|  |  |
| --- | --- |
| Rapport de projet developpement  CESI | Rapport de projet développement du groupe : Brut, Nunesse, Sorrenti et Rochette |

Table des matières

[Table des Illustrations 1](#_Toc44402682)

[I. Introduction : 2](#_Toc44402683)

[II. Rappel du besoin : 2](#_Toc44402684)

[III. Découpage du projet : 2](#_Toc44402685)

[IV. Planification initiale : 2](#_Toc44402686)

[a. Répartition des tâches : 2](#_Toc44402687)

[V. Modélisation de l’application distribuée : 3](#_Toc44402688)

[VI. Analyse des écarts : 4](#_Toc44402689)

[VII. Analyse des compétences acquises par étudiants : 4](#_Toc44402690)

[VIII. Bilan : 4](#_Toc44402691)

# Table des Illustrations

[Figure 1 - Planning prévisionnel 2](#_Toc44402726)

[Figure 2 - Architecture logicielle globale 3](#_Toc44402727)

# Introduction :

Suite aux choix de la spécialisation développement de 4ème année, les étudiants on à réaliser un projet sur les notions vue en cours lors des différents blocs. Ce projet nécessite une équipe d’expert .NET(C#/WCF/SQL Server) et d’expert Java(EJB/JMS/web services/Oracles).

# Rappel du besoin :

Ce projet implique la création d’une plateforme logiciel permettant à un utilisateur de s’authentifier, de déchiffrer des fichiers cryptés et d’obtenir le fichier cible en clair ainsi que la clé. La plateforme doit être composé et codé dans différent langages (.Net et Java).

La plateforme .Net se charge de :

* Permettre l’authentification d’un utilisateur
* Permettre la réception de fichier crypter
* Générer une série de clés utile au déchiffrage
* Déchiffrer les fichiers cryptés avec les clés
* Envoyer les fichiers décryptés à la plateforme JEE
* Générer un PDF avec les retours de la plateforme JEE
* Envoyer un mail à l’utilisateur avec le PDF

La Plateforme JEE se charge de :

* Réceptionner les fichiers décrypter
* S’assurer que les fichiers sont correctement décryptés
* Chercher l’informations secrète
* Faire un retour a l’application .NET (Nom du fichier, information secrète et clé)

# Découpage du projet :

Ce projet peut être découper de la manière suivante :

Figure - Découpage du Projet

# Planification initiale :

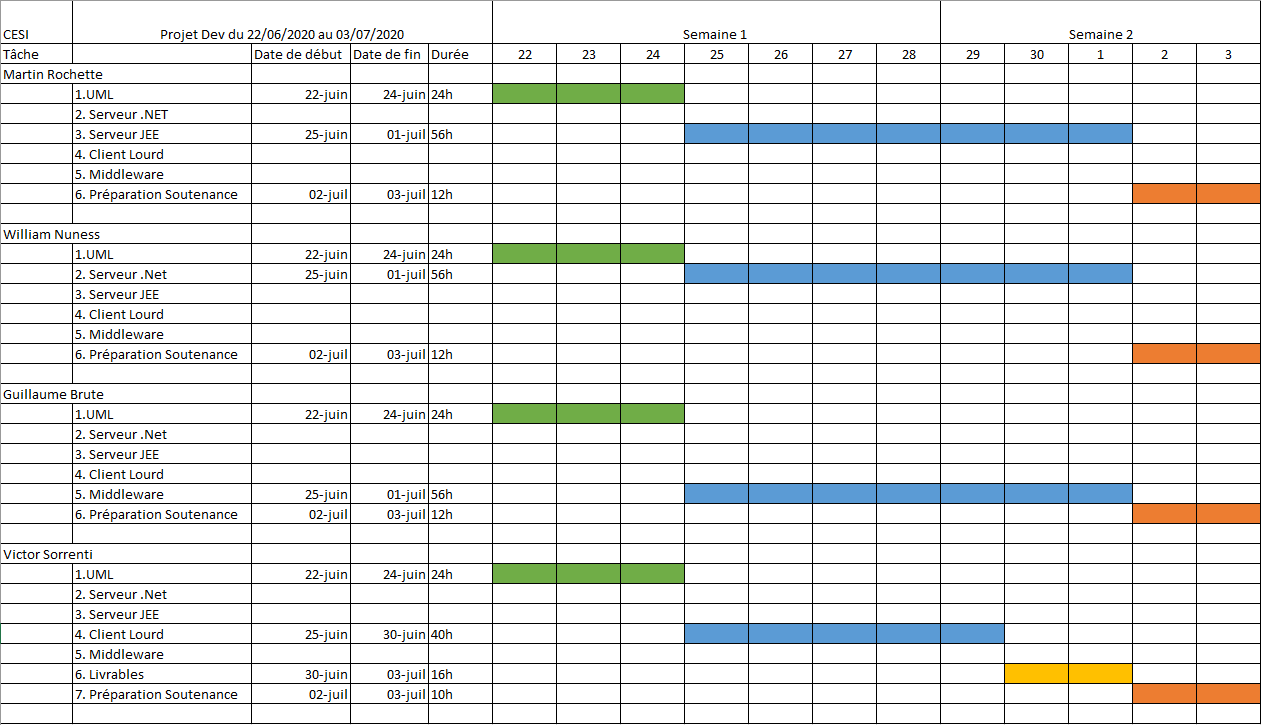
Après un brainstorming nous avons conclu la planification prévisionnelle suivante :

