de la solution seront monsieur Sprio et madame Heuchamps qui exploiteront toutes les fonctionnalités en tant qu'administrateurs.

Les administrateurs sont les seuls à avoir accès aux données financières de type **AKP**.

Les commerciaux pourront utiliser l'outil pour vérifier la disponibilité de consultants pour répondre à leurs offres et consulter tous les projets des clients auxquels ils sont rattachés.

Avec les administrateurs, ils sont les seuls à avoir accès aux données financières de type **VKP**.

Les chefs de projets pourront consulter tout ce qui touche aux projets sur lesquels ils travaillent et à leurs équipes.

Les consultants pourront consulter tout ce qui touche aux projets sur lesquels ils travaillent et toutes les données qui les concerne.

Un formulaire de soumission de modification des occupations par les commerciaux, les chefs de projets ou les consultants est envisagé.

Ils demandent une adaptation, et le gestionnaire vérifie les infos et les valide, en les adaptant si nécessaire.

4.5 Exigences fonctionnelles et non-fonctionnelles

Une fois cela fait, nous avons commencé par lister les exigences fonctionnelles, celles demandées explicitement, et les exigences non-fonctionnelles, celles qui sont explicites.

Certaines des exigences fonctionnelles sont donc :

- Visualiser le planning d’occupation des consultants.
- Rentrer/éditer les occupations des consultants par projet.
- Importer/éditer la liste des projets, consultants et clients.
- Gérer un minimum de 4 profils utilisateurs en matière de sécurité.
- Afficher différentes vues en fonction du profil utilisateur connecté.
- Gérer les notions de planning définitif et prévisionnel.

Les exigences non-fonctionnelles sont, elles :

- L’ergonomie :
 - L’application doit être intuitive pour permettre une prise en main facile et efficace.
 - L’application doit être utilisable par des personnes parlant des langues différentes.
 - L’application doit respecter la charte graphique de l’entreprise et répondre aux exigences visuelles du client.
- La sécurité :
 - L’application doit comprendre différents niveaux de sécurité.
 - Les appels aux services web doivent être effectués directement via l’application.
- L’accessibilité :
 - L’application devra être accessible dans le navigateur choisi par l’utilisateur qui sera connecté à l’active directory de Realdolmen.

Et chacune de ces fonctionnalités est divisée en fonctionnalités métiers, plus simples et plus *atomiques*.

Pour éditer la liste des projets, nous avons alors des fonctionnalités métier telles que : *Créer un projet*, *éditer un projet* et *Supprimer un projet*.

4.6 Profils et rôles

Nous sommes alors passés à l'avant-dernière étape de l'analyse : le choix des différents profils disponibles dans l'application et leurs rôles.

Nous sommes partis, dans un premier temps, sur six profils : *Administrateur système*, *Administrateur*, *Ressources manager*, *Account manager*, *Project manager* et *Consultant*.

Mais nous avons rapidement fusionné l'administrateur système et l'administrateur pour plus de facilité. L'utilité de l'administrateur système était d'avoir un utilisateur spécifique pour la gestion des utilisateurs de l'application et pour la modification de leur profil. Désormais, c'est l'administrateur qui se charge de ce rôle.

L'administrateur a un accès de façon générale à toutes les informations de l'application. Il peut donc lire et modifier toutes les informations, même les données de références.

Le ressources manager a les mêmes privilèges que l'administrateur, mais seulement sur le ou les cost centers dont il est responsable. De plus, il n'a pas accès aux données de référence et n'a pas la possibilité d'importer des données depuis un fichier *.csv*. L'account manager a un accès en lecture de toutes les informations de la base de données, sans les prix et sans les données de référence. Les vues pouvant être triées par rapport à un account manager le sont pour celui qui est connecté. Il peut effectuer des demandes qui devront être validées par un administrateur ou un ressources manager.

Le project manager peut consulter tous ses projets, les consultants assignés à ses projets et a aussi accès à une vue sur la disponibilité des autres consultants. Il peut, tout comme l'account manager, effectuer des demandes qui devront être validées.

Pour finir, le consultant n'a accès, en lecture seule, qu'à son occupation et à ses données personnelles.

Malheureusement, suite à une surcharge de travail de monsieur Boucher qui devait nous aider sur l'implémentation de la sécurité, nous avons dû réduire le nombre de profils disponibles dans l'application pour cette première version. Désormais, seuls les trois profils les plus importants sont intégrés, et la fonctionnalité des demandes à valider a été retirée.

Les profils disponibles sont donc : *Administrateur*, *Ressources manager* et *Account manager*. Quant à celui du consultant, le premier à avoir été implémenté, il est fonctionnel, mais n'a ni été testé, ni approuvé.

4.7 Interface

Nous avons commencé l'analyse de l'interface en partant des exigences fonctionnelles et avons créé un storyboard⁷ d'abord simple sur un tableau, puis plus complet, sur ordinateur.

Celui-ci nous a permis de lier les différentes vues entre-elles et de les lier aux différents profils. Afin de lier correctement les fonctionnalités aux vues, nous sommes passés au format papier, grâce à l'aide de monsieur Bouillon. Chaque page représentait une vue, elle-même représentée par un code suivant une convention. Pour chaque page, nous avons regardé quelles fonctionnalités métiers étaient utilisées : lesquelles devaient ouvrir une modale (*création ou édition*), lesquelles devaient ouvrir une nouvelles vue et lesquelles appelaient cette vue.

Cette étape nous a permis d'améliorer et de finaliser notre storyboard.

Nous sommes alors passés à la conception de maquettes des pages importantes telles que la liste des consultants (*sensiblement identiques à celles des clients et des projets*), la page d'accueil (*Dashboard*),...

Ces maquettes⁸ ont d'abord été dessinées sur papier afin de nous faire une idée du rendu et de diviser les pages en blocs primaires tels que des tableaux, des fenêtres modales, des formulaires, etc. Une fois cela fait, nous sommes passés à une version de démonstration où des vues étaient codées en dur afin de pouvoir présenter notre vision de l'application que ce soit lors de la réunion de fin d'analyse, ou lors de la présentation du stage à l'école.

