RAPPORT TECHNIQUE
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
DANS LE CADRE DU COURS GTI792 PROJET DE FIN D'ÉTUDES EN GÉNIE DES TI

Analyse statistique - Intervalles qualités - Robots de Fabrication CMM

JHOJANDRELLI CONGOTE

CONJ28038807

IGNACE BERTRAND TAKUPO CHENDJOU

TAKI24098600

NADIR HAMROUN

HAMN07027402

DÉPARTEMENT DE GÉNIE LOGICIEL ET DES TI

Professeur-superviseur

Alain APRIL

MONTRÉAL, 13 AVRIL 2017
HIVER 2017

REMERCIEMENTS

À toute l’équipe de la société Netsen Group, pour leur disponibilité et leur professionnalisme durant toute la période du PFE.

Analyse statistique - Intervalles qualitées - Robots de Fabrication CMM

JHOJANDRELLI CONGOTE CONJ28038807
IGNACE BERTRAND TAKUPO CHENDJOU TAKI24098600
NADIR HAMROUN HAMN07027402

RÉSUMÉ

Ce document présente l'analyse, réingénierie, exigences fonctionnelles, exigences non fonctionnelles, cas d'utilisations, conception des amélioration et analyse de faiblesses, outils de développement et d'implémentation des nouvelles fonctionnalités d'un système informatique qui effectue le contrôle statistique de procédé permettant l'extraction, la transformation, le traitement et l'affichage dans des rapports et tableaux de bord des données générées par plusieurs robots dans l'industrie automobile.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

Tout au long de ce document, des abréviations et acronymes sont utilisés, pour faciliter la compréhension du contenu, le tableau sous dessus illustre une description détaillée de chaque acronyme.

Acronymes	Description
PFE	Projet de fin d'études
SQL	Langage de requête structurée qui sert à effectuer des opérations sur des bases de données relationnelles (Structured Query Language)
MVC	Modèle Vue Contrôleur
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
ORM	Mapping Objet-Relationnel
BD	Base de données
SCRUM	Méthodologie de découpage du projet en boîtes de temps, nommées « sprints ».
BO	Business Object
DTO	Data Transfer Object
DAO	Data Access Object
CSV	Comma Separated Values

INTRODUCTION

NetSen Group est une entreprise œuvrant dans le domaine des technologies de l'information (TI) et de la fourniture d'équipement et de consommables informatiques. Elle propose aux entreprises et aux organismes des secteurs privés et publics des services de conseil en réingénierie de processus d'affaires, en développement web et logiciel, en intégration et optimisation de systèmes informatiques qui permettent à ses clients d'augmenter leur productivité et leur compétitivité et de réaliser des résultats mesurables.

Dans sa stratégie commerciale de moderniser ses services, notre client ambitionne de mettre en place un nouveau site web. Nous avons comme objectif de faire l'étude et la conception d'un module de contrôle statistique de procédé permettant l'extraction, la transformation, le traitement et l'affichage dans des rapports et tableaux de bord des données générées par plusieurs robots dans l'industrie automobile.

Cette nouvelle conception devra permettre d'améliorer le temps de réponse de la création d'un rapport. La stratégie mise en place devra faire de la prédiction en temps réel basé sur l'historique des mesures, le comportement des pièces mesurées et en fonction des limites de contrôles et de spécifications définies par le client.

Des alertes et graphiques seront générés et envoyés automatiquement en fonction des résultats de pièces non conformes ou hors contrôles. Les graphiques devront être générés dans des rapports et envoient au besoin par mail. Un plan d'action corrective via un formulaire et un workflow avec cycle de validation devra être mis en place. Vu le nombre de données générées, une architecture en big data devra être considérée pour une meilleurs optimisation et temps de réponses.

CHAPITRE 1

PRÉSENTATION DE L'ENTREPRISE NETSEN

1.1 Présentation de l'entreprise NetSen

NetSen Group est une entreprise qui fait partie du domaine des technologies de l'information (TI) et de la fourniture d'équipement et de consommables informatiques. Elle suggère aux entreprises et aux organismes des secteurs privés et publics des services de conseil en réingénierie de processus d'affaires, en développement web et logiciel, en intégration et optimisation de systèmes informatiques qui permettent à ses clients d'augmenter leur productivité et leur compétitivité et de réaliser des résultats mesurables.

1.2 Situation actuelle du système existant

NetSen Group a développé un système qui effectue le contrôle statistique de procédés permettant l'extraction, la transformation, le traitement et l'affichage dans des rapports et tableaux de bord des données générées par plusieurs robots dans l'industrie automobile. Nous nous sommes donc fait approcher par cette compagnie pour implémenter plusieurs améliorations et modifications au système actuel.

Le système actuel est composé principalement du langage C# et fonctionne en utilisant de technologies Microsoft comme la suite de logiciels de développement Visual Studio 2012, le système de gestion de base de données SQL Server 2012, la collection d'outils Bootstrap pour le design des interfaces graphiques, qui est un ensemble de code HTML et CSS, l'ORM (object-relational mapping) open source nHibernate pour gérer la persistance des objets en base de données relationnelle, le framework Spring.Net pour construire et définir l'infrastructure de l'application, ainsi que la bibliothèque logicielle Dojo en JavaScript.

CHAPITRE 2

MISE EN CONTEXTE

2.1 Processus existants

Comme nous l'avons mentionné dans l'introduction, le but du système est d'effectuer le contrôle statistique de procédés permettant l'extraction, la transformation, le traitement et l'affichage dans des rapports et tableaux de bord des données générées par plusieurs robots dans l'industrie automobile. Lo robot envoie les données à l'aide d'un fichier CSV qui est stocké dans un dossier au serveur. Le système est composé de plusieurs modules, par exemple le module CMMLISTENER est le responsable du processus ETL du système, celui qui fait la lecture, le traitement et la charge de données dans la base de données en étoile. Le système est connecté à deux bases de données, une base de données relationnelle et une autre en étoile, dans lesquelles la table principale contient les faits et les autres tables contiennent les dimensions. Finalement, le module BASECMM est le module responsable de la génération de rapports statistiques, lesquels sont affichés dans l'interface du système et peuvent être envoyés par courriel aux utilisateurs en format PDF.

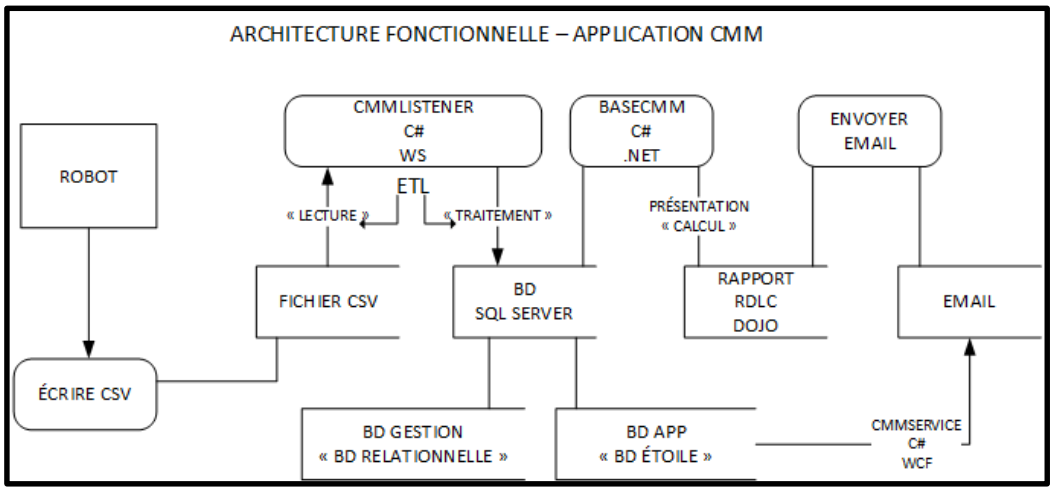

Figure 1: Architecture fonctionnelle - application CMM

2.2 Faiblesse du système actuel

Dans l'analyse du système actuel, nous avons remarqué plusieurs faiblesses dans la conception et dans l'implémentation de l'application. Nous avons remarqué que les interfaces graphiques ne sont pas responsives au moment de l'affichage des diverses plateformes. Une autre faiblesse que nous avons trouvée dans le système actuel est le temps d'exécution de rapports, car l'affichage des rapports après avoir lancé la requête prend du temps excessif. Finalement, nous avons identifié plusieurs fonctionnalités de base qui ne sont pas disponibles pour les utilisateurs du système, comme l'inscription ou la gestion des utilisateurs.

2.3 Problématique du système actuel

Le système actuel présente plusieurs problèmes de performance au moment de la génération de rapport statistique par rapport les indicateurs de qualités des pièces automobiles, car la génération d'un seul prendre beaucoup de temps et impact l'efficacité de l'application. En autre, le système n'a pas été implémenté avec une structure MVC, cela rendre complexe la modularité, la maintenabilité et l'évolution en performance du système. Un autre défi pour ce système est la migration des données vers le nuage, à l'aide de l'implémentation d'Azure, laquelle est la plateforme applicative en nuage de Microsoft.

CHAPITRE 3

ANALYSE DU SYSTÈME EXISTANT

3.1 Le But

Dans la stratégie commerciale de moderniser ses services, le client ambitionne de mettre en place un nouveau site web. Nous avons comme but de faire l'étude et la conception d'un module de contrôle statistiques de procédé permettant l'extraction, la transformation, le traitement et l'affichage dans des rapports et tableaux de bord des données générées par plusieurs robots dans l'industrie automobile.

3.2 Objectifs

Après l'analyse du système actuel et selon les besoins du client, la conception du nouveau système devra être responsive et devra permettre d'améliorer le temps de réponse de la création d'un rapport. La stratégie mise en place devra faire de la prédiction en temps réel basé sur l'historique des mesures, le comportement des pièces mesurées et en fonction des limites de contrôles et de spécifications définies par le client.

En addition, des alertes et graphiques seront générés et envoyés automatiquement en fonction des résultats de pièces non conformes ou hors contrôles. Les graphiques devront être générés dans des rapports et envoient au besoin par mail. Un plan d'action corrective via un formulaire et un workflow avec cycle de validation devra être mis en place. Vu le nombre de données générées, une architecture en big data devra être considérée pour une meilleurs optimisation et temps de réponses.