Figure - Planning prévisionnel

# Répartition des tâches :

Suite à la planification des tâches, nous avons réparti les tâches comme ceci au départ :

* Conception du projet tous ensemble (Brainstorming, diagramme) ;
* Martin s’occupe du développement du serveur JEE ;
* William s’occupe du développement du serveur .NET ;
* Guillaume s’occupe du développement du Middleware ;
* Victor s’occupe du développement du client lourd ;

# Modélisation de l’application distribuée :

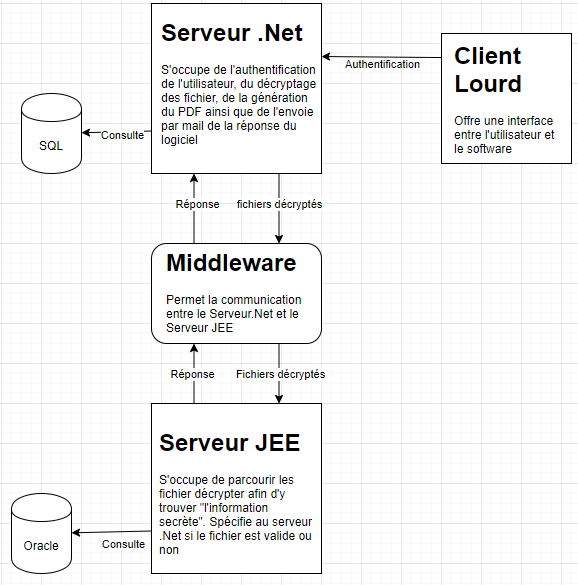


Figure - Architecture logicielle globale

Ce diagramme représente l’architecture logiciel de notre plateforme. Elle permet de mettre en évidence le fonctionnement de cette dernière. On précisera que le Middleware réceptionne les données entre .NET et JEE sérialisées en JSON afin de garantir la compatibilité.

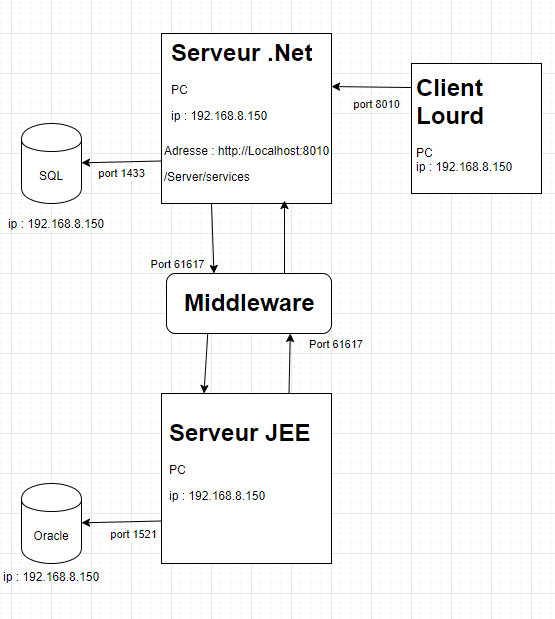


Figure - Architecture physique globale

Ce diagramme représente l’architecture physique que notre plateforme. Il permet de spécifier comment sont répartie les serveurs ainsi que leur manière de communiquer.