Realdolmen ReMaTo

Administrator

Dashboard

Clients list

Consultants list

Projects list

Reporting- Assignments

Advanced Tables

Import .csv files

Manage users

Clients List

Filters (0 activated)

Show 10 entries

Search:

Reset all filters

Name	Region	Account Manager
ABSENCE MALADIE	WALLON	WALLON
ACTIVITÉS PRESALES	WALLON	WALLON
AGC	WALLON	MAJD SEGHRUCHNI
AGRIPHAR	WALLON	GREGORY STEYLEMAN
ARISTA	WALLON	WALLON
AVIQ	WALLON	WALLON
BAXALTA	WALLON	SEBASTIEN BRILMAKER
BELFIUS	WALLON	WALLON
BIM - IBGE	WALLON	GREGORY STEYLEMAN
CGT - COMMISSARIAT GÉNÉRAL DU TOURISME	WALLON	MAJD SEGHRUCHNI

Showing 1 to 10 of 58 entries

Previous 1 2 3 4 5 6 Next

Copyright © 2018 Realdolmen. All rights reserved.

Version 1.0b5.0

FIGURE 10 – Version finale de la vue de la liste des clients

7. Voir la page 6 dans les **Annexes**

8. Voir la page 9 dans les **Annexes**

5 Conception

5.1 Organisation de la phase de développement

La phase de développement s'est étalée sur une période de six semaines.

La méthode de gestion du projet étant la méthode Agile (*pas tout à fait rigoureuse*), chacune de ces six semaines représente un sprint. Voici l'organisation d'un sprint.

Premièrement, nous planifions le prochain sprint (***sprint planning***), d'abord entre nous, puis lors de la réunion de fin de semaine avec monsieur Sprio et, souvent, avec monsieur Bouillon. Lors de cette phase, nous prévoyons une ou plusieurs fonctionnalités pour la fin du sprint. Celles-ci pouvant être plus ou moins importantes et comprendre d'autres fonctions plus petites et explicites. Par exemple, la gestion des régions conduit à leur récupération depuis la base de données (*passage par les différentes couches de l'application*), mais il faut aussi pouvoir les modifier et les supprimer.

Deuxièmement, nous effectuons un **daily scrum**⁹ le premier jour du sprint. Nous divisons la ou les fonctionnalités en parties atomiques que nous écrivons sur des Post-It® placés sur un tableau divisé en trois parties : **À faire**, **En cours** et **Fait**. Chaque jour au matin, nous repassons rapidement sur ce qu'il restait à faire et ce qui était en cours.

Troisièmement, le vendredi nous finalisons le sprint, préparons la présentation pour le **sprint review**¹⁰ et nous planifions le prochain sprint.

Lors de la réunion, nous commençons par une revue de ce qui avait été prévu pour ce sprint. Nous passons alors à une démonstration de celui-ci qui présente la ou les nouvelles fonctionnalités et les modifications apportées d'après les remarques de messieurs Sprio et Bouillon. Nous passons alors à une partie de questions, que ce soient les nôtres ou celles de nos référents.

Nous faisons ensuite un bilan de l'avancement du projet et du sprint.

Puis, nous commençons le sprint suivant.

9. Voir *Daily scrum* dans le **Lexique**.

10. Voir *Sprint review* dans le **Lexique**.

5.2 Premier sprint

Lors du premier sprint, nous avons passé du temps à configurer et à apprendre à utiliser nos outils.

Nous avons mis notre dépôt git en place, et chacun d'entre-nous s'est acclimaté à sa couche. Par exemple, j'ai dû apprendre à effectuer des appels REST avec *ajax* et Arnaud a dû apprendre à utiliser *Hibernate*.

Le programme du sprint était, premièrement, de récupérer et de se familiariser à la sécurité développée par monsieur Boucher. Celle-ci devait permettre l'authentification automatique de l'utilisateur avec l'active directory de Realdolmen. Malheureusement, il n'a pas eu le temps de la développer.

La seconde fonctionnalité à développer pendant ce sprint était la création de la page d'accueil de l'application et l'ajout d'une table simple. On devait pouvoir y effectuer un **CRUD**¹¹. La table choisie a été celle contenant les régions.

Le sprint review a été repoussé au mardi suivant, car l'horaire de monsieur Sprio ne permettait pas de la faire le vendredi. Le début du second sprint a alors été repoussé au mardi après-midi, ce qui n'était pas un cadeau.

5.3 Deuxième sprint

En effet, ce sprint a été assez court : il a débuté le mardi après-midi et s'est terminé le vendredi à midi.

Il consistait en l'ajout des vues listant tous les clients, consultants et projets, ainsi qu'en l'ajout des vues *personnelles* des clients, consultants et projets.

Ce sprint a été très compliqué pour moi vu le travail à effectuer dans le temps imparti. De plus, ma partie dépendant du back-end, j'ai dû attendre qu'il soit fonctionnel pour pouvoir travailler réellement sur le code du sprint. Mes soirées ont été bien occupées.

Paradoxalement à tout cela, nous avons, quand même, eu le temps de débiter la vue liée aux assignements de consultants. Un autre plus de ce sprint : mes collègues ont eu le temps de mettre à jour l'UML de la base de données et ont rajouté des contraintes sur tous les champs de la base de données.

Pour régler mon problème de dépendance envers le back-end ou, du moins, le diminuer, monsieur Bouillon m'a proposé de travailler avec des données codées en dur ou de me baser sur un fichier contenant des données de tests.

Grâce à cela, le développement du front-end a été plus simple et plus rapide.

11. Voir *CRUD* dans le **Lexique**.

5.4 Troisième sprint

Ce sprint était important, car sa fonctionnalité phare, qui consistait à la prise en charge des assignements, était liée aux tables du deuxième sprint.

En effet, un consultant est assigné à un projet qui, lui-même, est lié à un client. On peut donc créer un assignement sur un projet depuis un consultant ou en créer un sur un consultant depuis un projet.