Au final, nous aspirons à fournir une application web dynamique qui donnera une nouvelle image à l'entreprise. L'interface sera conviviale et responsive. La création de rapports se fera dans un délai raisonnable grâce à la performance de nouvelles formules et architecture qui seront mises en place.

3.3. Planification

Le plan était de concevoir un plan de travail et des échéanciers pour établir l'ordre de l'avancement du projet. La compréhension et l'analyse du système actuel, ainsi que la conception et le l'implémentation de modifications suggérées devraient être basées sur des techniques éprouvées. Ainsi, dans l'étude de cas, il a fallu identifier les attributs fonctionnels et non fonctionnels, la suggestion des nouvelles interfaces et la vérification de la performance du système avec le représentant de l'entreprise Netsen Group.

3.4 Réingénierie du système existant

L'objectif principal de cette section consiste à inspecter un système déployé en production dans le but de comprendre son fonctionnement, son architecture et ses faiblesses, pour ce faire, l'application des patrons de réingénierie s'avère indispensable pour tenter de comprendre le système afin d'apporter une solution adéquate à l'amélioration de la performance et l'intégration d'une nouvelle interface usager.

Le processus général du système est assez complexe, mais peut être résumé comme suit :

- Écriture des données du robot dans un fichier CSV
- Extraction des données CSV et l'écriture dans la base de données
- Affichage des données dans un rapport
- Gestion des utilisateurs

Lors de notre réingénierie, nous voulions apporter un début de solution au problème de performance du système. Nous sommes attardés sur la partie logique du système. La réingénierie du système est basée sur le principe des patrons décrit dans la section réalisation de rétro-ingénierie.

3.4.1 Réalisation de la rétro-ingénierie

Le but de cette section consiste à analyser le système pour comprendre son fonctionnement afin de créer une représentation à un plus haut niveau d'abstraction. pour ce faire nous appliquer les patrons suivants:

Patrons	**Justification / Explication**
Agree on Maxim's : Accepter les maximes	Avoir une idée claire du travail à accomplir et d'établir les priorités pour le projet.
Appoint a navigator : Définir un navigateur	Garder le cap sur le but déterminé, préserver la vision architecturale durant le déroulement du projet en assignant la tâche à un membre de l'équipe.
Speak to the round table: Organiser des reunions	Assuré de maintenir une bonne communication entre les parties prenantes et de faire le suivi des travaux.
Most Valuable first : Commencer par le plus important	Avoir le plus de résultats utiles pour le moins de temps dépensé.
Read All the code in one hour: Lire le code en une heure	Prendre connaissance des tendances générales du code.
Skim the documentation : Faire un survol de la documentation.	Prendre connaissance des informations considérées importantes à documenter par l'équipe de développement.
Do a mock installation : Faire une installation factice	Comprendre le but et l'utilité du programme, et de valider son bon fonctionnement.
Analyze the persistent Data : Analyser les données persistantes	Faire une première image de l'architecture du système grâce à l'analyse de la documentation et du code.
Study the Exceptional Entities : Étudier les entités exceptionnelles	Focaliser sur les points plus particuliers du code afin d'en comprendre les particularités.
Step through the execution : Suivre l'exécution pas à pas	Comprendre le rôle et l'utilité de chaque classe et le flux de données.
Look for the contracts : Trouver les contrats	Comprendre le but de chaque classe et surtout la manière d'utiliser les classes interfaces

Tableau 1 : Patrons

3.4.2 Architecture

L'architecture client-serveur mise en place, gère et traite les données stockées dans la base de données dans le but de les envoyer ensuite au client.

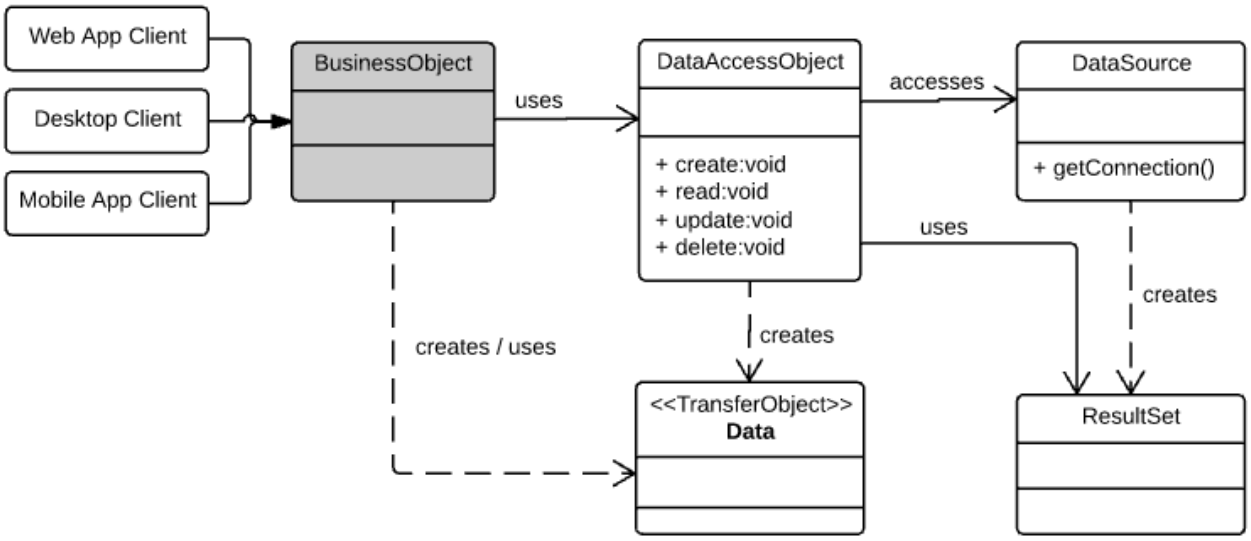

Figure 2: Architecture logique

Le modèle d'objet d'accès aux données ou le modèle DAO est utilisé pour séparer les API ou les opérations d'accès aux données. Ce modèle permet d'encapsuler la liaison avec le système de données et des objets métiers. Voici les composantes qui le composent :

- Data Access Object (DAO) : entités chargées de faire des requêtes avec la base de données
- Data Transfert Object (DTO) : entités chargées de simplifier le transfert des données qui transitent du serveur au client.
- Business Objects (BO) : Entités chargées de la logique d'affaire de chaque module du système.

3.5 Base de données

Le système actuel est composé de deux bases de données, la première est une base de données relationnelle, dans laquelle se trouve l'information par rapport les utilisateurs du système, et cela nous permet la gestion de groupes, des permissions et de rôles des utilisateurs. La deuxième base de données, est une base de données en étoiles, dans laquelle il y a une table principale appelée *Enregistrement Mesure*, laquelle stocke les données par rapport les mesures envoyées par le robot. Cette table est liée avec d'autres tables de dimension, lesquelles vont permettre de faire des agrégations des faits, selon les besoins des utilisateurs dans l'exécution de rapports. Nous avons ajouté deux exemples des schémas des bases de données ci-dessous.