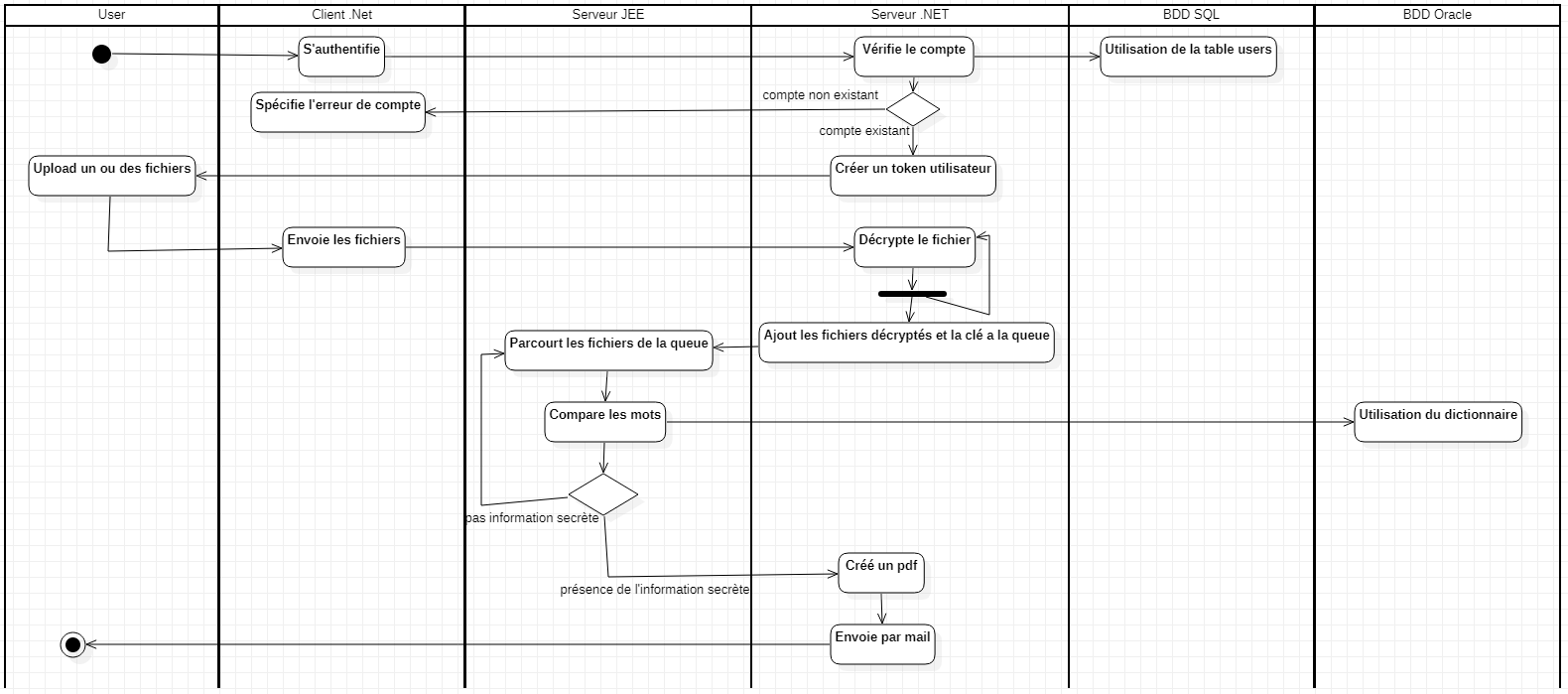


Figure - Diagramme d'activité

Ce diagramme représente l’activité notre plateforme. Elle permet de mettre évidence les différentes interactions entre les différentes composantes de notre plateforme.

