

# PROJ M1-FA-MIAGE

*CRM 365*  
*Bilan projet*

*Groupe :*  
**Antoine Craske**  
**Tarik Djebien**  
**Eric Sitraka Rakotobe**  
**Rudy Stienne**

## Table des matières

[Introduction](#)

[Bilan projet](#)

[Travail en équipe](#)

[Organisation](#)

[Technique](#)

[Perspectives](#)

[Rencontre PME](#)

[Prochaines étapes](#)

[Besoins non comblés](#)

[Augmenter la visibilité de notre application](#)

[Conclusion](#)

# Introduction

Ce document a pour but de faire un point sur le projet avant sa livraison finale. C'est l'occasion pour nous de prendre du recul sur le travail accompli afin d'identifier les points positifs et négatifs, les différences entre prévu et réalisé et également d'entrevoir les perspectives du projet.

L'équipe étant la base de réalisation du projet, nous allons commencer par analyser le fonctionnement de l'équipe projet, l'organisation effective et les points d'améliorations. Nous verrons ensuite comment s'est organisé la collaboration. Ce sera également l'occasion de présenter la performance de l'équipe. Nous ferons également un point sur la technique mise en place et les apprentissages de chacun.

Nous terminerons par l'avenir de ce projet en dehors de l'université en présentant le premier prospect et les prochaines étapes.

# Bilan projet

## Travail en équipe

L'équipe projet était composée de quatre membres : Antoine Craske, Tarik Djebien, Eric Sitraka Rakotobe, Rudy Stienne. Nous avons quasiment le même bagage universitaire, chacun ayant des connaissances spécifiques par ses missions en alternance et son expérience.

Une des première étape a été de partager la vision du projet entre les membres. Cette étape est cruciale à la bonne conduite du projet, elle permet de donner un cap à suivre. Trois semaines ont été nécessaires à la réalisation des spécifications et au partage d'informations sur le contenu du projet. Nous avons également défini les responsabilités de chacun ainsi que leur périmètre d'autorité. Les règles de bonnes conduites, les standards de documentation et de codage ont été définis afin de fixer le cadre d'exécution du projet.

Cette étape a également été l'occasion d'initier la collaboration entre les membres. Chaque tâche était évaluée par les membres, les remarques constructives étant nécessaires à l'amélioration de chacun. L'utilisation d'outils collaboratifs a été mise en place afin de faciliter la collaboration. Une salle de travail virtuelle disponible sur le Wiki mis à notre disposition nous a permis d'y mettre à disposition les documentations principales du projet. Ces pratiques et outils ont continués à être utilisés tout au long du projet.

Des points projet hebdomadaires ont été réalisés à l'université afin d'être ensemble lors de cette revue. Dès le départ, ces points ont été standardisés avec un ordre du jour à suivre : revue des tâches avec affectation des prochaines tâches, priorités pour la semaine à venir, décisions à prendre, ressenti sur le projet ou les tâches (ie feedback), question, remarques et revue de code.

La revue des tâches était l'occasion pour tous les membres de présenter leur avancement, leurs difficultés et de valider ensemble la réalisation d'une tâche. Une communication informelle avait bien sûr lieu durant les autres jours entre les membres de l'équipe. Une des règles mise en place et qui a été suivie était simple : anticiper les retard et ou risques de développements et prévenir au plus tôt l'équipe. Ceci dans un but d'adaptation au changement pour essayer d'anticiper plutôt que de réagir. Plusieurs fois cela nous a permis de tenir des délais en ayant un membre apportant son aide.

Des présentations de certains outils ou conseils ont régulièrement été mise en place afin qu'un membre puisse partager sa connaissance aux membres de l'équipe et ainsi améliorer la performance de l'équipe.

La revue de code était l'occasion pour les membres de comprendre les nouveaux développements, comment ils s'intégraient dans l'architecture existante, comment intégrer les modifications à leurs développements si nécessaire. Le but était que chaque membre conserve

une vue transversale de l'application et puisse si nécessaire modifier n'importe quelle partie du code sans avoir besoin d'autres membres de l'équipe.