- **BD Gestion d'utilisateurs:**

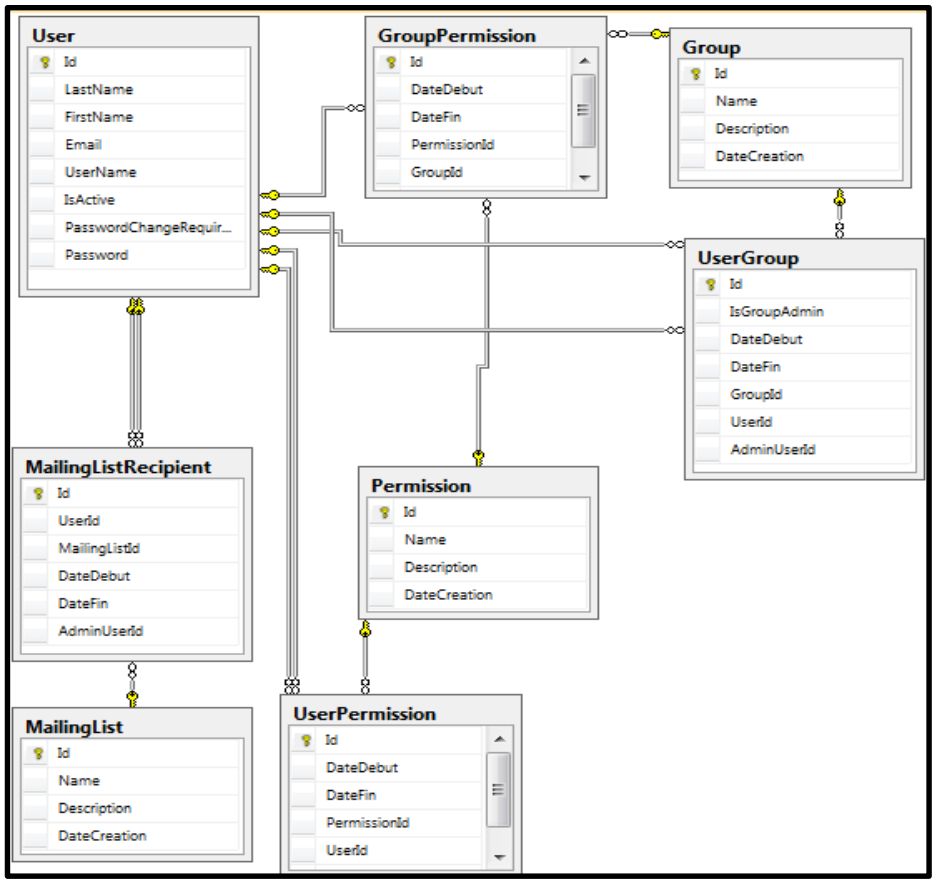

Figure 3: Schéma de la base de données relationnelle

- **BD en Étoile:**

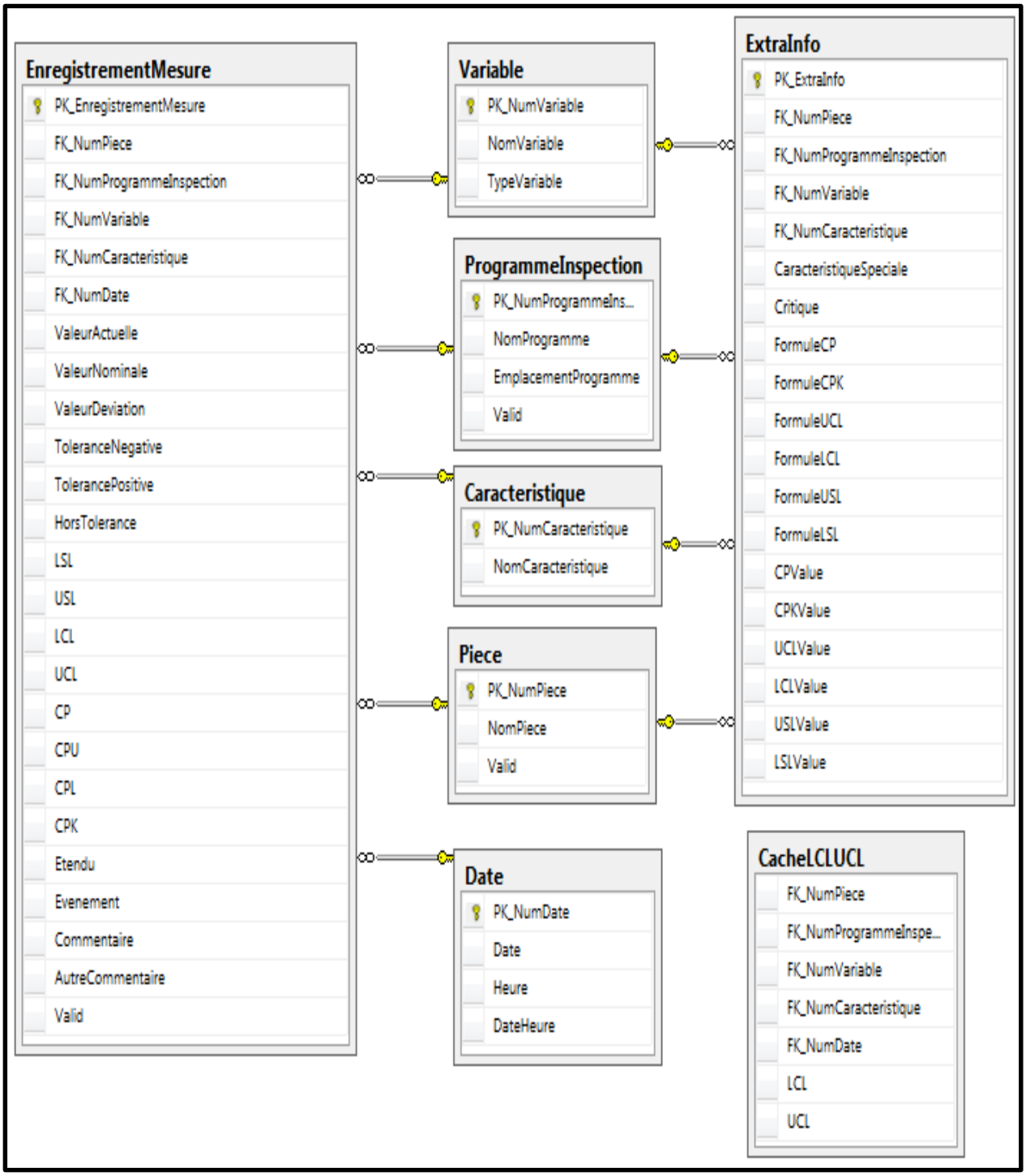

Figure 4: Schéma de la base de données en étoile

CHAPITRE 4

CONCEPTION DE LA SOLUTION

4.1 Objectifs

Principalement, nous avons travaillé dans l'amélioration des interfaces graphiques de l'application avec l'objectif de rendre les interfaces responsives est plus amiables du côté client, ainsi que l'affichage de rapports statistiques par rapport l'analyse des pièces automobiles. Également, nous avons effectué l'analyse de l'architecture de bases de données relationnelles et en étoile présentement dans le système, ainsi que la recherche de l'amélioration du processus ETL dans l'intégration de données.

En outre, nous avons mis en place une architecture MVC et nous avons intégré la plupart de modules de l'application pour la rendre fonctionnelle. De plus, nous avons suggéré de modifications à effectuer pour augmenter la performance dans la génération des rapports statistiques et pour l'intégration de données dans le nuage.