Nous avons aussi effectué quelques modifications et ajouts tels que la navigation dans les tableaux à l'aide du clavier et la vue de l'occupation du consultant sur 4 mois.

La fin de ce sprint a marqué la moitié de la phase de développement ainsi que la moitié de notre stage.

De plus, nous avons passé une interview avec le RH de l'entreprise le vendredi après-midi. Le but était triple : faire connaissance avec lui, donner notre avis sur le stage et, si nous le voulions, postuler pour un emploi en tant que développeur Java junior, ce que j'ai fait.

5.5 Quatrième sprint

Lors de ce sprint, ce sont de petites fonctionnalités qui ont été ajoutées.

En effet, j'ai ajouté un breadcrumb¹² dynamique et la vérification de la majorité des champs de formulaires à l'aide d'expressions régulières.

Du côté du back-end, Pierre a ajouté les calculs concernant l'occupation des consultants, Arnaud a commencé l'importation des données depuis un fichier `.csv`, et Kevin a commencé une ébauche des profils de sécurité à l'aide de *Spring security*.

De nombreuses modifications ont été effectuées dans le front-end.

Par exemple, tous les placeholders¹³ des champs de formulaires ont été retirés pour plus de lisibilité, l'encodage des dates doit être possible de manière manuelle et l'état des tableaux doit être gardé en mémoire.

De plus, toutes les tables « *simples* » contenant les données de références ont été ajoutées.

12. Voir *Breadcrumb* dans le **Lexique**.

13. Voir *Placeholder* dans le **Lexique**.

5.6 Cinquième sprint

Lors de ce sprint, Kevin et moi avons ajouté la sécurité dans le back-end comme dans le front-end.

Pendant qu'Arnaud continuait le développement de l'importation des données, j'ai ajouté une page qui lui était dédiée. Quant à Pierre, il a continué de travailler sur les calculs liés aux prix et aux marges.

Afin de préparer la troisième phase du stage (*Tests & bugs fixing*), nous avons mis l'application sur un serveur dédié aux tests. C'est principalement Kevin qui s'en est chargé.

De mon côté, il a été question de régler des bugs et de modifier des fonctionnalités pour accéder aux demandes de monsieur Sprio.

Pour terminer, cette semaine a été très chargée.

Nous sommes allés au restaurant, mardi soir. C'était une réception organisée en l'honneur de la fin de l'année fiscale. Il était conseillé de s'y rendre. Les responsables et le personnel de Realdolmen y étaient conviés. Cela s'est déroulé à Waterloo.

Dès le mercredi matin, nous partions, toute la journée, à Huizingen pour passer des interviews techniques. Moi, j'ai été interrogé en anglais. Certains candidats l'étaient en français. Cela s'est très bien passé pour moi.

Quant à vendredi, nous avons présenté notre stage à M. Tricarico au matin, et, l'après-midi, nous avons passé le troisième round d'interviews avec M. Sprio.

5.7 Sixième sprint

Lors du dernier sprint, la perte de temps, liée à la mise en place de sécurité par nous-même, s'est ajoutée à l'absence d'Arnaud qui était malade et à la perte du lundi de Pâques.

De plus, Pierre a dû préparer la phase de tests, ce qui lui a laissé moins de temps pour travailler sur la finalisation du sprint.

Nous avons alors dû faire un choix sur des fonctionnalités que nous ne développerions pas. La sécurité étant importante et déjà implémentée, nous avons donc décidé, avec nos référents, des profils qui seraient disponibles dans l'application. C'est ainsi que les profils disponibles sont : *Administrateur*, *Resources manager* et *Account manager*. Nous avons alors retardé la mise en place de la gestion des utilisateurs, des alertes et des filtres. Les requêtes ont, elles, été abandonnées.

5.8 Résultat

Le résultat de la phase de développement a été positif.

Nos deux référents, ainsi que monsieur Boucher, étaient satisfaits de notre travail.

Monsieur Boucher a déclaré qu'il ne s'attendait pas à ce qu'on réussisse à développer l'application en deux mois et a été agréablement surpris du rendu final.

Monsieur Sprio n'attendait qu'une chose : pouvoir tester l'application.

Quant à monsieur Bouillon, il était fier de notre évolution depuis le début du stage.

Le fait d'avoir un sprint par semaine était enrichissant dans le sens où nous devions apprendre rapidement, mettre en place ce que nous apprenions et régler les éventuelles erreurs que nous commettions.

Le point négatif a été que le temps était bien souvent court et qu'il fallait terminer le travail à la maison pour ne pas prendre de retard sur la planification.

En outre, ce rythme était fatigant, surtout que, pour ne rien arranger, nous avons tous été malades pendant plusieurs semaines, mais, à part Arnaud, nous étions présents au travail.

6 Tests & bugs fixing

6.1 Organisation

Lors du dernier sprint, Pierre a préparé cette phase de tests en listant toutes les fonctionnalités et vues à tester, dans un fichier *Excel*. D'après les remarques de monsieur Bouillon, Arnaud ajouté une feuille au fichier. Celle-ci contient les bugs rencontrés par les testeurs, leurs difficultés, une explication si possible, qui est responsable de sa correction, son état (*corrigé ou non*) et le numéro de la version dans laquelle le bug a été trouvé et dans laquelle il a été réglé.

J'ai alors été nommé responsable du numéro de version, par mes collègues. Chaque semaine, le numéro de version a été incrémenté. Et, pour chaque nouvelle version installée sur le serveur de test pendant la semaine, le numéro de sous-version était incrémenté.

Ainsi, nous avons débuté la phase de tests à la version **v1.0b1.0**, et, lorsqu'Arnaud et moi avons eu fini notre stage, la version était **v1.0b4.2**.

La première semaine, nous avons eu un testeur attitré, monsieur Lefranc, un consultant de Rsud. De nombreux petits bugs et changements à apporter ont été découverts par lui, mais aussi par nous, car nous effectuions aussi des tests de notre côté.

Il a alors été convenu que, chaque mercredi après-midi, nous allions effectuer un point avec nos référents.