L'organisation de notre équipe nous a permis de tenir les délais planifiés et de partager nos connaissances. La standardisation de la documentation et des points projets nous a permis d'améliorer au fil du temps les synergies au sein du groupe.

## Organisation

Ce chapitre a pour but de présenter le cycle de vie du projet et comment nous avons avancé sur ce projet.

Le découpage du projet était défini comme suit : prototype, lot 1 et lot 2. Chacun de ces lots a duré entre 7 et 10 semaines. Nous avons choisi d'orienter l'organisation du projet en mode agile, nous avons donc défini des sprints de trois semaines pour avancer sur les développements.

Cet intervalle de temps s'est révélé adapté à notre équipe, une livraison de fonctionnalité ayant été en moyenne réalisée en 1 semaine et demie, le cycle de sprint nous permettait de mesurer concrètement l'avancement du projet.

En début de projet, la vision à 6 mois du projet n'était pas possible car nous n'avions pas établi toutes les fonctionnalités à développer, nous n'avions pas validé l'architecture applicative et n'avions pas non plus d'idée de la vélocité de l'équipe. Après le premier jalon de livraison du prototype, nous avons une bien meilleure vision de l'application et des outils à mettre en place, un planning à plus long terme a donc pu être réalisé avec les dépendances entre les tâches.

En terme des types de tâches réalisées, le prototype était constitué de tâches d'analyse, de documentation, d'intégration d'outils et de frameworks. Cette première étape ne pouvait donc pas sembler productive mais est bien nécessaire à l'établissement d'une vision projet et d'un socle technique.

Nous avons au départ convenu que chaque membre découvrirait les technologies par lui-même en effectuant un développement. Ce choix s'est avéré mauvais car les développements étaient en retard. Nous avons modifié notre logique d'affectation des tâches en mettant une personne connaissant la technologie pour un premier développement. Cela permettait à cette personne de partager le fonctionnement à l'équipe lors des revues de code et laissait l'opportunité à un autre membre de l'équipe de développer avec une base de code fonctionnelle.

Les tâches pour livrer le premier lot étaient des spécifications détaillées de développement, de l'automatisation de tâches pour les tests, le packaging de l'application et bien sûr de développement. Les développements du premier lot étaient principalement des développements d'accès aux données et de services, peu de présentation.

Lors du dernier lot, nous avons à notre disposition le socle d'accès aux données, une partie des services finaux et une automatisation maximale des tâches d'intégration et de tests. Cela nous a permis de livrer un maximum de nouvelles fonctionnalités côté service et d'avoir une refonte complète de la couche de présentation. Si un autre lot serait réalisé par la suite, nous pourrions garder ce rythme avancé car tout est en place pour améliorer la vélocité de l'équipe : tests, automatisation et standardisation des pratiques.

La notion de tâche dans notre équipe projet comportait une analyse fonctionnelle, des développements avec les tests unitaires et d'intégration adéquats ainsi que l'utilisation de l'outil d'audit de code Sonar. Ce standard nous a permis d'assurer la maintenabilité de l'application après livraison d'une fonctionnalité en ayant à notre disposition des tests automatisés sur

toutes les fonctionnalités livrées et une qualité de code accrue par l'outil d'audit de code et les standards mis en place. Les standards nous ont permis de faciliter la compréhension du code par tous les membres de l'équipe.

Les standards de communication et le partage de la même vision d'un livrable nous ont permis d'améliorer la compréhension du projet et la collaboration.

## Technique

Ce projet a été également été l'opportunité de découvrir des technologies et d'améliorer notre conception d'architecture logicielle. Nous nous sommes basés sur des méthodologies et frameworks éprouvés en entreprise. Ceci afin de monter en compétences sur des savoir-faires recherchés mais aussi de comprendre réellement leur fonctionnement en les utilisant.