4.2 Diagramme de cas d'utilisation

4.2.1 Cas d'utilisation CU1: Se connecter

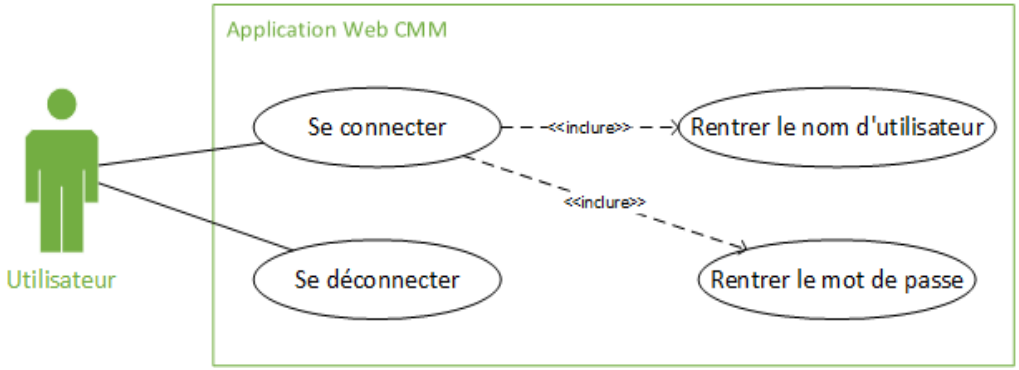

Figure 5: CU1: Se connecter

4.2.2 Cas d'utilisation CU2: S'enregistrer

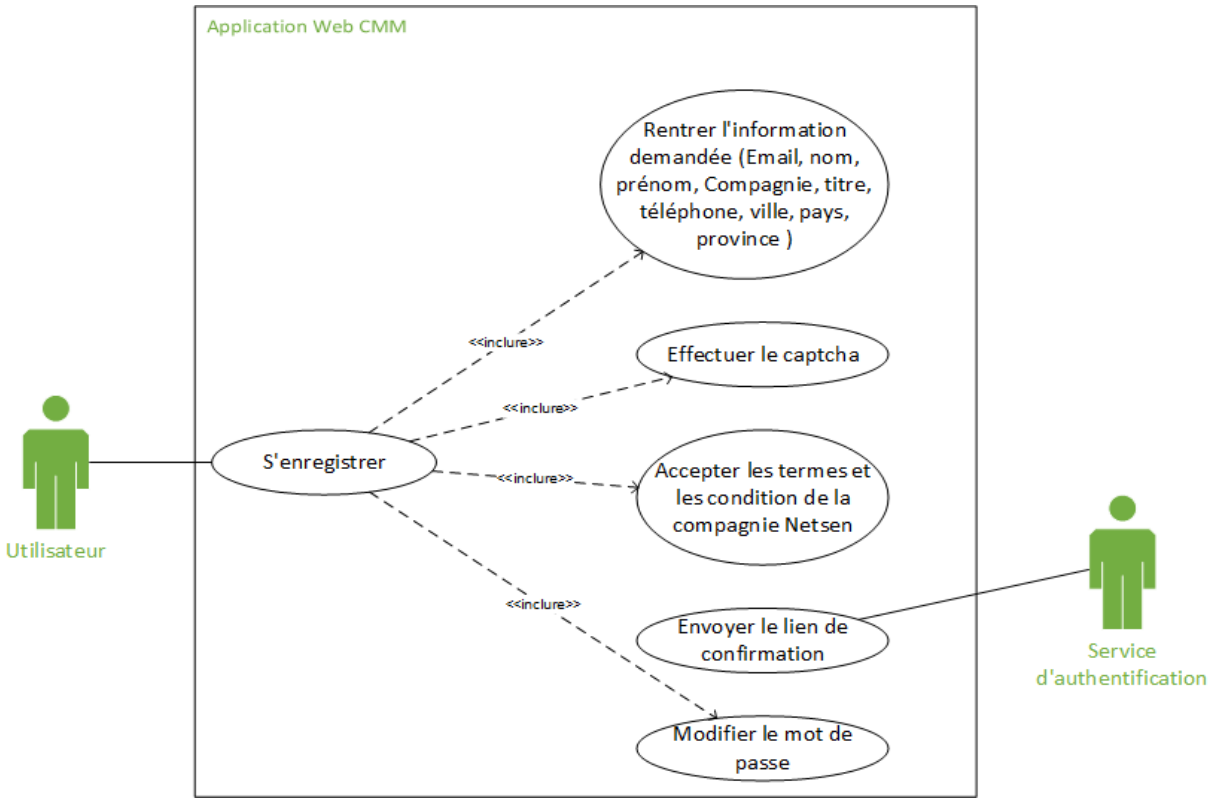

Figure 6: CU2: S'enregistrer

4.2.3 Cas d'utilisation CU3: Afficher un rapport

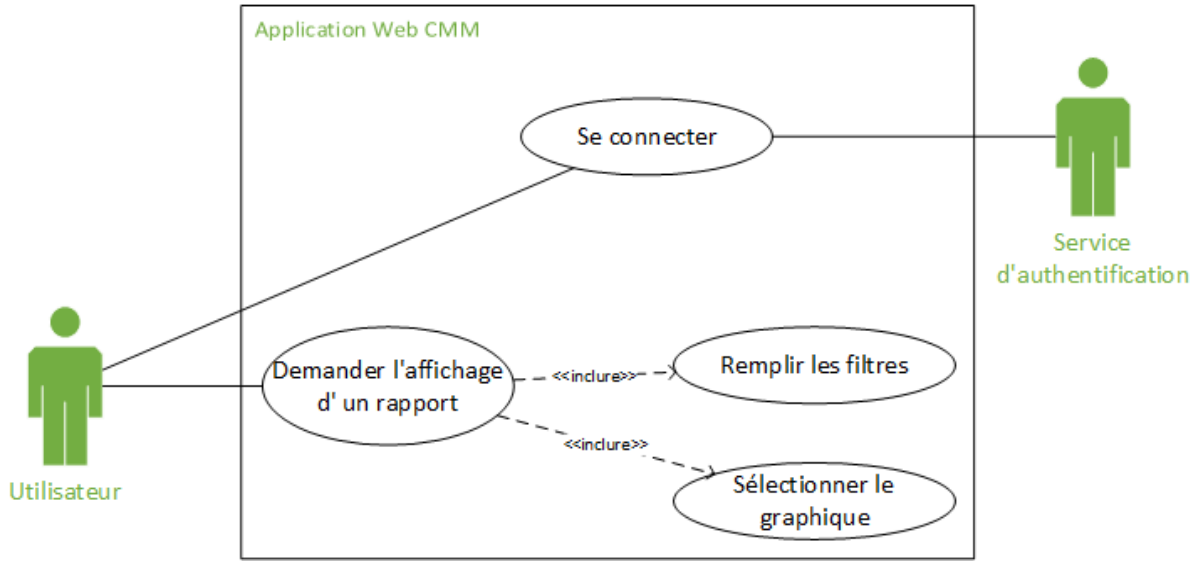

Figure 7: CU3: Afficher un rapport

4.2.4 Cas d'utilisation CU4: Modifier le mot de passe

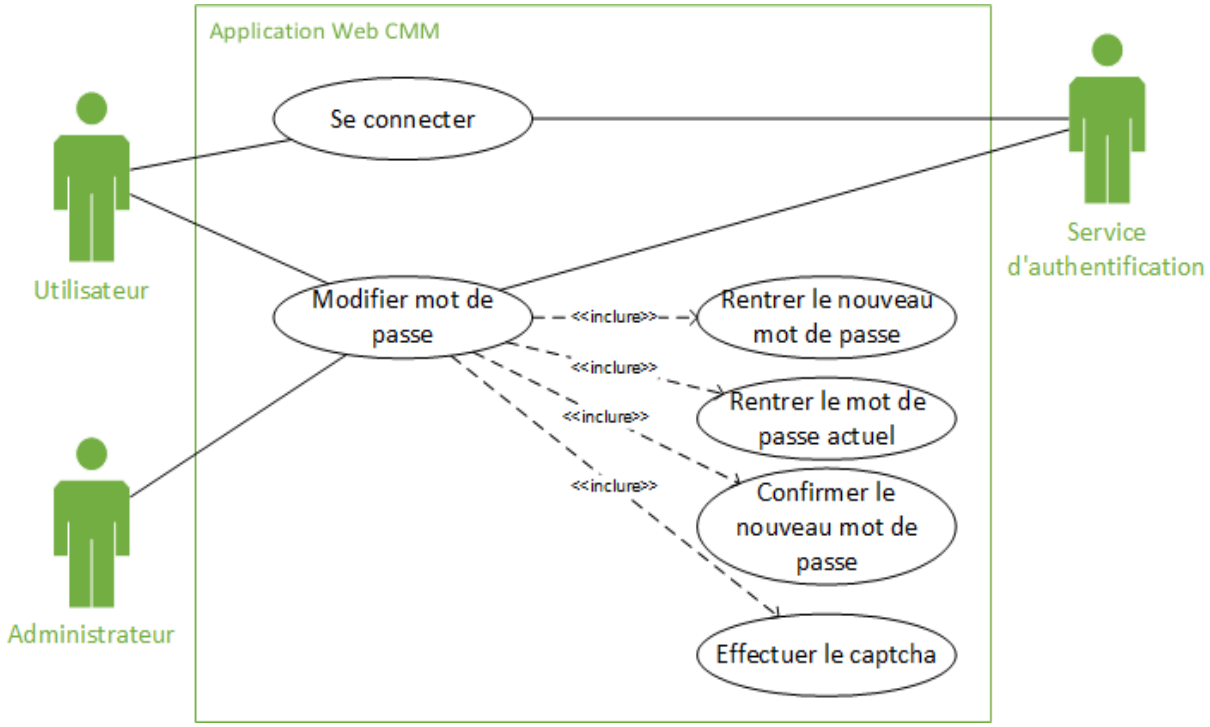

Figure 8: CU4: Modifier le mot de passe

4.2.5 Cas d'utilisation CU5: Crée/modifier/supprimer un utilisateur

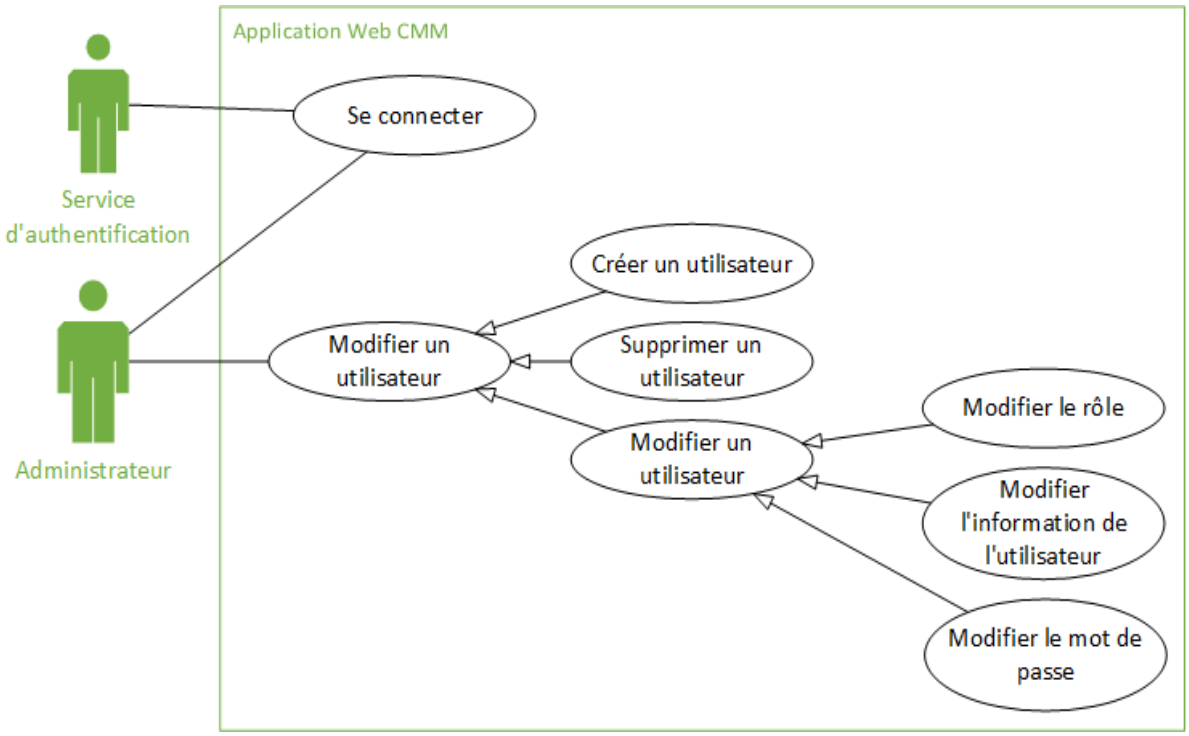

Figure 9: CU5: Créer/modifier/supprimer un utilisateur

4.3 Les acteurs

Utilisateur

L'utilisateur correspond à une personne intéressée dans l'utilisation de l'application et qui utilisera l'application afin d'afficher les rapports statistiques disponibles dans l'application.

Administrateur

L'administrateur est la personne avec le rôle d'administrateur, lequel peut modifier, créer et modifier des utilisateurs.

4.4 Les exigences fonctionnelles du nouveau système

4.4.1 EF01 - Permettre la connexion des utilisateurs

Un utilisateur existant doit pouvoir se connecter dans l'application avec son nom d'utilisateur et son mot de passe.

4.4.2 EF02 - Enregistrer des utilisateurs

Un nouvel utilisateur doit pouvoir être en mesure de s'enregistrer dans l'application sans l'aide de l'administrateur.

4.4.3 EF03 - Générer les rapports statistiques

L'application doit permettre la génération et l'affichage de rapport, ainsi que de graphiques statistiques par rapport les données stockées dans la base de données.

4.4.4 EF04 - Modifier de mots de passe des utilisateurs

Les utilisateurs peuvent modifier ses mots de passe, ainsi que les utilisateurs avec le rôle d'administrateur.

4.4.5 EF05 - Créer, modifier ou supprimer des comptes d'utilisateurs

L'utilisateur avec le rôle d'administrateur peut créer, modifier ou supprimer de comptes existants.

4.5 Les exigences non fonctionnelles du nouveau système

4.5.1 Modificabilité

Les changements au code du système pour ajouter de nouvelles fonctionnalités doivent être

possibles dans un délai minimum possible, ainsi qu'un nouveau développeur doit être en mesure de comprendre un module du code rapidement.

4.5.2 Performance

L'application doit permettre aux utilisateurs d'afficher un rapport avec un délai maximum de 2 minutes, ainsi que les graphiques et les calculs statistiques.

4.5.3 Convivialité de l'interface

Les Interfaces du système doivent être faciles à utiliser par les utilisateurs du système, elles doivent être responsives et bien structurées.

4.6 Démarche de conception

4.6.1 Approche itérative

Pour atteindre les objectifs dans les meilleures conditions et livrer un produit de qualité conçu selon les normes de l'assurance qualité logicielle et les bonnes pratiques, l'approche itérative est privilégiée par rapport aux autres méthodes, car elle permet le découpage en itérations pour mieux suivre l'état d'avancement du projet. Dans le but de s'approcher le plus possible du besoin réel, une partie exécutable du système final est produite à la fin de chacune itération et validée avec le client et les parties prenantes impliquées dans le projet.

4.6.2 Méthode agile (SCRUM)

Nous avons implémenté le cadre méthodologique SCRUM pour l'avancement et l'itération des implémentations dans le développement du projet. Chaque vendredi nous avons eu de rencontres avec l'ensemble de l'équipe, dans ces rencontres les avancements, les bloquants, les questions et les tâches étaient définies pour chaque "sprint", lesquels la plupart du temps étaient d'une semaine.

4.7 Présentations des solutions possibles

L'application existante avait entre autres des problèmes de performance et elle n'était pas responsive. Pour pallier à ce problème, nous avons présenté plusieurs solutions. Ces solutions reposent sur des patrons de conceptions.

4.7.1 Patron MVC

MVC est un patron de conception très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données.