Lors de ce point, les bugs qu'ils rencontraient nous étaient communiqués. C'est aussi à ce moment que de nouvelles fonctionnalités ou modifications nous étaient communiquées.

Entités/Vues	Description du problème	Bloquant	Complexité	Personne en charge	Test développeur concluant ?	Validation du testeur ?	Version de l'application où le problème apparaît	Version problème corrigé
DashBoard	1. Nombre de consultants, préciser actif/non-actif?			Pierre	Oui	Oui	V1.0b1.0	1.0b2.3.1
	2. 9/8 incohérent, à inverser							
	3. Pas de mise à jour du graphique							
Assignment	Pas de données dans la table et si créé par admin, non visible par RM			Kevin	Oui	Oui	V1.0b1.0	V1.0b1.1
Assignment	A la création, pas de contrôle sur projet ???			Arnaud	Oui	Oui	V1.0b1.0	V1.0b2.1

FIGURE 11 – Aperçu du fichier Excel contenant les bugs remontés

6.2 Nouvelles fonctionnalités

Lors de la première semaine, nous avons implémenté la gestion des utilisateurs, les filtres ainsi que l’export des données dans un fichier `.xlsx`.

Lors de la deuxième semaine de tests, les alertes ont été rajoutées, et le développement de l’import des données depuis un fichier `.csv` a été terminé.

De mon côté, l’ergonomie générale de l’application a été améliorée en ajoutant : des messages personnalisés lors de la suppression d’éléments, des repères visuels sur les entrées utilisateurs et sur les filtres, etc.

J’ai ensuite rédigé le guide utilisateur¹⁴ que j’ai illustré par des captures d’écran. C’est aussi cette semaine que Kevin a amélioré le script d’installation de l’application sur le serveur de tests. L’une de ces améliorations vient de moi : lors de l’exécution du script, le fichier lié à la base de données du serveur est créé alors que les fichiers utilisés pour la base de données locale sont supprimés. L’autre point important de cette mise à jour est que le code est directement récupéré depuis le dépôt git et exécuté sur le serveur. Avant, il fallait passer par une connexion FTP pour envoyer le fichier `.jar` contenant l’application sur le serveur.

Pendant la troisième semaine, ce sont principalement des petits bugs et petites fonctionnalités sur lesquels nous nous sommes focalisés.

Seul changement notable de mon côté, j’ai rédigé des fichiers **README**¹⁵ et un fichier **CHANGELOG**¹⁶.

Pour notre dernière semaine, des fonctionnalités indispensables ont été rajoutées. Ces fonctionnalités comprenaient : la suggestion du prix de vente d’un consultant d’après les marges à engranger (*selon son expérience et la technologie qu’il utilise*), l’affichage de ces marges et de celle effectuée lors de l’ajout d’un prix de vente¹⁷,... Cette semaine s’est ensuite focalisée sur la rédaction de la documentation du front-end et du back-end. C’est à ce moment que j’ai intégré le guide utilisateur au site. J’ai aussi expliqué le fonctionnement de ma partie à Kevin et à Pierre, car ils n’y avaient presque jamais touché.

Après notre départ, les deux stagiaires restants se sont occupés principalement de la documentation technique et de la documentation du back-end. C’est à ce moment que la gestion des qualifications d’un consultant a été finalisée.

J’ai alors dû corriger, de chez moi, des bugs découverts dans ma partie qui leur était trop inconnue que pour savoir la déboguer.

J’ai aussi modifié la manière dont les appels REST étaient effectués afin que l’application fonctionne correctement sur le serveur de production.

14. Voir la page 17 dans les **Annexes**

15. Voir **README** dans le **Lexique**.

16. Voir *Changelog* dans le **Lexique**.

17. Voir la page 12 dans les **Annexes**

6.3 Documentation

Tout au long de la phase précédente, j'avais documenté la majorité du code *JavaScript*.

Lors de cette phase, je suis passé au niveau supérieur.

J'ai commencé par repasser sur chaque fonction afin de mettre à jour et d'étoffer la documentation ou bien d'en rajouter le cas échéant. J'ai aussi rédigé, en anglais, un guide destiné aux utilisateurs de l'application, écrit un **changelog** pour le front-end afin de garder un historique des changements et modifications apportées, et j'ai écrit un fichier **README** pour chaque dossier du front-end dans le but d'en améliorer la documentation générale.

Le guide utilisateur est disponible depuis l'application en passant par un raccourci clavier (**ctrl** + **enter**) ou par un bouton d'aide représenté par un point d'interrogation. Dans **Chrome** et les navigateurs basés sur lui, le guide s'ouvrira à la page idoine.

Du côté du back-end, Kevin a commencé la rédaction de la documentation technique, en français, lors de notre dernière semaine. Afin d'être complet, j'ai rédigé la documentation technique liée à la mise à jour du guide utilisateur ainsi qu'à sa mise en place dans l'application.

Arnaud, lui, a documenté certaines classes afin de donner un exemple de documentation aux deux stagiaires à qui il restait encore deux semaines.

7 Réalisation

7.1 Développement du front-end

Le début de la conception du front-end a commencé pendant la création de la démonstration codée en dur. Nous avons essayé plusieurs designs sur différentes vues d'après les maquettes papiers.

Lorsque la phase de développement a débuté, j'avais alors une base sur laquelle me reposer tout en apportant les modifications provenant des remarques de messieurs Sprio et Bouillon.

J'ai alors procédé à la création de la page d'accueil (*ou dashboard*) qui est longtemps restée à l'état de page codée en dur. C'est d'ailleurs Pierre qui s'en est occupé, plus tard, pour ajouter les appels REST affichant le nombre de consultants actifs, de clients, de projets ou encore l'occupation des consultants.

J'ai aussi ajouté la première page de l'application : celle utilisée pour la gestion des régions. Pour y avoir accès, j'ai tiré parti de la barre latérale gauche disponible dans *AdminLTE*. J'en ai profité pour ajouter la majorité des liens qui seraient utiles lors de l'ajout des autres pages.

Dans cette barre latérale, chaque lien est associé à une icône le représentant. Ces icônes sont réutilisées dans le dashboard pour les consultants, clients et projets.