Pour les méthodologies, nous avons structuré l'architecture technique en 3-tiers : Accès à l'application par navigateur web, application déployée sur un serveur d'application et données stockées dans une base de données relationnelle. L'application en elle-même a également été structurée en trois couches : présentation, service et persistance. Cette structuration nous a permis de réutiliser des composants et de garder la flexibilité de changer une brique si nous le voulions. La couche présentation web est elle-même structurée en Modèle-Vue-Contrôleur (MVC) dans le même but de découpage des responsabilités afin de faciliter l'évolution de chaque brique sans impacter les autres.

Le choix du langage Java s'est avéré être un bon choix pour notre projet. Connu de tous, permettant la structuration de l'application en objets et couches applicatives et avec un grand nombre de frameworks qui ont pu nous faciliter le développement. Différents frameworks techniques ont été mis en place avec chacun un but précis. Leur apport après utilisation est détaillé ci-dessous.

Nous avons dès le début choisi d'utiliser Maven pour la gestion du cycle de vie de l'application et des dépendances. Cela nous a permis de standardiser les environnements de développement de l'équipe et surtout d'automatiser un maximum de tâches : déploiement sur serveur d'application, tests unitaires, tests d'intégration, packaging de l'application. Nous avons pu augmenter notre productivité en diminuant les risques d'erreurs humaines et la rapidité d'exécution de tâches répétitives.

Le coeur de notre application est basé sur des composants du framework Spring. C'est un framework éprouvé en constante évolution. Nous avons mis en place le conteneur Spring IoC pour gérer la dépendance de nos modules. La journalisation a été également mise en place avec Spring AOP avec le support AspectJ, nous permettant de centraliser la gestion des logs. La couche de présentation structurée en MVC a été développée avec Spring MVC nous permettant de faciliter son développement et sa maintenance. Nous avons également mis en place Spring Security pour l'authentification, l'accès aux fonctionnalités de l'application en fonction de rôles prédéfinies et la sécurisation des urls.

La couche web a été développée en JSP en respect des standards HTML5/CSS3. Plusieurs frameworks web ont également été utilisés pour réutiliser des composants existants. Pour éviter de faire du javascript pur et diminuer la gestion de compatibilité avec les différents navigateurs, nous avons choisi d'utiliser JQuery. Ce framework javascript a été intégré en milieu de projet, il était au départ prévu d'utiliser ExtJs mais en creusant son intégration, il s'est révélé trop intrusif au niveau de la couche présentation. Il fallait en effet modifier les contrôleurs et cela nous aurait rendu dépendant de ce framework. La structure des couches et le respect de standards d'architecture nous a permis de facilement utiliser JQuery. JQuery nous a également permis de rapidement mettre en place AJAX dans notre application, améliorant l'expérience utilisateur car seule des parties de pages sont rafraîchies. Aujourd'hui, nous avons la flexibilité de changer de framework javascript en modifiant uniquement la partie JSP.

Nous avons développé une partie reporting, nous avons donc eu besoin d'un framework gérant les graphes de type lignes et les pie chart. Jqplot a été choisi pour sa facilité d'intégration avec JQuery et d'intégration de données provenant des contrôleurs. Dans un même but d'amélioration de l'ergonomie, nous avons également intégré un framework pour améliorer les tableaux de données, dataTables. Ce framework s'intégrant également avec JQuery nous a permis de très rapidement mettre en place des tableaux avec tri, recherche et charte graphique.

Pour la couche de persistance des données, nous étions confronté au problème récurrent de la base de donnée relationnelle et du code Java objet. Dans un premier temps, Spring JDBC a été mis en place. Cela s'est avéré moins efficace que prévu car nous devons toujours gérer la transformation entre objet et relationnel, Spring JDBC gérant la récupération d'une connexion et la fermeture. Nous avons donc par la suite migré Hibernate gérant parfaitement cette problématique. Cela nous a permis de réduire le code à développer dans la couche d'accès aux données. Un autre avantage est qu'il nous a dispensé d'écrire du langage SQL directement, Hibernate gérant la portabilité de notre code d'accès aux données sur la plupart des modèles de bases de données.