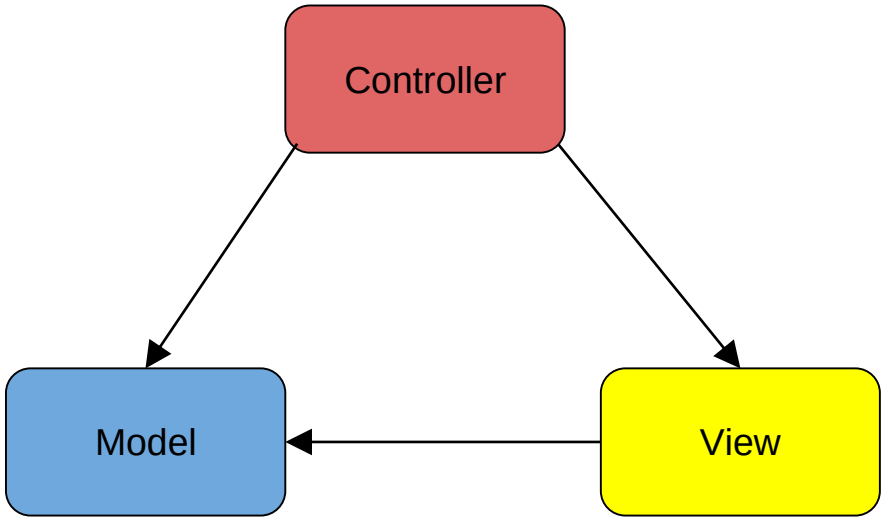

Figure 10: Architecture MVC

- Couche vue : La couche vue gère l'affichage des interfaces. Elle retourne la présentation des données.
- Couche contrôleur : Elle gère l'interaction avec les utilisateurs. Elle travaille avec le modèle et sélectionne la vue indiquée.
- Couche modèle : La couche modèle met en œuvre la logique de l'application de données. C'est cette couche qui gère la lecture des données, la conversion selon des concepts de l'application et la validation.

4.7.2 Cache mémoire

La mise en cache est une technologie de stockage des données utilisateur en mémoire, de sorte que, lorsque les mêmes données sont demandées la prochaine fois, elles peuvent être directement extraites de la mémoire au lieu d'être générées par l'application. Dans le cas de l'application NetSen, cette technique est complètement absente dans l'implémentation, c'est pourquoi les délais d'accès aux données sont problématiques. La mise en cache des données est l'une des solutions à mettre en œuvre pour améliorer la performance d'accès aux données. On distingue deux concepts de mise en cache à savoir : Output Caching et Object Caching. La section suivante illustre les différentes façons de mise en place de cette technique.

4.7.2 .1 Patron Proxy

L'utilisation du patron proxy est l'une des options envisagées pour régler le problème de performance lors de chargement des données dans les différentes vues de l'application. L'objectif de proxy est de fournir un intermédiaire entre le client et un objet pour contrôler les accès à ce dernier. Dans notre cas, son rôle consiste à gérer le résultat des requêtes sollicitées pour le client, toutes les requêtes envoyées par le client passent par le proxy, avant d'exécuter une fonction, ce dernier doit vérifier d'abord si l'objet demandé par le client n'est pas déjà dans le cache, sinon il doit exécuter la fonction et sauvegarder le résultat dans le cache. Le diagramme de classe de la **figure 11** illustre les différentes classes qui composent le patron proxy.

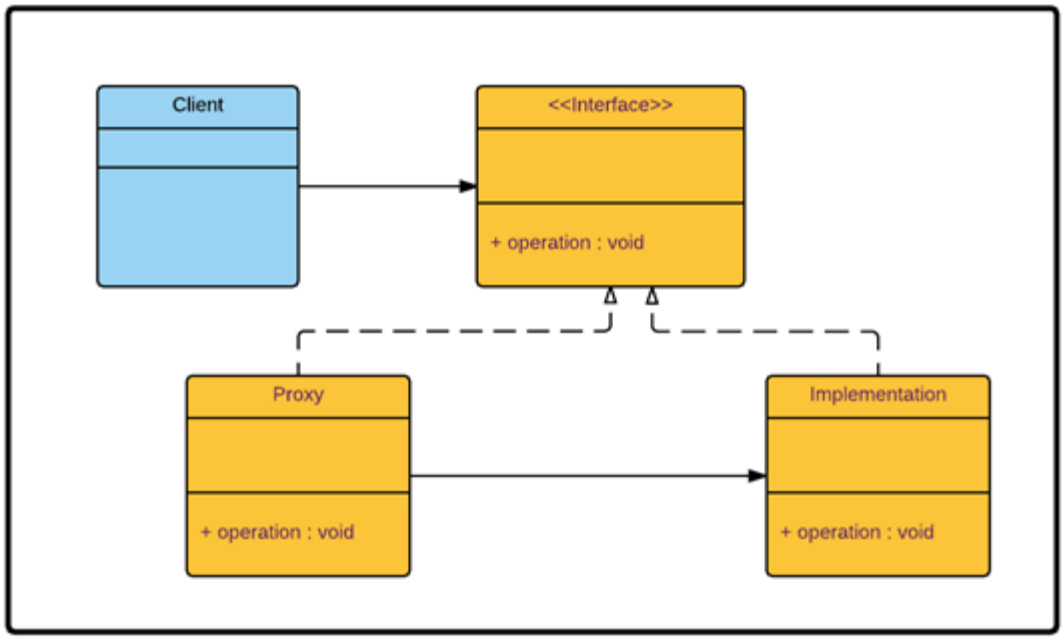

Figure 11: UML Proxy

4.7.2.2 Spring :

Le framework Spring est intégré dans le projet dans la persistance des objets, il permet d'implémenter une gestion de cache avec le moindre effort, autrement dit sans trop écrire de code, par contre, il ne permet pas de gérer tous les cas.

4.7.3 Patron Adapter

Pour permettre à la nouvelle application MVC de fonctionner avec l'application NetSen, il est nécessaire de traiter le problème d'incompatibilité, car leurs interfaces sont différentes. Pour se faire, l'implémentation d'un adaptateur est indispensable. L'objectif de l'Adapter consiste à :

- Convertir l'interface d'une classe en une autre interface qui est attendue par un client.
- Faire collaborer des classes qui n'auraient pas pu le faire à cause de l'incompatibilité de leurs interfaces.

Le résultat de l'application de l'Adapter est d'isoler le sous-système (Application MVC).

Le rôle de chaque classe peut être défini comme suit :

- ***IAdapter*** définit l'interface spécifique à l'application que le client utilise,
- Le **Client** collabore avec les objets qui sont conformes à l'interface de ***IAdapter,***
- La classe à adapter ***Adaptee*** est l'interface existante qui a besoin d'adaptation,
- L'adaptateur ***Adapter*** adapte effectivement l'interface de ***Adaptee*** à l'interface de ***IAdapter*** par la traduction des accès.

Le schéma de la **figure 12** est un du diagramme UML du patron Adapter.

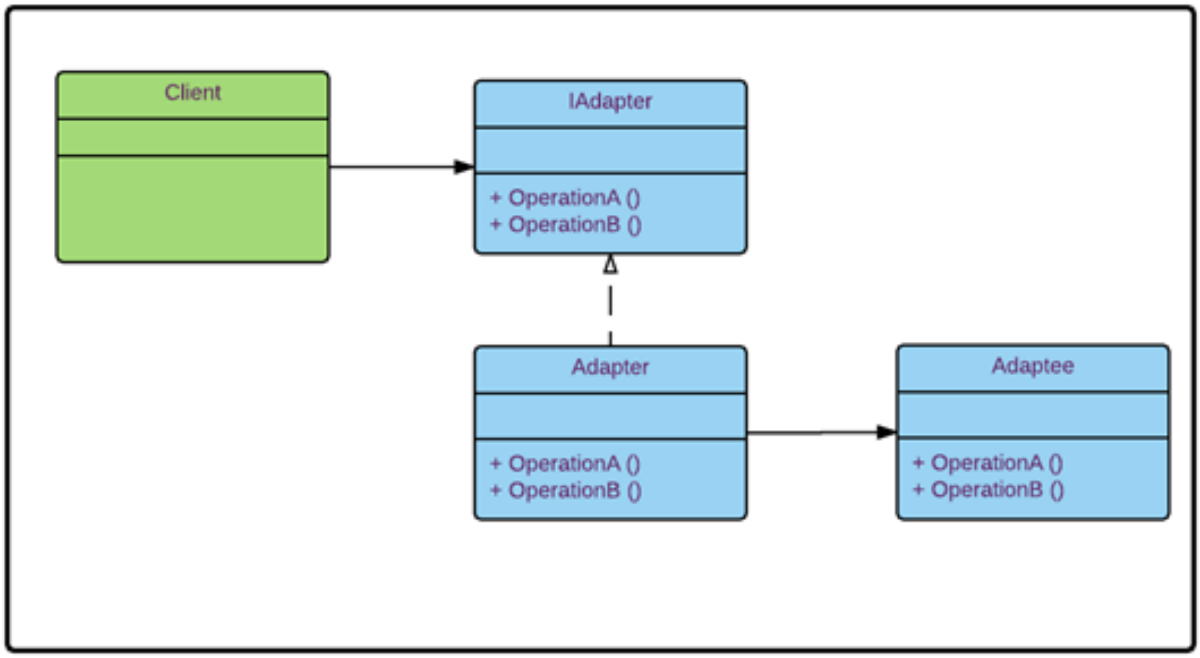

Figure 12: UML Adapter

4.7.2.3 OutPut Cache

Cette technique permet de sauvegarder la vue dans un cache pour une durée définie, pour éviter une communication intempestive avec le serveur.

4.7.4 Solution retenue

La solution retenue pour répondre aux exigences fonctionnelles et non fonctionnelles est la suivante :

1. Créer un Module BaseCmmMVC;
2. Implémenter le patron de conception Adapter pour pouvoir :
    - Adapter les objets du projet BaseCmmMVC aux abjects du projet NetSen.
    - Adapter les objets du projet NetSen au BaseCmmMVC
3. Implémenter le patron proxy pour augmenter la performance en utilisant la technique de gestion de cache mémoire.

