



**CRÉDIT AGRICOLE
ALPES PROVENCE**

**DOSSIER DE SOUTENANCE
TITRE CONCEPTEUR
DÉVELOPPEUR D'APPLICATION**

Gestion des investissements dans les pays à risque

RNCP Concepteur développeur d'application

COLAS Jean Davidson

22/07/2021

La Plateforme - CAAP

SOMMAIRE

1. Introduction

2. Description et cahier des charges du projet

3. Gestion de projet

4. Analyse fonctionnelle

5. Analyse technique

6. Réalisation

7. Conclusion

INTRODUCTION

Présentation de l'entreprise

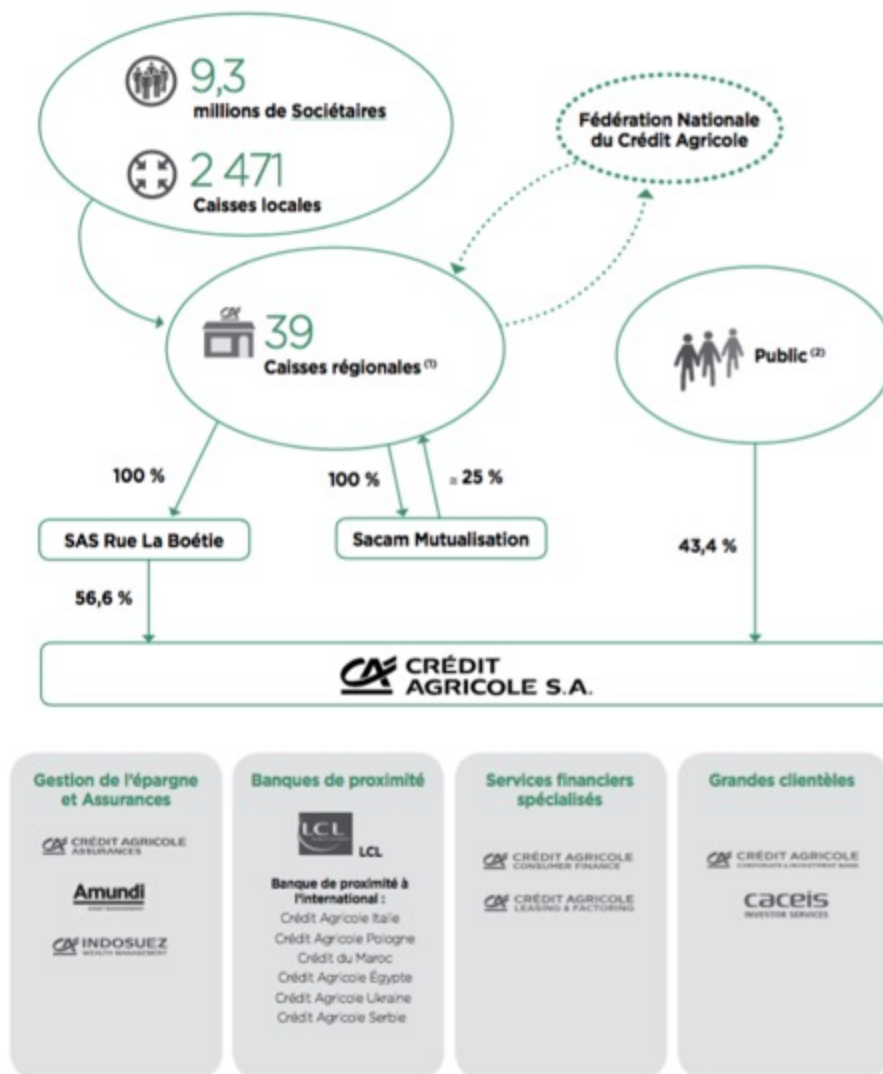
Le Crédit Agricole

Né en 1894, le Crédit Agricole est organisé sur une base mutualiste. La « banque verte » compte 10,9 millions de sociétaires, regroupés dans 2 410 caisses locales qui détiennent l'essentiel du capital des caisses régionales.

La Fédération Nationale du Crédit Agricole est l'instance d'information et de dialogue entre ces caisses régionales. L'échelon national, le Crédit Agricole S.A, dit CASA, est l'organe central qui veille à la cohésion du groupe et le représente auprès des autorités bancaires. Il s'agit d'une holding cotée en Bourse et détenue majoritairement par les caisses régionales.