La page de gestion des régions est la « *page mère* » de toutes les autres vues : le code utilisé pour la créer a été servi pour toutes les autres. De fait, j'ai appris à utiliser les appels *REST* au travers d'appels *ajax* en passant par *jQuery*.

Ensuite, j'ai ajouté six vues différentes. Ces vues sont similaires trois par trois, c'est-à-dire que les vues listant les clients, les consultants et les projets sont très proches que ce soit au niveau de la construction ou du fonctionnement, et les trois autres vues sont les vues personnelles des éléments sélectionnés dans les listes que l'on peut voir comme des *cartes d'identité*.

Le fonctionnement des vues « *liste* » étaient, à ce moment-là, identique à celui de la vue des régions.

J'ai alors opéré une *standardisation* du code de création des tableaux des listes et de celui lié à leur fonctionnement. Pour le passage des *listes* vers les *cartes d'identité*, j'ai alors ajouté une action permettant de récupérer l'élément sélectionné dans la liste et de passer à sa carte d'identité.

Pour cela, nous utilisons un menu déroulant permettant plusieurs actions : *Ajouter*, *Voir* et *Supprimer* un élément. Il fallait donc trois clics pour avoir accès aux informations d'un élément sélectionné. J'ai alors implémenté un double clic permettant le même résultat.

Finalement, le menu déroulant a été changé en une barre d’actions, constituée de boutons représentés par des icônes. Et, d’après la demande de monsieur Sprio, la navigation au clavier a été intégrée aux tableaux. Celle-ci s’effectue avec les flèches haut et bas, et la touche `enter` permet aussi le passage vers la carte d’identité.



FIGURE 12 – Barre d’actions simplifiée

Pour ajouter un élément, il faut cliquer sur le + vert, et une fenêtre modale s’ouvrira. C’est un formulaire de création de l’élément choisi. Les champs obligatoires sont marqués d’un astérisque rouge. Les champs sont aussi vérifiés par du HTML5 et par des expressions régulières que j’ai écrites moi-même. Celles-ci désactivent le bouton de sauvegarde si les valeurs des champs sont mauvaises (*champ obligatoire vide, mauvais format, caractères interdits,...*).

La carte d’identité des éléments ne propose, dans un premier temps, les données qu’en lecture seule. Pour modifier les informations, il faut passer par le bouton *Modify info*.

Au début, toutes les listes déroulantes étaient remplies à la main et, pour sélectionner une valeur par défaut, la valeur du champ lui était passée par une variable. Désormais, je passe par une fonction unique pour remplir et sélectionner les données dans les listes déroulantes, et c’est le numéro d’identification qui sert à la sélection de la valeur.

Grâce au découpage du code en différentes fonctions, il a été facile de rajouter, dans les cartes d’identité, un tableau pour lister les projets d’un client, et les assignements des consultants et des projets.

Cela a permis d’uniformiser rapidement le rendu des vues, ce qui améliore l’ergonomie.

Tous les boutons sont identiques que ce soit dans leur ordre ou dans leur couleur. Le bouton d’ajout est toujours à la gauche et de couleur verte, celui pour passer à la vue suivante est à sa droite et est bleu foncé, et le bouton de suppression est à droite et est rouge.

Tout au long du développement, de nouvelles fonctionnalités sont apparues et, par moment, le code a été dupliqué. J’en ai alors profité, plusieurs fois, pour améliorer le code, pour en faire des fonctions réutilisables dans différents cas et, bien sûr, pour écrire ou mettre à jour la documentation du code.

Vers la moitié de la phase de développement, j’ai restructuré tout le code et j’ai modifié l’architecture du front-end afin d’y voir plus clair et de le simplifier.

Le code lié aux vues des listes se retrouve dans un dossier `lists` et les vues *carte d'identité* sont dans un dossier nommé `views`.

Les fichiers ont, eux aussi, été renommés afin que leur nom soit logique. Par exemple, si je veux voir le code lié aux formulaires, alors je vais dans le fichier `forms.js`. Je veux travailler sur le code des tableaux? C'est certainement dans le fichier `tables.js` que je dois aller.

Grâce à cela, la création de toutes les tables avancées a été simplifiée. Ces tables contiennent les valeurs de références et ne sont accessibles qu'aux administrateurs. Elles sont toutes créées dans le fichier `advanced.js` et utilisent presque exclusivement le même code. Bien entendu, certaines tables utilisent plus de champs (*plus de colonnes dans le tableau*) et ont donc été créées différemment.

Il y a un endroit où l'ergonomie n'est pas la meilleure : les formulaires. En effet, monsieur Sprio a demandé que les champs des formulaires soient vides et donc sans *placeholder* comme requis pour une ergonomie optimale. Il en a été de même pour les listes déroulantes qui ne doivent pas, par défaut, avoir d'option sélectionnée, et elles doivent être vides.

Lors de la phase de tests, le testeur, monsieur Lefranc, en a été étonné, car cela n'était pas très clair.

Toujours dans l'ergonomie, j'ai ajouté, dans tous les formulaires et toutes les modales de confirmation, un champ pour les messages d'erreur. Il est caché par défaut et n'est affiché que si une erreur est levée (*champ incorrect, erreur renvoyée par l'appel REST*).

En fait, toutes les modales partagent une même structure dans laquelle j'ai inséré ce champ d'erreur.