Cette solution a été retenue, car elle est indépendante du système existant. Le module BaseCmmMVC créé n'a aucun impact sur les fonctionnalités du système déjà en production. En plus elle permet de livrer toutes les exigences fonctionnelles et non fonctionnelles en respectant les attributs de qualités logicielles suivantes :

- Interopérabilité : les composants de BaseCmmMVC interagissent facilement avec les autres composants du système NetSen.
- Extensibilité : Facilité à ajouter des nouveaux composants à BaseCmmMVc sans aucun impact sur le système.
- Utilisabilité : la facilité d'un usager à accomplir une tâche avec le moindre effort
- Réutilisabilité : réutilisation des composants déjà implémentés.
- Performance : Amélioration du temps de réponse aux requêtes d'interrogation de la base de données,
- La maintenabilité : Facilité et stabilité aux changements sans impact sur le reste du système.
- Interchangeabilité : Changer un composant facilement et sans impact sur le reste du système.
- Adaptabilité : il s'agit de l'adaptation du module sans aucun impacter les fonctionnalités du système existant.

Le schéma de la **figure 13** illustre l'architecture de l'intégration de la solution dans le système existant.

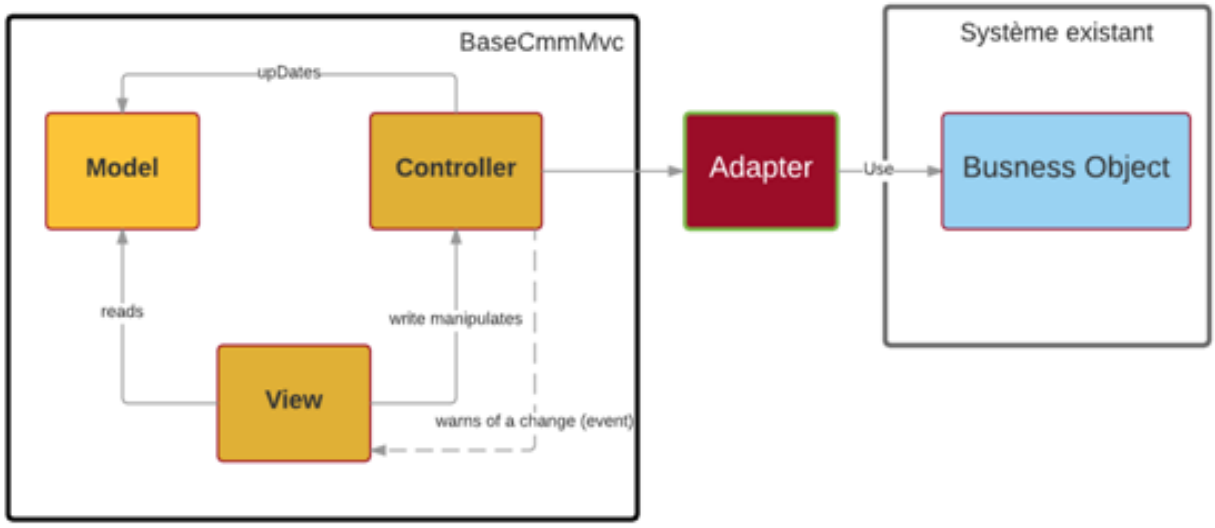

Figure 13: Architecture de la solution dans le système existant

CHAPITRE 5

MODÉLISATION CONCEPTUELLE DE LA SOLUTION

5.1 Intégration des patrons de conception dans le projet

5.1.1 Patron MVC dans le projet NetSen

La technologie utilisée est l'ASP MVC. Cette technologie repose principalement sur le patron MVC, qui permet de concevoir des applications de manière claire et efficace grâce à la séparation des intentions.

- Couche vue : Nous avons utilisé la technologie ASP RAZOR pour réaliser les vues de l'application. Razor est basé sur ASP.NET et conçu pour créer des applications Web. Il a le pouvoir de baliser ASP.NET traditionnel, mais il est plus facile à utiliser et plus facile à apprendre.
- Couche modèle : La couche modèle met en œuvre la logique de l'application de données. C'est cette couche qui gère la lecture des données, la conversion selon des concepts de l'application et la validation.
- Couche contrôleur : Elle gère l'interaction avec les utilisateurs. Elle travaille avec le modèle et sélectionne la vue indiquée. Nous avons mis en place trois principales actions :
    - ActionResult : Elle encapsule le résultat d'une méthode d'action et retourne à une vue (.html)
    - FileResult : Elle retourne un fichier. Nous l'avons utilisé pour permettre à l'utilisateur de visualiser le rapport généré.

- JsonResult : Elle retourne du Json. Nous l'avons utilisée pour envoyer les données du graphe à la vue.

5.1.2 Patron proxy dans le projet NetSen

La solution retenue pour gérer le problème de performance est l'application d'un patron proxy, ce dernier est intégré de la façon suivante :

1. Le client **UserController** sollicite les services de l'interface ***IMangeUser*** sans se préoccuper de la façon dont ses services son implémenter,
2. Les deux classes ***Proxy*** et ***DTOVsModels*** implémentent l'interface ***IMangeUser***
3. La classe ***DTOVsModels*** a pour rôles d'adapter les résultats obtenus et de retourner des objets compatibles avec le module BaseCmmMVC
4. La classe proxy implémenter l'interface ***IMangeUser*** et utilise les services de ***DTOVsModels***, mais avant d'appeler une méthode de ***DTOVsModels***, le proxy doit vérifier si un cache mémoire est déjà créé pour sauvegarder le résultat, si le cache est vide alors il exécute la méthode sinon il retourne le contenu du cache mémoire.

Le schéma de la **figure 14** est un aperçu de toutes les classes nécessaires pour implémenter le patron Proxy.

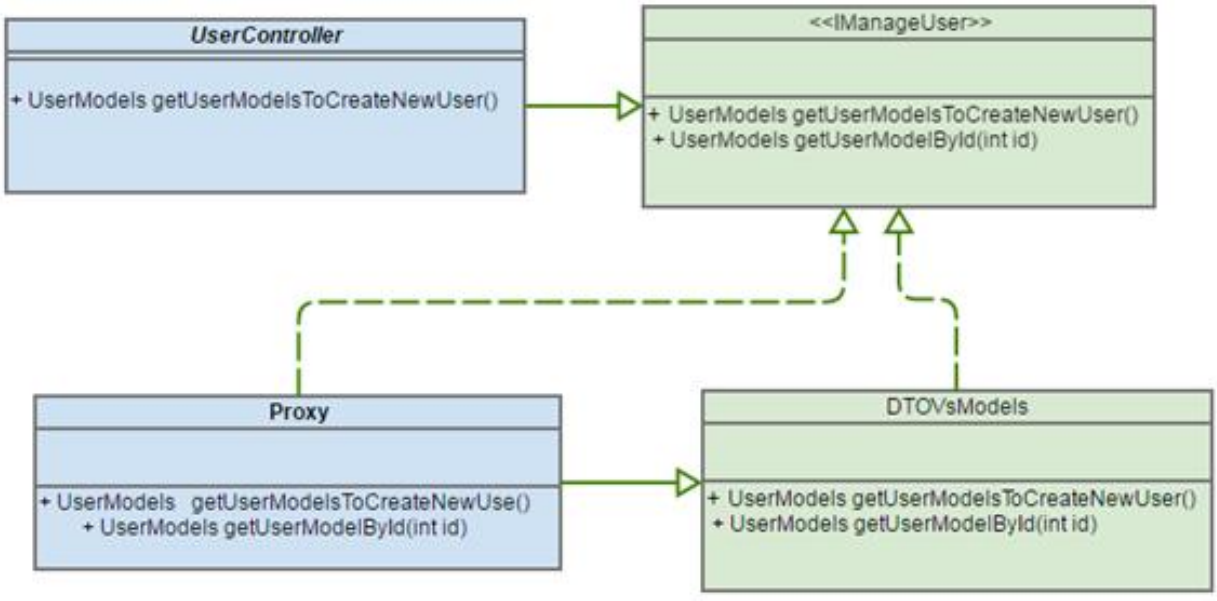

Figure 14: Diagramme UML du patron Proxy

5.1.3 Patron Adapter dans BaseCmmMVC

Le module BaseCmmMVC doit communiquer avec les autres modules du projet, pour ce faire il doit utiliser la classe ***FactoryBO*** , par contre, cette dernière retourne des objets incompatibles avec le module BaseCmmMVC, d'où l'application du patron Adapter pour assurer la compatibilité entres les modules, cette tâche est confiée la classe ***DTOVsModels*** dont le rôle consiste à utiliser les services de la classe ***FactoryBo*** et adapter les objets et retourner des objets compatibles avec le module BaseCmmMVC.

Le schéma de la **figure 15** est un aperçu de toutes les classes nécessaires pour implémenter le pattern Adapter.

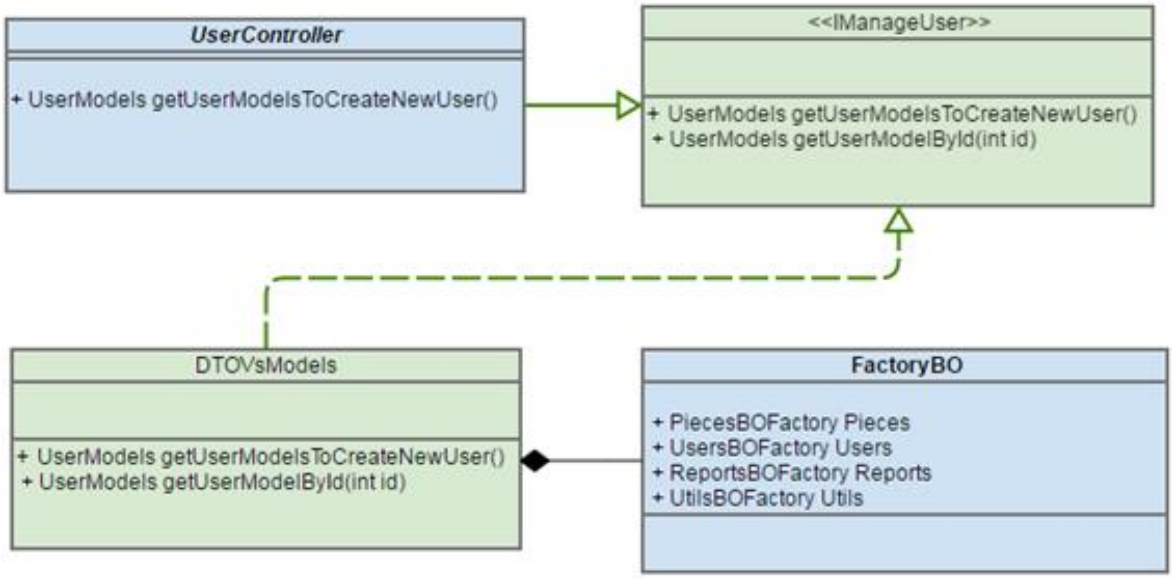

Figure 15: Diagramme UML du patron Adapter

5.3 Présentations des vues du nouveau système

5.3.1 Vue structurelle

5.3.1.1 Diagramme de classes

Le diagramme de classe illustre l'intégration des patrons de conceptions dans la solution

MVC et son interaction avec le système existant.