À la fin 2020, l'enseigne compte 39 caisses régionales, 6 600 agences, 72 000 collaborateurs et 21 millions de clients particuliers, ce qui lui permet d'être le premier bancassureur de France.

Le groupe Crédit Agricole détient également LCL tandis que la banque en ligne BforBank est la propriété des caisses régionales.



Organigramme de la compagnie

La Crédit Agricole - Alpes Provence

Le Crédit Agricole Alpes Provence (CAAP) est une caisse régionale du groupe Crédit Agricole. Elle est active depuis 27 ans et est domiciliée à Aix-en-provence (13090). Son effectif est compris entre 2000 et 4999 salariés.

Project summary

Management and consultation of the list of countries at risk

Within the framework of the control of the risk when with the investment carried out by the fund outside the territory Crédit Agricole Alpes Provence is asking us to find a solution for managing files concerning countries at risk.

Their handling, historization and quick consultation with a view to improving performance for users and greater control of financial flows to other countries or regions of the world.

This is a new project knowing that previously no tools allowed the quick consultation of the list of countries at risk we must therefore already take into account the existing process constituting an excel file listing the countries declare at risk by the government and the fund, located in the internal server with agricultural credit and searchable via the employee interface partabt of this postulate we decide to set up a solution from the C # language.

This project will therefore be developed using C # language which I will learn at the same time and which is widely used in the world of web development and programming in general which will allow greater agility when maintaining the application, Entity Framework will be used to perform the database and migrations.

For the client side, it is the Telerik (Kendo) framework associated with Js (Jquery) which will be used in order to properly separate the database from the client interface.

My work on this project will therefore be to make the mock-up of the application, that of the database and then set up this one.

Subsequently I should create a layer system through the service and interface which will be the "pass-through" of the application, finally I would create an interface using Telerik (Kendo) Js (jQuery). It is therefore a large-scale project that will be implemented initially in the region, then probably at the national level.

This will involve certain constraints such as not exposing the data to the outside, allowing a rapid update of the database in order to be reactive to change

geo-policy and of course allow an improvement in processing times concerning the opening of loans to the customer who wishes it.

This project is carried out following the request from the Compliance & Credit departments and will be carried out as part of my work-study program at Crédit agricole Alpes Provinces within internal development and the "Digital Factory" service