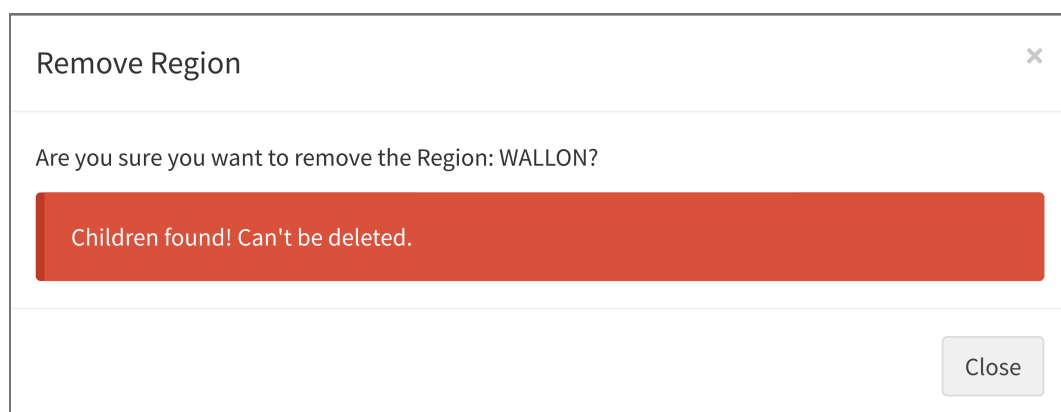


FIGURE 13 – Modale affichant un message d'erreur

Lorsqu'un administrateur ou qu'un directeur des ressources veut importer un fichier `.csv`, différents messages peuvent être affichés.

- Si le format du fichier est incorrect, un message d'erreur est affiché et le bouton d'exécution du script est désactivé.
- Si des erreurs sont détectées à l'importation, les lignes concernées ne sont pas importées, et l'erreur est affichée dans la page.
- Si les barèmes sont importés, un journal (*log*) est affiché à la fin de l'importation. Afin de montrer qu'il ne s'agit pas d'une erreur, j'ai décidé d'utiliser la couleur bleue de *Bootstrap* qui est utilisée pour afficher des informations.

Finalement, j'ai rajouté des filtres pour les tableaux les plus importants. Ceux-ci permettent de filtrer les tableaux d'après les listes déroulantes proposées. Un filtre différent permet également de cacher ou d'afficher des colonnes d'après les besoins de l'utilisateur.

Puisque les filtres restent activés tout le long de la session de l'utilisateur, j'ai décidé de les cacher par défaut, et d'afficher le nombre de filtres activés sur la page.

En outre, j'ai ajouté un bouton afin de réinitialiser tous les filtres ainsi que le champ de recherche intégré au tableau.

L'une des exigences fonctionnelles était de pouvoir exporter les données au format *Excel*. Pour garder une certaine logique d'utilisation, j'ai pris en compte les différents filtres lors de cet export.

8 Le relationnel

8.1 Communication des informations

La majorité des informations nous est communiquée de vive voix que ce soit lors des réunions du vendredi ou lorsque nos référents viennent nous voir dans notre bureau. C'est ainsi que, lors de notre journée d'accueil, le 6 février, l'objectif du stage nous a été communiqué.

Au cours de la phase d'analyse du projet, nous avons découvert différents besoins que nous devons prendre en compte (*création d'un breadcrumb, choix des rôles d'utilisateurs de l'application,...*) et d'autres nous furent dévoilées lors de réunions avec messieurs Sprio et Bouillon (*déplacement au clavier dans les tableaux, ajout d'affichages en accordéon,...*).

À ce moment, nous avions une à deux réunions ou points par jour, que ce soit avec monsieur Sprio, Bouillon ou les deux.

Le dernier jour de la phase d'analyse, nous avons travaillé avec monsieur Boucher qui nous a présenté le fonctionnement du framework *Spring* et d'*Hibernate*.

Pendant la phase de développement, c'est nous qui choisissons les objectifs des sprints. Nous les faisons valider lors de la présentation de fin du sprint précédent. Durant cette phase, nous avons continué de recevoir des conseils venant de monsieur Bouillon qui nous rendait souvent visite dans notre bureau.

Lors de la phase de *tests & bugs fixing*, les changements à effectuer pouvaient venir de différentes manières :

- au travers du fichier Excel mis à leur disposition ;
- par mail, le plus souvent depuis Huizingen ;
- de vive voix dans notre bureau ou dans le leur ;
- ou encore lors des réunions prévues à cet effet.

8.2 Difficultés

Ce stage m'a semblé bien équilibré.

Le fait d'avoir mis une équipe de quatre personnes sur ce projet le rend plus facile. En effet, nous n'avons eu que six semaines pour développer l'application web et, pour un seul développeur, le développement aurait été trop compliqué et trop long. Le choix de l'architecture est un autre facteur de diminution du niveau de difficulté du stage. Elle est composée de cinq couches dont l'une, la couche **Database**, est créée et rendue disponible par une autre, la couche **Repository**.

Grâce à cela, chaque développeur travaille sur sa couche, ce qui simplifie le développement vu que chacun *se spécialise* dans sa couche.

De plus, si quelqu'un rencontre un problème, les autres peuvent l'aider.

En conclusion, le stage a une difficulté moyenne grâce au travail d'équipe et à la phase d'analyse. Le plus compliqué a été d'avoir un bon rythme lors de la phase de développement. En effet, les sprints d'une longueur d'une semaine nous ont obligés à avoir un rythme soutenu, mais fatigant. Et le fait d'être malades pendant plusieurs semaines ne devait pas nous empêcher de terminer le développement.

Pour finir, les réunions de fin de sprint nous ont permis d'avoir un rythme soutenu et de recentraliser les objectifs.

Par contre, tout n'a pas été sur des roulettes.

Nous avons rencontré une grosse complication : notre support technique ponctuel, monsieur Boucher, n'était que très rarement disponible, même sur Slack, qu'il nous avait recommandé afin de pouvoir le contacter en tout temps.

Ainsi, la sécurité et la démonstration de la mise en place de rôles dans l'application, promises le 6 février (*et à nouveau le 23 février*), ne nous sont jamais parvenues. Pour le back-end, de nombreuses questions lui ont été posées, mais il n'y eut que peu de réponses.

Nous nous sommes donc débrouillés sans lui.

Nous avons rencontré un autre *accroc* lors du démarrage de la phase de développement. Il nous a fallu mettre en place notre environnement de travail (*git*, *Bitbucket*, *IntelliJ IDEA*,...) ainsi que commencer à développer.

Nous avons donc eu un peu de mal au démarrage, surtout en fin de sprint quand le front-end a dû attendre que le back-end fonctionne.