Les méthodologies et frameworks mis en place nous ont permis de comprendre l'intérêt d'une structure applicative robuste et de l'intérêt de certains frameworks. Chaque membre connaissant un framework a pu partager aux autres membres sa connaissance sur le sujet permettant de faire progresser tout le groupe. Nous avons bien sûr été confronté à des soucis d'intégration ou de fonctionnement, mais le choix de frameworks éprouvés et supportés nous ont permis de rapidement trouver une réponse à nos problèmes.



# Perspectives

Notre application s'était dès le départ fixée comme cible les PME. Nous allons donc partager l'expérience toujours en cours avec une PME pour la mise en place de notre application, quelles sont les fonctionnalités encore à développer et les prochaines étapes à suivre pour le déploiement et la diffusion de notre application à plus grande échelle.

## Rencontre PME

Une fois l'application avec un socle utilisable et présentable, il fallait trouver un client potentiel qui exprimerait ses besoins effectifs et ses retours sur l'application. En effet, le cahier des charges et l'analyse fonctionnelle ayant été réalisé par l'équipe avec une mise en situation, il fallait confronter nos idées avec la réalité des entreprises.

Un membre de l'équipe ayant un contact de PME, nous avons pu partager les besoins de la société en terme de CRM. Cette PME est une société de gestion de patrimoine comportant une dizaine d'employés.

L'activité principale est le conseil en placements immobiliers, financiers, etc. Les employés sont donc principalement dédiés au contact avec le client, c'est pourquoi une solution de CRM était appropriée. Le SI de l'entreprise était simple, un fichier excel pour les clients et la comptabilité gérée avec un logiciel comptable standard installé sur deux postes de travail.

La première étape a été d'avoir un rendez-vous pour partager les besoins de l'entreprise en terme de CRM. Notre application a un but de fidélisation en automatisant des tâches basées sur la reconnaissance de comportements clients. Ce besoin a été confirmé par l'entreprise qui effectuait cette fidélisation de manière informelle en se basant sur le savoir-faire de ses employés.

Cela pose plusieurs problèmes, chacun a ses compétences, donc certains clients peuvent être perdus quand les employés posent des congés ou bien quittent l'entreprise. Notre application se révélait donc capable de partager le savoir-faire des employés, d'assurer une fidélisation constante et automatisait la décision d'une tâche à effectuer vers un client.

Notre application répondant à un besoin, la seconde étape fut de recueillir plus précisément les besoins de l'entreprise. Nous avons donc construit un jeu de données pour l'entreprise. Cela nous a permis de nous rendre compte que le moteur de génération d'actions de l'application gérait uniquement la moitié des cas. Etant organisé en agile, nous avons pu rapidement réorganiser les tâches de façon à faire évoluer notre moteur de génération d'actions rapidement.

Cette étape franchie, nous avons pu partager les besoins fonctionnels attendus par l'entreprise. Certains étaient prévus comme la possibilité de configurer les événements à récupérer et les actions à générer, la saisie d'un événement et le traitement des actions dans l'interface. En revanche, nous n'avions pas prévu l'administration de l'application en ligne : gestion des utilisateurs et des équipes. Nous avons donc incorporé ce besoin à nos sprints de développements, notre organisation nous permettant de répondre au changement.

Aujourd'hui, des actions sont encore nécessaires afin de rendre déployable notre application dans cette PME et d'autres. Ces prochaines étapes sont décrites dans le chapitre suivant.

## Prochaines étapes

Cette partie détaille les fonctionnalités manquantes et les points à clarifier pour valoriser notre application et la rendre utilisable dans des conditions d'entreprise.

### Besoins non comblés

Le coeur de l'application est quasiment complet pour être déployable dans la PME mais certains briques sont encore nécessaires.

Un des gros manque actuel est la documentation en ligne et un document de formation utilisateur, car la partie gestion du changement n'est pour l'instant pas gérée. Nous devons en effet prouver que notre application peut être déployable mais également facilement utilisable.

Un développement qui n'a pas pu être traité est également la migration vers notre application. Il faut en effet faire une migration des données clients de l'entreprise dans notre application. Cela sera géré par un outil simple et externe à l'application, facilement intégrable dans différents environnements. Ceci dans le but de garder le coeur de notre application dans la cible définie, et utiliser un outil externe lors de la mise en place car uniquement nécessaire lors de cette phase.