DESCRIPTION ET CAHIER DES CHARGES DU PROJET

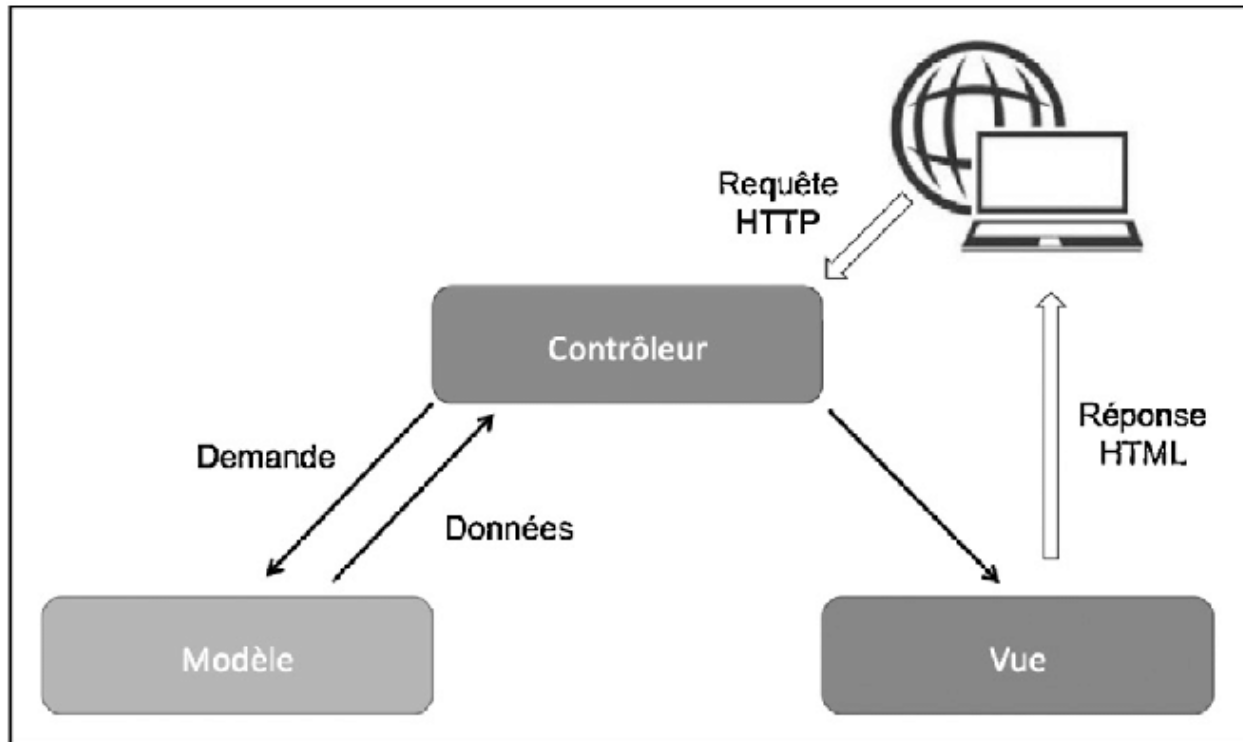
Analyse de l'existant

Au départ du projet tout est à faire et à construire ce qui pourrait être avantageux car cela m'a permis de choisir entièrement (avec l'aide de mon tuteur Nicolas PARIS) les outils qui seront utilisés pour réaliser la solution ainsi nous choisissons d'utiliser Figma pour la maquette Bdd et applications puis C# , EF6, MSMS et Telerik (Kendo) par la suite.

Architecture du projet

Le motif d'architecture logiciel choisi pour réaliser ce projet est le Modèle-vue-contrôleur ou MVC auquel nous ajouterons une couche de services.

Ce motif est composé de quatres types de modules ayant chacun leur responsabilité.



Motif MVC

Pour stocker les données du fichier , il nous faut évidemment une base de données relationnelle. Entity Framework est un framework orienté objet, qui nous permettra de manipuler des objets quand nous le pouvons. Par exemple, au lieu d'écrire des instructions SQL pour récupérer des enregistrements de la base de données, nous préférons utiliser des objets. Nous avons fait le choix d'utiliser EF6 pour générer ces objets, étant celui avec lequel nous étions le plus à l'aise.

Exemple de model

Les Modèles seront créer en

Exemple de vue

Les vues seront gérées par Telerik au travers de l'outil Kendo permettant l'ajout simple et rapide de vue au travers d'objet Json, Cette outil a l'avantage de permettre d'être très modulable quand à ses composants

Exemple de contrôleur

Les contrôleurs auront pour rôles de vérifier la bonne exécution des requêtes récupérer, lire les objets (Json) et renvoyer vers les vues "routes" au client

Exemple de service

La couche de services sera là pour réaliser le métier de notre application. Comme mentionné plus tôt, elle manipule les entités, appelle les web services, réalise la persistance des données en appelant les model nécessaires. Concrètement c'est le cœur de notre application.

Partie client

Cette partie de l'application était relativement terminée à mon arrivée sur le projet. Certains correctifs étaient cependant nécessaires. Je me suis basé sur son aspect graphique pour orienter la partie conseiller/administrateur de l'application.

Exemple de la partie client

Partie conseiller

Enfin, cette partie de l'application est celle où j'ai le plus travaillé. Hormis le tableau qui recense les dossiers clients en cours de traitement ou traités, rien n'était fait.

Exemple de la partie conseiller

Les utilisateurs du projet

Les utilisateurs de ce projet seront répartis en trois catégories et la solution doit rester simple d'utilisation, afin de faciliter l'appréhension, la manipulation et donc améliorer les process

Les échéances du projet

Début du projet

Comme mentionné plus tôt, ce projet démarre de zéro il commence donc au moment où la demande a été remontée soit en janvier

Date limite du projet

Pour ce projet, nous nous sommes fixé comme objectif de rendre disponible la nouvelle version de l'outil pour la rentrée 2021.