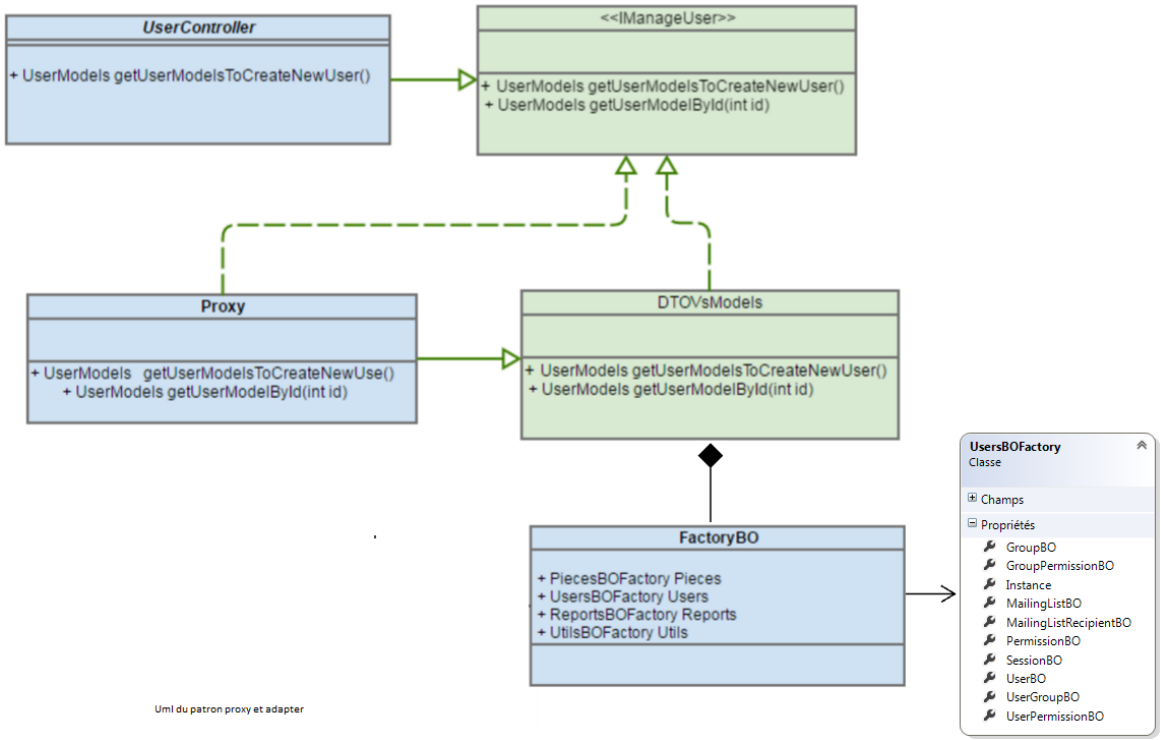

Figure 16: Diagramme de classes de l'intégration des patrons de conception dans la solution

5.3.2 Vue comportementales

5.3.2.1 Diagramme de séquence

Pour illustrer d'une manière claire la collaboration entre les objets, voici un exemple de diagramme de séquences qui montre les interactions entre les objets lorsque le client sollicite les services proxy.

1. La classe ***UserController*** créer une instance de la classe proxy pour utiliser ses services
2. La classe proxy utilise la méthode ***chechCacheMemory()*** pour vérifier l'état de la mémoire cache, si ce dernier et non vide, alors le contenu d sera retourné, sinon un

objet ***DTOVsModels*** est instancié puis exécuter la méthode ***getUserModelsToCreateNewUser().***

3. ***DTOVsModels*** return un Objet de type ***User Models***
4. Le ***Proxy*** exécute la méthode ***checkUserModel()*** pour vérifier le résultat retourné par ***DTOVSModels***, si le résultat est différent de nul alors, le ***Proxy*** devra créer un cache mémoire pour sauvegarder et retourner le résultat à ***UserController***.

Le schéma de la **figure 16** est un aperçu des séquences d'interactions entre les objets.

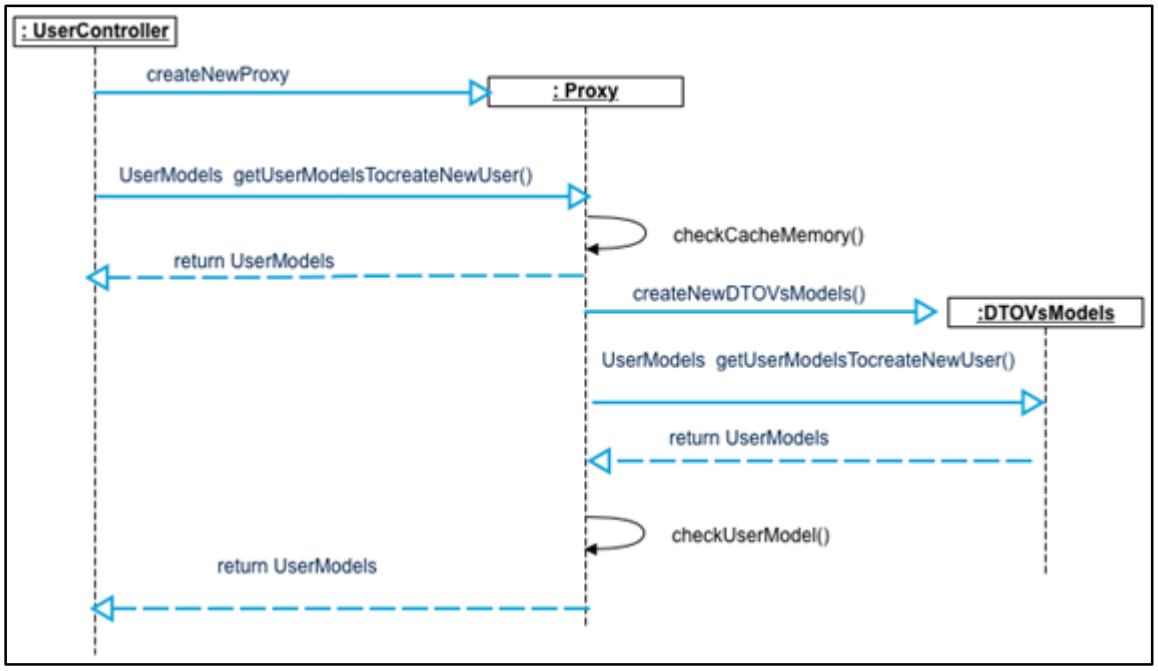

Figure 17: Diagramme de séquence pour obtenir UserModels

CHAPITRE 6

IMPLÉMENTATION DES INTERFACES

6.1 Objectif

Un des objectifs principaux du projet était l'amélioration des interfaces et la mise à jour de librairies par rapport aux pages web qui composaient l'application. Dans ce chapitre nous décrivons les différentes stratégies implémentées dans l'amélioration des interfaces graphiques. Il s'agit bien sûre de l'implémentation des cas d'utilisations décrits à la section 4.2.

6.2 Gestion des usagers

Cette section consiste à implémenter les interfaces graphiques pour chaque cas d'utilisation relatif à la gestion des usagers. Il s'agit de de créer les vues pour permettre aux usagers d'effectuer les opérations de création et de mise à jour des données.

6.3 Graphisme et Rapports

Les graphes des rapports sont générés grâce à la technologie AmCharts. AmCharts est une librairie JavaScript qui permet de générer des diagrammes. Elle est Open Source, mais on doit accepter avoir un lien publicitaire au-dessus de notre graphe. Il est possible de faire un paiement pour ne plus avoir cette publicité.

Dans le cadre de notre projet, nous avons exclusivement utilisé les graphiques en ligne (Line Charts). Les graphes devaient montrer l'évolution de la mesures d'une ou plusieurs pièces dans le fil du temps.