En effet, une fois celui-ci fonctionnel, le front-end a pu être développé d'après les valeurs retournées, et à retourner, en passant par la couche REST.

Les autres semaines, l'environnement de travail étant installé, nous nous sommes organisés afin que le problème n'apparaisse plus.

8.3 Intégration dans l'entreprise

Dès janvier, nous avons bien été intégrés à l'entreprise.

Nous avons, en effet, été invités au drink d'inauguration des nouveaux bureaux de Realdolmen Sud. C'était le soir, et nous y avons participé

Nous avons ensuite assisté à la présentation du travail d'un an de monsieur Russo, consultant de Rsud.

Le 27 mars, nous faisons même partie de la soirée *Kick-Off* de l'entreprise qui a marqué le début de la nouvelle année fiscale 2018/2019.

Afin d'égayer nos journées, j'ai amorcé un petit jeu. Celui-ci impliquait la découverte d'un nouveau mot par jour pour le mois de mars.

Le jeu a tellement bien fonctionné que les autres stagiaires, monsieur Bouillon et d'autres consultants se sont pris au jeu et qu'il a été prolongé jusqu'à la fin du stage.

Ainsi, nous comptons une liste de près de deux cents de mots tels que *apocope*, *aprosexie*, *bonace*, *coquecigrue*, *difficultueux*, *maritorne* et *ploutocrate*.

9 Conclusion

9.1 Conclusion du projet

Le but du stage était de développer une première version utilisable d'un outil de gestion de ressources pour Realdolmen Sud afin de faciliter le travail de monsieur Sprio, Delivery Manager de Rsud.

En effet, jusqu'à présent, il utilisait un fichier Excel qui ne proposait pas de contrôle à l'encodage, de données de référence et dont la gestion était lourde.

Nous y sommes arrivés, et j'ai même écrit un guide et des fichiers *README* pour plus de rapidité et de facilité.

L'équipe en charge du développement de l'outil était composée de quatre stagiaires : Kevin et Pierre, deux élèves de la haute école Condorcet , et Arnaud et moi qui venons du campus technique de la HeH.

L'entente au sein de l'équipe a été excellente tout au long du stage. Nous nous complétons très bien vu que nous avons chacun un domaine bien précis sur lequel travailler. Elle était nécessaire puisqu'à la fin, chaque partie dépendait des autres. Cette bonne entente a facilité les discussions et choix à prendre tout au long de celui-ci.

Notre outil propose une meilleure productivité ainsi qu'une meilleure maîtrise des données grâce à l'utilisation de formulaires permettant la vérification des données ainsi que l'enregistrement de données de référence au sein de l'application. Celles-ci empêchent l'encodage erroné d'un même élément sous deux noms différents comme il était facile de le faire précédemment. En effet, les données de références ne peuvent être encodées que par un administrateur. Une fois cela fait, les éléments à créer ou à modifier pourront tirer parti de ces nouvelles données de référence au travers de listes déroulantes.

Cost: *

PRDAOF, level 4 - 50 €
PRDAXP, level 3 - 50 €
PRDAXP, level 4 - 50 €
PRDAXP, level 5 - 50 €
PRDAXP, level 6 - 50 €
PRDBA, level 3 - 50 €

FIGURE 14 – Liste déroulante

L'administrateur peut, bien entendu, importer des données dans l'application depuis un fichier `.csv`. Lors de cet importation, les données sont vérifiées afin d'éviter des erreurs.

Une fois les données importées, l'utilisateur connecté y a accès en passant par des vues ergonomiques, une amélioration sans comparaison par rapport au fichier Excel. D'abord, l'utilisateur accède à la liste des clients, des consultants ou des projets affichés dans un tableau dynamique. Celui-ci permet de trier chaque colonne et d'y faire des recherches sur tous les champs. Des filtres sont aussi accessibles et les données peuvent être facilement exportées au format Excel (`.xlsx`) en un clic. Ensuite, un élément peut facilement être ajouté ou supprimé en passant par la barre d'actions située au-dessus du tableau.



FIGURE 15 – Barre d'actions

Pour avoir plus d'informations sur un élément, il suffit d'effectuer un double clic dessus ou de le sélectionner et d'appuyer sur **Enter**.

L'utilisateur se retrouve alors sur la page personnalisée de l'élément. On y retrouve ses données ainsi que les projets qui lui sont liés pour un client ou un consultant, et les consultants qui lui sont liées dans le cas d'un projet.

Comme précédemment, pour avoir plus d'informations, il suffit de sélectionner l'élément voulu et de double cliquer ou d'appuyer sur **Enter**.

La vue d'un assignement est différente, car elle n'affiche que les données de celui-ci.

Un autre avantage de notre outil : sa portabilité. De fait, il est disponible quel que soit l'appareil utilisé étant donné qu'il s'agit d'une application web. Elle a été principalement pensée pour un usage depuis un ordinateur, mais s'adapte aux écrans des tablettes et smartphones. Pour l'utiliser, il suffit d'être ajouté par l'administrateur et de se trouver soit sur le réseau de Realdolmen, soit d'utiliser leur VPN¹⁸.

Afin d'être utilisable par le plus grand nombre, nous avons décidé que l'application serait en anglais. En effet, nous parlons tous français à Rsud et, la majorité des personnes travaillant chez Realdolmen est néerlandophone.

18. Voir *VPN* dans le **Lexique**.

Pour terminer, voici quelques fonctionnalités qui améliorent le travail de l'utilisateur par rapport à l'utilisation d'Excel.

Premièrement, il est possible d'afficher un planning prévisionnel s'étalant jusqu'à douze mois dans la vue listant les consultants.

Ensuite, lors de l'enregistrement du nouvel assignement d'un consultant, une proposition sur le prix de vente est affichée d'après son prix d'achat et les marges qui lui sont liées. Si le prix de vente est rentré au clavier, sa marge est aussi calculée afin que l'utilisateur puisse rapidement savoir s'il doit négocier ou non pour un nouveau prix.