Contexte technique

Concernant les conseillers, la partie front-end qui leur est destinée doit être capable de fonctionner sur tout type de navigateur. La responsivité du site ici n'a que peu d'importance car l'outil sera utilisé en plein écran néanmoins le responsive sera appliqué au travers de Kendo et son système de Grid (proche de celui de Bootstrap ou Materialize)

Concernant les administrateurs leur matériel est standardisé et l'outil doit fonctionner sur google chrome et windows 10. La responsivité du site ici n'a que peu d'importance car l'outil sera utilisé en plein écran néanmoins le responsive sera appliqué au travers de Kendo et son système de Grid (proche de celui de Bootstrap ou Materialize)

Définition des entités

Comme mentionné plus tôt, EF6 est un framework orienté objet, nous aimons manipuler des objets quand nous le pouvons. Ainsi de nombreuses entités sont créées dans ce projet.

Si on devrait dégager les principales entités du projet, on pourrait le faire de la sorte:

- Pays : Un pays avec les code ISO qui correspondent
- Enrollment : une liste de dates appartenant à un pays .

Fonctionnalités attendues

Consultation

Les conseillers en agence doivent pouvoir consulter et identifier rapidement un pays à risque.

Gestion

Les gestionnaires doivent pouvoir consulter et modifier la liste des pays et leur enrôlement au sein d'un outil simple et facile à maintenir et à appréhender.

Dépose de fichiers

Le gestionnaire doit avoir la possibilité de mettre à jour la liste au travers d'un import. Afin de faciliter son expérience, il aura à sa disposition une zone prévue à cet effet ou il pourra faire glisser les fichiers nécessaires.

Gestion des administrateurs et conseillers

Le site doit disposer d'une section Administrateur qui permettra la gestion des fichiers et des entités.

Les conseillers eux pourront uniquement consulter la liste.

Historisation

Les Actions de chacun seront Historiser au sein d'un fichier Log réalisé à l'aide du package Nuget réalisé par les collaborateurs du service.

L'historisation des Pays se fera quand à elle au travers de ses enrollment visible de gestionnaire et signifier par une checkbox accompagné d'un message

Environnement de travail

Ce projet est réalisé en entreprise, avec l'ordinateur fourni par la société.

Il est réalisé sur l'éditeur de code Visual Studio et le serveur de développement sera quant à lui le serveur Windows R2 de l'entreprise.

Collaborateurs

- Product Owner: Christophe Brunel / Tailedet Lionel
- Tuteur : PARIS Nicolas
- Lead Developer: COLAS Jean Davidson
- Lead UI/UX: RAMALLI Aurelia

GESTION DE PROJET

En intégrant l'entreprise, j'ai commencé par renforcer mes compétences en C# car c'est une véritable découverte pour moi.

Pour cela j'ai réalisé deux semaine de veille technologique consacrer uniquement au fonctionnement de C# et EF6.

Puis j'ai mis en pratique ce que j'avais appris en répétant les tutoriels fournis dans la Doc Microsoft et en réalisant mes premiers projets comme "student" qui est une solution permettant la gestion d'un groupe d'étudiants.

Du moment où je me suis mis à travailler sur ce projet, je n'avais aucun autre projet en parallèle à l'entreprise, j'ai donc pu consacrer 100% de mon temps la bas sur ce projet.

</

Planning d'alternance / projet

Des points réguliers ont été faits avec les parties prenantes du projet à différents niveaux du développement afin de tenir compte des potentielles évolutions.

A l'heure où j'écris ce dossier, le projet est terminé et attend l'aval du Data Protection Officer avant sa mise en production

ANALYSE FONCTIONNELLE

Les acteurs

Suite à l'expression des besoins, 3 rôles ont été définis:

- Les conseillers
- Les gestionnaires
- Les administrateurs

Ces trois rôles auront des droits bien spécifiques à chacun, et aur

Les conseillers :

Les conseillers sont les acteurs principaux du projet. Ils doivent pouvoir consulter rapidement la liste et y retrouver un pays

Les gestionnaires :

Les

Les administrateurs :

Les administrateurs ont un rôle de gestion sur les conseillers. Ils doivent être en mesure de gérer les droits des conseillers et d'en ajouter ou de les supprimer. En plus de ces droits, ils auront accès aux mêmes choses que les conseillers, c'est-à-dire réaliser une

entrée en relation.

Séparation des domaines fonctionnels

La partie client

Ce domaine regroupe toute la partie concernant les clients. Il accueille leur création de compte ainsi que le formulaire récoltant les informations nécessaires à leur demande.

La partie collaborateurs

Ce domaine regroupe toute la partie concernant les collaborateurs. Il leur permet de gérer les demandes clients ainsi que la gestion des conseillers et de leurs droits.