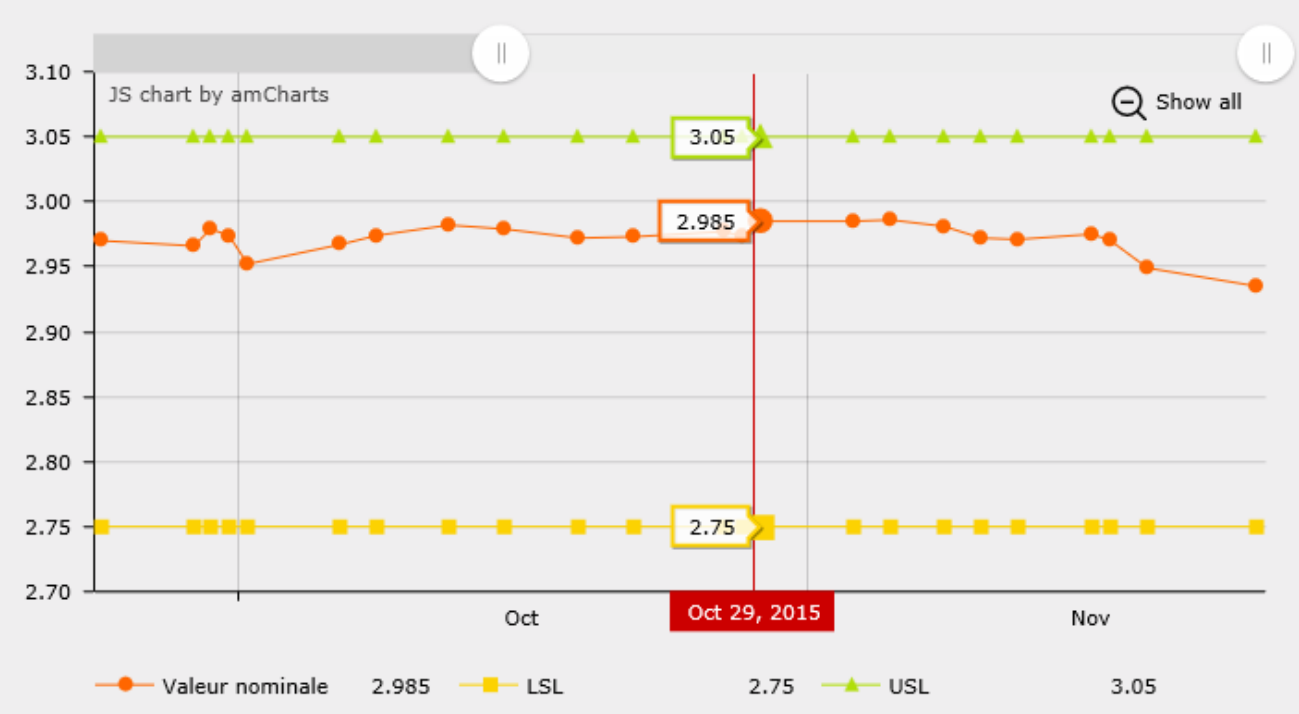

Figure 18 : AmCharts - graphe de présentation de données

Une fois le rapport généré, il est possible de le pré visualiser et de l'exporter en PDF. Pour ce faire, nous avons mis en place une maquette (Skin) par défaut pour tous les rapports. La maquette comporte 3 parties :

- le haut de page : il comprend le logo de l'entreprise à gauche, le nom du rapport au centre et le numéro de page à droite.
- le corps de page : c'est ici que se trouvera les données du rapport.
- le bas de page : il comprend le nom de l'entreprise avec l'adresse à gauche, la date et l'heure à droite.

6.4 Système d'identification

Principalement dans la modification des interfaces graphiques par rapport l'identification des utilisateurs, nous avons utilisé la collection d'outils à la création du design de sites et d'applications web "Bootstrap", ainsi que les feuilles de style en cascade "CSS, lesquelles nous ont permis de

respecter les contraintes d'affichages du client et d'implémenter code des interfaces d'une façon plus résumée et plus claire.

6.5 Les outils

Pour l'accomplissement du projet, nous avons utilisé plusieurs plateformes informatiques par rapport le développement logiciel, la gestion des projets et la gestion de versions du code.

6.5.1 BitBucket

Une des contraintes établies par une des parties prenantes du système a été l'utilisation du BitBucket pour la gestion de développement, car BitBucket est un service web d'hébergement et de gestion de développement logiciel utilisant les logiciels de gestion de versions Git et Mercurial.

6.5.2 SoureTree

Le client nous a suggéré également l'utilisation de l'interface graphique pour la gestion du développement logiciel SourceTree, car cette interface simplifie la façon dont nous interagissons avec les dépôts Git afin de nous concentrer sur le codage. Cette interface nous a aidés dans la gestion de nos référentiels, hébergés ou locaux dans nos ordinateurs.

6.5.3 Jira

Un autre système que le client nous a suggéré a été JIRA. Le client avait déjà en place les différents sprints à suivre, auquel les tâches et les sous-tâches à effectuer étaient déjà définies. Ce système a été de grande utilité, car il nous a permis d'effectuer le suivi de bugs, de faire la gestion des incidents, et de suivre la gestion du projet par rapport le client.

6.5.4 Plateforme de développement

Le système existant a été conçu avec le langage de programmation C#. De plus, une autre contrainte du système a été l'utilisation de l'environnement de développement logiciel Visual Studio 2012 et le SGBD SQL Server 2012.

CONCLUSION

L'analyse et le suivi des données des pièces d'automobile générées par le robot, constituent un enjeu majeur pour notre client. Les statistiques obtenues devraient permettre vite prendre des décisions et ainsi gagner en temps et en argent.

Le projet a permis à chaque membre de l'équipe de trouver une tâche différente à faire parmi les besoins établis dans JIRA par le client. L'implémentation de la plupart des interfaces responsives a été déployée à l'aide de Bootstrap, CSS et JQuery. De plus, l'implantation du modèle MVC représente bien l'aspect trois-tiers du projet. La charge de travail étant très élevée selon le nombre de membres dans l'équipe a été bien gérée et a apporté un niveau de contribution équitable.

Pendant l'exécution du projet, nous avons eu besoin d'apprendre des nouvelles technologies, des nouveaux langages de programmation, des plateformes pour la gestion de projets et la gestion de développement, par exemple:

- Nouvelles connaissances qui ont été acquises: Visual Studio, C#, .Net MVC, CSS, Bootstrap, JQuery, DOJO, amChart, nHibernate, Spring framework, Entity framework, ETL.
- Plusieurs connaissances qui ont été renforcées: HTML, JavaScript, SQL, bases de données, patrons de conception, architecture logicielle, SCRUM.

Une des faiblesses majeures de ce système est sa performance. Pour s'attaquer à ce problème de performance, il faudrait restructurer la couche logique et la base de données. Ceci peut être considéré comme une suite à notre projet.

RECOMMANDATIONS

Avoir un document d’étapes à suivre, pour exécuter l’installation de l’application dans un poste de développement aurait nous épargner beaucoup de temps au début du projet.

Effectuer une analyse plus en détail de l’utilisation de la mémoire cache au moment de l'exécution de requêtes à la base de données par rapport du rôle attribué à l’utilisateur qui effectue la demande du rapport statistique.

Faire une recherche approfondie par rapport l'utilisation d’une base de données NoSQL, avec l’objectif de comparer des améliorations de performance au moment de l'exécution de rapport et la visualisation de graphiques.

Réfléchir à une nouvelle façon de travailler. Il serait intéressant de rencontrer physiquement de temps à autres le client. Il pourrait ainsi apporter une réponse rapide à nos questions et nous faire des suggestions dans notre façon de faire.

LISTE DE RÉFÉRENCES

SITE WEB

Wikipedia. (2017, avril.). Netsen Group [site Web]. Consulté le 30 janvier. 2017. http://www.netsengroup.com/

SITE WEB

Wikipedia. (2017, avril.). Scrum (Boîte à outils) [site Web]. Consulté le 28 mars. 2017. https://fr.wikipedia.org/wiki/Scrum_(Boite_%C3%A0_outils).

SITE WEB

Wikipedia. (2017, janvier.). Azure [site Web]. Consulté le 20 février. 2017. https://azure.microsoft.com/fr-ca/overview/what-is-azure/

SITE WEB

Wikipedia. (2017, mars.). Bootstrap (framework) [site Web]. Consulté le 1er avril. 2017. https://fr.wikipedia.org/wiki/Bootstrap_(framework).

SITE WEB

Wikipedia. (2014, août.). ASP MVC [site Web]. Consulté le 15 février. 2017. http://www.microsoftvirtualacademy.com/training-courses/introduction-to-asp-net-mvc.

SITE WEB

Wikipedia. (2016, octobre.). Dojo Toolkit [site Web]. Consulté le 1er mars. 2017. https://fr.wikipedia.org/wiki/Dojo_Toolkit.

SITE WEB

Wikipedia. (2016, novembre.). Spring Framework [site Web]. Consulté le 18 mars. 2017. https://fr.wikipedia.org/wiki/Spring_(framework).

ANNEXES

Annexe A : Planification

Voici le plan de travail qu'on a tenté de respecter. Certes, on n'a pas pu honorer à la lettre le temps attribué à chacune des tâches, car ce tableau représentait uniquement une estimation. Il était prévu que chacun réalise une moyenne de 180 heures de travail, mais ce seuil a été dépassé lors de la phase de conception et d'implémentation. Cela s'explique par la complexité de l'application.

	Commence	**Termine**	**Efforts estimés ***	**Tâches/Jalon**	**Livrable(s)/Artéfacts**	**Responsable**
	2017-01-13	2017-01-13	2	Remise de la fiche de renseignements	Fiche de renseignements	# heures par chaque membre
	2017-01-20	2017-01-26	16	Installation et préparation de l'environnement technologique de l'application	Tutoriel d'installation de l'application	# heures par chaque membre
	2017-01-27	2017-01-30	16	Analyses et apprentissage de Visual Studio et les applications MVC		# heures par chaque membre
.	2017-01-31	2017-02-03	2	Rencontre – professeur superviseur		# heures par chaque membre
.	2017-02-04	2017-02-10	14	Rencontre – professeur superviseur, apprentissage de Bootstrap		# heures par chaque membre
	2017-02-11	2017-02-17	10	Conception des nouvelles Interfaces graphiques de l'application	Prototype interface de l'application	Nadir
			10			Jhojandrelli
			20			Ignace
	2017-02-18	2017-02-24	10	Remise de la proposition de projet	Proposition de projet	# heures par chaque membre
			10	Analyse de la gestion de rôles d'utilisateurs du système		Nadir

	2017-02-24	2017-03-03	10	Analyse du processus ETL de l'application		Jhojandrelli
	2017-03-03	2017-03-10	16	Rédaction d'un document de Vision et d'un SRS en fonction des requis	Document de vision et SRS	# heures par chaque membre
0	2017-03-10	2017-03-17	10	Modification des rapports statistiques générés par l'application	Code modifié sur Bitbucket	Jhojandrelli
			10			Nadir
			10			Ignace
1	2017-03-13	2017-03-24	10	Intégration de l'application MVC	Code modifié sur Bitbucket	Jhojandrelli
			10			Nadir
			14			Ignace
2	2017-03-15	2017-03-24	12	Remise du rapport d'étape	Rapport	# heures par chaque membre
3	2017-03-24	2017-04-01	8	Rédaction d'un document d'architecture	Document d'architecture	Ignace
			12			Nadir
			12			Jhojandrelli
4	2017-04-01	2017-04-07	20	Rédaction du guide d'utilisateur et implémentation de l'application web.	Application web et Guide d'utilisateur	# heures par chaque membre
5	2017-04-07	2017-04-14	20	Remise du travail et exécution de dernières modifications du code source	Rapport et code source	# heures par chaque membre

Sous-total: 180 heures de travail dans la session pour chaque membre de l'équipe.