Un autre manque est la présentation de la configuration effective de l'application : quelles sont les événements et leurs paramètres, les règles de générations d'actions. Cela ne présente pas de difficultés techniques particulières, des pages sont à ajouter sur l'application.

A un niveau moins important, l'accès à l'application sur smartphone et tablette a été évoqué. Notre structure en couche nous permettant de développer une partie présentation dédiée au mobile branchée sur notre couche service, assurant la cohérence de l'application par réutilisation de la même couche applicative service et présentation.

Bien sûr, notre application ne pourra pas aujourd'hui répondre aux besoins de toutes les PME. C'est pourquoi notre application structurée et dotée de tests est plus facilement évolutive. Nous pourrions donc en fonction de différents besoins faire évoluer notre application. Nous pourrions également penser à une mise à disposition de l'application en mode SaaS (Software As A Service), permettant de limiter le frein de mise en place de serveurs dépendant fortement des PME.

Plus spécifiquement à la PME avec laquelle nous sommes en contact, nous devons clarifier l'intégration de l'application dans l'entreprise. En effet, l'entreprise disposant d'un SI simple mais éprouvé peut difficilement accueillir un serveur d'application avec une base de données. Il sera surement nécessaire de mettre en place un ou deux serveurs pour installer notre application et ainsi permettre aux employés d'y accéder. La solution d'installer les composants sur un poste client n'étant pas satisfaisante en terme de performances.

Nous avons au départ prévu de récupérer des événements sur les réseaux sociaux Facebook et Tweeter. La PME avec qui nous sommes en contact et celles que nous ciblons ne disposent pas pour la majorité d'un site web et il n'y a donc pas de sens à avoir une telle fonctionnalité.

## **Augmenter la visibilité de notre application**

Notre application est disponible en anglais et en français. Notre réseau de contacts avec des PME étant très limité, nous devons déployer des moyens pour faire connaître notre application sans nous limiter à la rencontre physique.

Un des premiers points à clarifier est la politique de l'application en terme d'utilisation. Rendons-nous notre projet open-source en mettant à disposition du support entreprise ? Complètement open-source ? Gratuit ? Des plugins payants ? Est-ce que notre projet devient propriétaire et implique la création d'une petite entreprise distribuant le logiciel avec des licences ? Quelle serait la politique de licence : à l'utilisateur, à la configuration machine ? Toutes ces questions n'ont pas encore été abordées ni décidées mais ce cadre devra être défini avant d'entreprendre d'autres actions.

L'augmentation de la visibilité se fera par la mise en place d'un site web à l'aide d'un CMS. Ce site présentera l'application en terme de ciblage, de plus-value et de politique d'utilisation. En parallèle, nous pourrons démarcher des PME physiquement. Un des points crucial sera de rendre notre application crédible aux yeux des entreprises, c'est pourquoi des premiers déploiements avec succès nous permettront d'augmenter la crédibilité du produit.

Toute l'équipe continuera cette aventure. Nous commencerons par combler les besoins les plus importants en mettant en place en parallèle les mécanismes pour faire connaître notre application. Seul l'avenir nous permettra de connaître les débouchés de ce projet.

## Conclusion

Ce projet a donc été l'occasion pour tous les membres de travailler en équipe et de développer de nouvelles compétences à la fois humaines, organisationnelles et techniques. Chacun a pu partager aux autres membres ses connaissances et ainsi faire progresser le groupe. L'organisation en mode agile et le respect de normes et standards tout au long du projet nous ont permis de répondre au changement et d'améliorer notre application. La maintenance de l'application est réduite ce qui nous permet de nous focaliser sur le développement de nouvelles fonctionnalités.

L'aventure CRM 365 n'est pas terminée, les actions engagées nous permettront de continuer son développement et nous verrons bien où cela nous mène, il est trop tôt pour anticiper l'avenir.

Je pense que chaque membre ressort de ce projet plus fort et cela justifie le fait que ce projet fût une belle opportunité pour chacun.