La partie statistique

Intégrée dans un lot 2 d'amélioration , elle contiendra des outils de data visualisation permettant au service Parcours Clients-Collaborateur d'optimiser le processus.

Cas d'utilisation

Demande en autonomie

Le client se rend sur le site de façon autonome et complète le formulaire. Il fournit les documents nécessaires et attend d'être rappelé par un conseiller.

Un conseiller de l'agence que le client a sélectionné reçoit l'email de notification dans la boîte au lettre de l'agence et prend en charge le client.

Il valide ou invalide les pièces d'identité fournies par le client et un email lui est envoyé

pour le notifier de la décision du conseiller. Cette étape ne sera considérée comme terminée que si le client fournit les bon documents.

Une fois les documents validés, il contacte par téléphone le client et lui pose les questions nécessaires afin de lui proposer les options les plus adaptées à sa demande.

Cet appel terminé, il se rend sur le logiciel de gestion interne de l'entreprise et réalise les opérations nécessaires sur le SI de l'entreprise.

A la suite de ces opérations, le conseiller récupère sur le SI les contrats à envoyer au client et les transmet via l'outil.

Le client réceptionne les contrats et choisit de les signer ou non.

Les contrats signés sont renvoyés vers l'outil et le conseiller reçoit un email de notification lui signalant que son client a signé ses contrats.

Il se rend sur l'outil et vérifie les contrats, il contacte ensuite son client pour l'accueillir et lui présenter les solutions disponibles au sein de l'entreprise.

Demande téléphonique

Cette démarche diffère peu de la précédente. La différence se situe au niveau de la création de la demande d'entrée en relation. Dans ce cas, le client contacte le conseiller et lui formule sa demande. Le conseiller pré inscrit le client sur le site et pré remplit les informations de son client. Le client reçoit un email contenant un lien l'invitant à se rendre sur le site pour choisir un mot de passe. Une fois chose faite, il se connecte sur l'outil, vérifie les informations que le conseiller à saisie et les modifie au besoin. Enfin, il fournit les pièces justificatives nécessaires. Le reste du processus est le même que pour une entrée en relation en autonomie.

Parcours agence habitat

Ce parcours diffère des deux précédents. Certains de nos clients se servent de notre outil d'entrée en relation à distance pour réaliser des projets immobiliers.

Ces dossiers sont réalisés pour des personnes qui sont déjà clientes de la caisse régionale et ont généralement fourni leur document au préalable.

Les conseillers se servent de l'outil jusqu'à l'envoi des contrats électroniques et leur signature. Le processus est ensuite détourné sur un de leur outil interne.

Diagramme de navigation

Client

Landing

Redirection outil

Connexion -> Mes démarches

Inscription -> Formulaire -> Mes démarches

Formulaire de la démarche

Récapitulatif de la démarche

Collaborateur

Login

Gestion des dossiers -> Sélection d'un dossier

Validation des pièces justificatives

Démarches administratives

Envoie des contrats électroniques

Finalisation des démarches administratives

Finalisation du dossier et accueil du client

ANALYSE TECHNIQUE

Base de données

L'object-relational mapping: Doctrine

Pour comprendre comment fonctionne Doctrine ORM il faut savoir qu'il se compose de deux grandes parties:

- Doctrine ORM
- Doctrine DBAL

L'ORM dépend du DBAL et ce dernier s'appuie sur PHP Data Object.

PHP Data Object

PDO est une interface objet qui permet de manipuler une base de données. Il inclut diverses mesures de sécurité, comme la préparation des requêtes. C'est une interface d'abstraction sur lequel se base Doctrine DBAL qui est incluse dans PHP depuis la version 5.1.

Doctrine DBAL

DBAL est une surcouche de PDO. Elle lui ajoute des fonctionnalités comme l'ajout de fonctions qui permettent par exemple de lister les tables, les champs, le détail des structures.

Doctrine ORM

C'est une interface qui nous permet de mapper les différents éléments de notre base de données qui sont gérés par DBAL.

ORM nous offre la possibilité de faire correspondre les attributs aux champs de la base de données. Pour cela on utilisera un fichier de mapping au format YAML ou XML. Dans un fichier PHP nous avons aussi la possibilité d'utiliser des annotations.