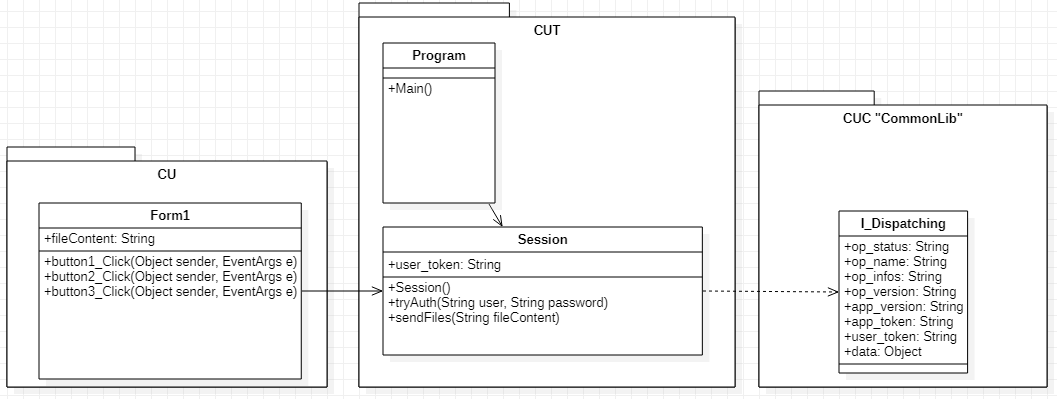


Figure - Diagramme de Classe Client Lourd

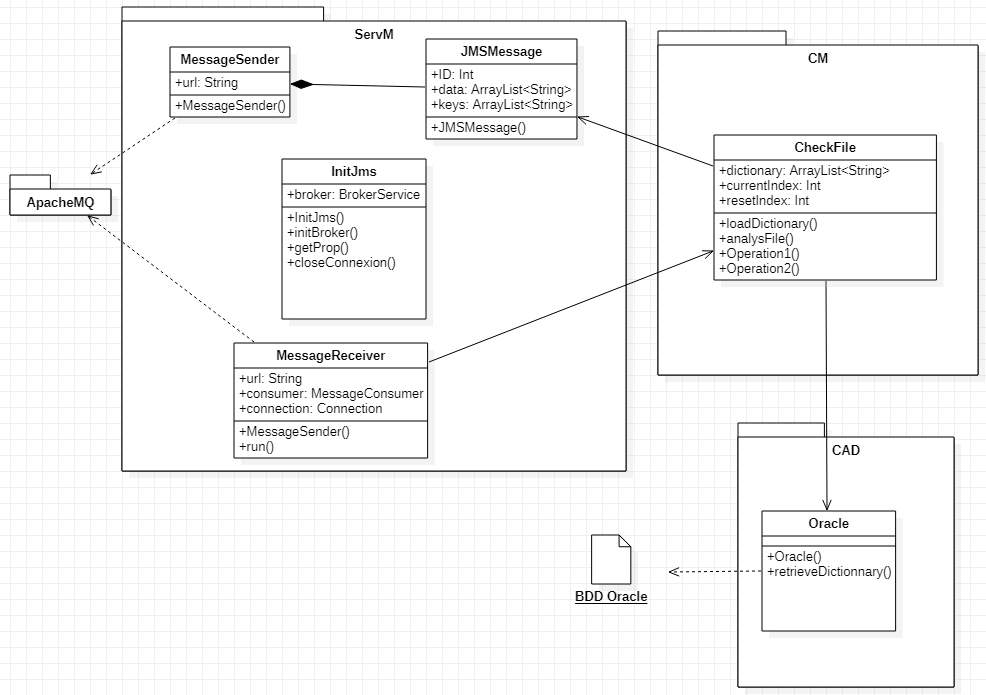


Figure - Diagramme de classe Serveur JEE

modélisation diagramme de classes par couche et par plateforme

diagramme de séquence par couche et par plateforme)

# Analyse des écarts :

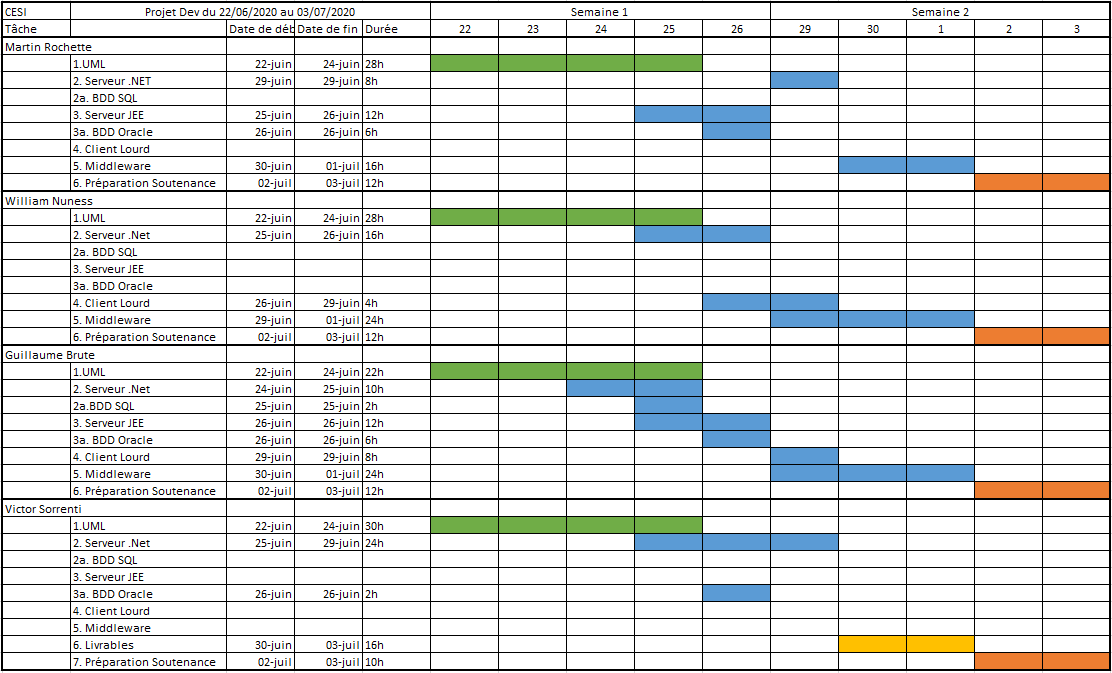
Tout d’abord nous avons remarquer de gros écart à propos de la répartition des tâches. Finalement les tâches été plus ou moins réalisées comme ceci :