Troisièmement, lors de la suppression d'un élément, il y a une vérification pour savoir si l'objet peut être supprimé ou non. En outre, la suppression demande une confirmation afin d'éviter les erreurs.

Finalement, les différents profils permettent d'avoir différents rôles liés à des vues montrant plus ou moins de données et d'actions. Cela permet à l'administrateur de déléguer du travail tout en filtrant les informations visibles.

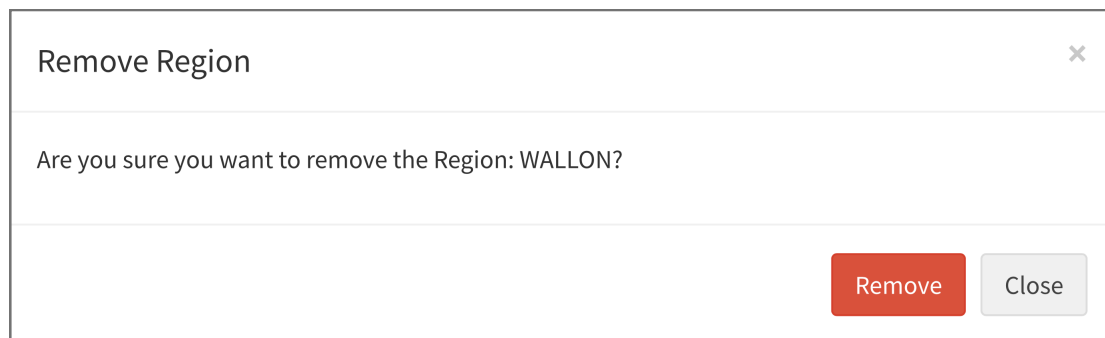


FIGURE 16 – Modale de confirmation

Malheureusement, nous n'avons pas réussi à implémenter toutes les fonctionnalités dans les temps. C'est ainsi que le nombre de profils disponibles n'est que de trois contre les six que nous avions prévus.

De plus, lorsqu'un utilisateur possédant des droits inférieurs voulait effectuer un changement ou créer un nouvel élément, l'administrateur aurait dû recevoir une requête. Celle-ci aurait pu être rapidement acceptée ou déclinée. Cette fonctionnalité a été mise de côté, vu qu'elle dépend des profils annulés.

9.2 Apports du stage

Pendant notre stage, nous avons appris à communiquer et à travailler en équipe comme les consultants font dans l'entreprise.

Nous avons aussi eu une part importante du stage dédiée à l'analyse du projet, chose à laquelle nous n'avons pas été fort préparés, ni habitués. Cette partie a été très importante, car elle nous a permis d'éviter certaines erreurs et de bien nous préparer à la phase de développement.

De plus, elle a permis à nos référents de Realdolmen Sud de voir que nous avions bien compris l'énoncé, et elle leur a aussi permis de nous recadrer sur des points importants que nous croyions moins importants.

En outre, chacun de nous a appris à utiliser de nouvelles technologies et pratiques qui nous ont permis de mener le projet à bien.

Pendant toute la durée du stage, nous avons dû effectuer des présentations et démonstrations de nos avancements. Ces présentations se sont révélées être de bons exercices pour nous permettre d'améliorer notre communication ainsi que notre gestion du stress.

Parallèlement à cela, le stage nous a permis de discuter avec des consultants experts dans les technologies que nous utilisions. Pour le Java, nous avons pu discuter, via Slack ou de vive voix, avec monsieur Boucher et, pour le front-end, j'ai eu l'occasion de parler plusieurs fois avec monsieur Russo, expert en développement web. En outre, monsieur Bouillon avait, lui aussi, de l'expérience dans les technologies web que j'ai utilisées et avait de nombreux conseils pour nous.

9.3 Conclusion

Pour conclure, je trouve que le stage a été très intéressant que ce soit du côté pratique ou du côté intégration dans l'entreprise. J'ai enfin vu à quoi pouvait ressembler le travail d'un développeur à plein temps, et quelles étaient les différentes étapes importantes d'un projet.

J'ai réalisé que le travail en entreprise exigeait un rythme très soutenu et que ce n'est pas parce que l'on quitte le bureau qu'il ne faut pas continuer chez soi pour atteindre les objectifs attendus.

En ce qui me concerne, j'ai décroché une belle récompense : un emploi de consultant chez Realdolmen. Ce qui n'était pas gagné d'avance vu mon caractère peu expansif et la concurrence.

Ce qui démontre que les bases vues à l'école et le courage peuvent venir à bout de tout.

Je sors de ce stage « grandi ».

Lexique

- **Breadcrumb** : fil d'Ariane, aide à la navigation indiquant la position dans un document.
- **Changelog** : journal des modifications.
- **CRUD** (*Create, Read, Update and Delete*) : désigne les quatre opérations de base pour la gestion de données : *Créer, Lire, Mettre à jour et Supprimer*.
- **Daily scrum** : la mêlée quotidienne est une réunion permettant de synchroniser et de planifier les actions des prochaines 24h.
- **Framework** : outil qui évite de réinventer la roue en réutilisant ce qui a déjà été fait, utilisé et validé par de nombreux utilisateurs. Cela représente, donc, un gain de temps, de fiabilité et une facilité de mise à jour.
- **Méthode Agile** : organisation de travail en cycles courts permettant aux développeurs de gérer un produit de manière souple, adaptative et itérative. Elle leur offre un retour rapide sur le code produit et leur permet de délivrer des solutions correspondant aux attentes des clients.
- **Placeholder** : texte affiché par défaut dans les champs d'un formulaire.
- **README** : fichier contenant des informations sur les autres fichiers du même répertoire.
- **Sprint review** : la revue du sprint est une réunion de fin de sprint pendant laquelle une inspection et une démonstration des changements sont effectuées.
- **UML** (*Unified Modeling Language*) : un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.
- **VPN** (*Virtual Private Network*) : un réseau privé virtuel est un système permettant de créer un lien direct entre des ordinateurs distants, en isolant ce trafic. On utilise notamment ce terme dans le travail à distance, ainsi que pour l'accès à des structures de type cloud computing.