Doctrine Query Language

Le DQL est un langage de requête orienté objet. Il est propre à Doctrine et peut être utilisé pour réaliser des requêtes à la place du langage SQL.

Voici deux exemples de tables que j'ai réalisé :

REQUEST :

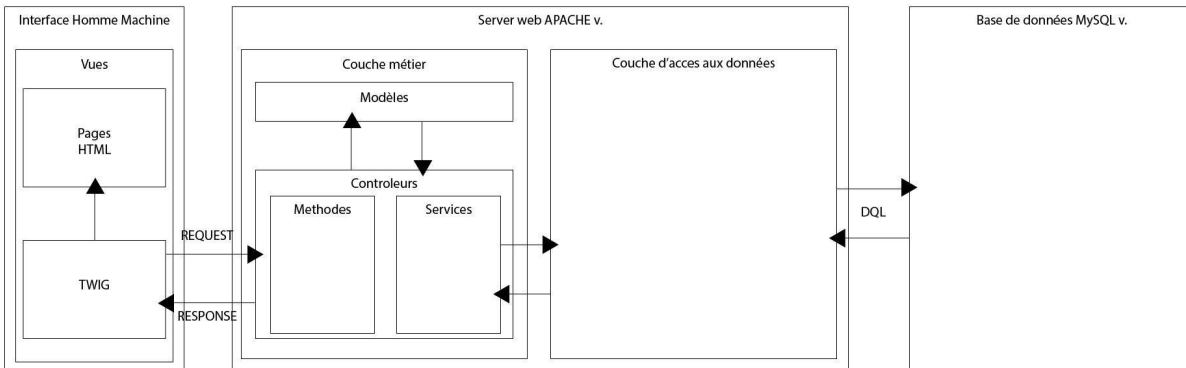
CUSTOMER :

Ci-dessous les script des repository générées par l'orm doctrine:

REQUEST :

CUSTOMER :

Architecture logiciel



RÉALISATION

Choix de développements

Choix des langages

Les langages que nous utiliserons pour réaliser ce projet:

1. *HTML*
2. *CSS*

3. *PHP*
4. *Javascript*
5. *SQL*

Choix des frameworks

Concernant les bibliothèques et framework que nous utiliserons:

1. *Jquery*
2. *Symfony*
3. *ReactJS*
4. *Twig*
5. *Bootstrap*

Logiciel et outils

L'IDE : PHPStorm

De l'avis de beaucoup, le meilleur IDE pour le développement d'applications web. J'ai eu l'occasion de l'essayer à mon entrée dans l'entreprise et je l'ai adoptée. Il est simple d'utilisation et facile à prendre en main pour quiconque possédant un peu d'expérience dans le développement.

Server web: Apache v.

Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde. Le nom officiel est Serveur Apache HTTP et il est

maintenu et développé par Apache Software Foundation. Il permet aux propriétaires de sites web de servir du contenu sur le web – d'où le nom « serveur web » -. C'est l'un des serveurs web les plus anciens et les plus fiables avec une première version sortie il y a plus de 20 ans, en 1995.

Base de données : MySQL v.

MySQL est un système de gestion de base de données relationnelles (SGBDR) basé sur SQL (Structured Query Language). MySQL fonctionne sur pratiquement toutes les plates-formes, y compris Linux, Unix et Windows. Il est entièrement multi-thread avec un noyau de threads, et fournit des API (Application Programming Interface) pour de nombreux langages de programmation, notamment C, C + +, Eiffel, Java, Perl, PHP, Python, et Tcl. MySQL est utilisé dans une large gamme d'applications, Le commerce électronique, les bases de données Web...

Arborescence du projet

Ci-dessous une explication sommaire des différentes couches de l'application, qui seront détaillées dans les exemples suivants.

Root

INSERT SCREENSHOT

Exemple de réalisation n°1 :

INSERT SCREENSHOT

Exemple de réalisation n°2 :

INSERT SCREENSHOT

CONCLUSION

Sur le respect du cahier des charges

Sur la gestion de projet

Sur l'avenir du projet

Concernant l'avenir du projet, un lot d'améliorations prévoit d'implémenter un suivi des statistiques en s'appuyant sur l'historisation présente sur le projet pour optimiser l'outil.

La connexion automatique des collaborateurs avec le Single Sign-On de l'entreprise est

aussi prévue, ainsi qu'une mise automatique des documents en Gestion Electronique des Documents (GED).