Figure - Planning réel

Ces écarts de planning sont dû au fait que, lors du premier brainstorming nous n’avions pas vraiment toutes les données du projet en tête. Donc nous n’avions pas vraiment conscience du travail nécessaire, du temps et des technologies à utiliser. Nous nous sommes donc adaptés en fonction des compétences et de la volonté de chacun.

Cependant la répartition des taches à pu être revue et nous a quand même permis de réaliser le projet en temps et en heures.

* Stx 1 - Architecture : 3 tiers - SOA /
* Stx 2 - Framework : Microsoft Framework 4.0. /
* Stx 3 - Langage : C# 4.0. /
* Stx 4 - Solution : 9 couches. /
* Stx 5 - Client :
  + A déployer sur les postes clients. /
  + Lourd / asynchrone. /
  + 3 couches (1 composant physique)./
  + Parallélisassions des traitements issus du middleware (traitements lourds). Client nonbloqué lorsqu’il attend les réponses de la plateforme. /
  + WPF possible, donc pas obligatoire. En effet un design riche n’est pas nécessaire (même s’il doit être soigné) /
* Stx 6 - Middleware :
  + A déployer sur un serveur. /
  + Architecture de type service.
  + Point d’entrée unique. /
  + Communication sécurisée basée sur le chiffrement de bout en bout (sécurité du message) 2 CAHIER DES CHARGES DU PROJET PROJET DE DOMINANTE DEVELOPPEMENT / 27/02/2020 14 X
  + Structure des messages : MSG [bool statutOp – string info – object[] data – string operationName – string tokenAppl – string tokenUser – string appVersion – string operationVersion] ?
  + Une seule méthode exposée sur le point d’entrée unique de signature : STG m\_service(STG msg) ?
  + Distribué : oui. ?
  + Journalisation de l’activité du CAM. ?
  + Les contrôleurs de workflow et les composants métiers doivent supporter des charges de travail importantes. ?
  + 5 couches. /
  + 1 composant physique pour les couches 5 – 4 – 3. /
  + 1 composant physique pour la couche 2. /
  + 1 composant physique pour la couche 1. /
* Stx 7 - Data :
  + SQL Server STD Ed 2008. /
  + Mode d’authentification mixte.
  + Mise en place de l’agent SQL Server dans un processus Windows distinct. /
  + Mise en place du moteur SQL Server dans un processus Windows distinct. /
  + Sauvegarde automatique journalière (sauvegarde complète). X
* Stx 8 – Communication
  + Liaison de type netTcp entre le client lourd et la plateforme. /
  + Communication sécurisée avec un chiffrement à 128b. X

# Analyse des compétences acquises par étudiants :

Grâce à ce projet nous avons pu approfondir la plupart des compétences que nous avions vu lors de nos cours dans le bloc dominante Dev, c’est-à-dire toute la partie Java JEE (JMS, EJB etc ...) toute la partie C# (WCF,Thread etc …. ). Ce projet nous as permit aussi de revoir toutes les notions de Conception de logiciel avec notamment les Diagrammes de composants et de classes. Enfin ce projet nous a permis d’améliorer nos techniques de management de projet (gestion du temps, des tâches ...)

# Bilan :