Rapport de projet JEE

-Jouvencel Guillaume -Biteghe Johann

-Porion Quentin -Collingwood Raphael

**Gestion de banque**

Table des matières

[Contexte du projet : 1](#_Toc519542582)

[Besoins : 1](#_Toc519542583)

[Cas d’utilisation : 1](#_Toc519542584)

[Choix des spécificités fonctionnelles : 2](#_Toc519542585)

Contexte du projet : Votre entreprise a remporté un appel d’offres et vous êtes responsables de la conception d’une application web reposant sur des technologies JEE suivant le cahier des charges de la banque Crédit Agrituel qui compte plusieurs agences.

Besoins :  
Notre client souhaite une interface informatique de gestion de sa banque.  
Il souhaite apporter aux conseillers ainsi qu’aux clients une interface simplifié mais néanmoins complète des principales fonctions qu’ils peuvent effectuer en agence, au guichet.

Notre application doit donc comporter deux types de profils utilisateurs, les clients et les conseillers bancaires. Nous allons spécifiés tous les cas d’utilisation ci-dessous :

# Cas d’utilisation :

Un client doit pouvoir :

* Posséder un ou plusieurs comptes
* Visualiser l’ensemble de ses comptes
* Afficher les dernières opérations
* Effectuer un virement
* Editer un RIB
* Afficher l’ensemble des opérations effectuées depuis 0

Depuis un DAB :

* Retrait (temps réel)
* Consultation de compte
* Responsive design

Un conseiller doit pouvoir :

* Visualiser tous les comptes dont il a la charge
* Visualiser les messages envoyés par les clients
* Fixer le seul de découvert à la création du compte pour lequel le client aura versé au moins 50 euro
* Créer un compte en banque du client dont il aura la gestion
* Lire un compte en banque quel qu’il soit
* Mettre à jour un compte en banque quel qu’il soit
* Supprimer un compte en banque quel qu’il soit
* Consulter les transactions d’un client quel qu’il soit
* consulter les transactions d'un compte en banque quel qu'il soit
* opérer virements de compte à compte quel qu'il soit
* versements suite à un passage au guichet du client quel qu'il soit

Un compte doit pouvoir :

* avoir 1 ou 2 titulaires
* mémoriser toutes les transactions

Les agences doivent pouvoir :

* Posséder des clients, des comptes, des conseillers

# Choix des spécificités fonctionnelles :

Langage :

Nous avons choisi de développer notre application en J2EE parce qu’il s’agit du langage que notre agence préfère utiliser, il est complet et permet de multiple usages.

Outils choisis :

Nous avons choisi de travailler sur Netbeans, il s’agit de notre IDE référent dans notre agence pour les développements JEE, il est puissant, facile d’utilisation, gratuit et complet.

Base de données :

Notre base de données est sur Apache Derby il s’agit d’une base fiable, efficace et surtout très légère, elle s’intègre parfaitement dans tout type d’application java.

Architecture :

Pour le développement à proprement parler, nous avons fait le choix d’utiliser le MVC comme architecture, pour sa simplicité de lecture, fiabilité d’usage et d’amélioration d’application.

Gestion de projet :

Notre méthode de gestion de projet choisie était un cycle en V, nous n’avions pas affaire au client pour mettre au point des méthodes type AGIL qui nous aurait permis d’être certain de ne pas faire fausse route quant aux besoins réels.  
Nous avons donc analysé les besoins, rédigé des spécifications fonctionnelles, techniques, codé, effectué des tests d’intégration qui ont abouti à une validation de notre part sur ce que l’on pensait être conforme aux besoins du client.  
L’analyse de cette méthode est qu’il nous était impossible d’avoir des spécifications fonctionnelles qui conviennent et qui ne seraient pas modifiées par la suite.  
Lors de la livraison et de la mise en production, le client nous fera alors son retour sur lequel nous rencontrerons certainement des problèmes de fonctionnalités mal interprétés.

En annexe, vous trouverez notre diagramme de cas d’utilisation et le diagramme de classe.

# Conclusion :

Un projet comme celui-ci nous a permis d’appréhender un nouveau langage que nous pensions connaître mais qui s’est avéré beaucoup plus complexe et complet que nous l’imaginions.  
par ces fait nous avons dû revoir notre stratégie de travail, nous avons alors fait du peer-programming afin de monter en compétences en fonctions des connaissances des uns et des autres.

Annexes :

Diagramme de cas d’utilisation :

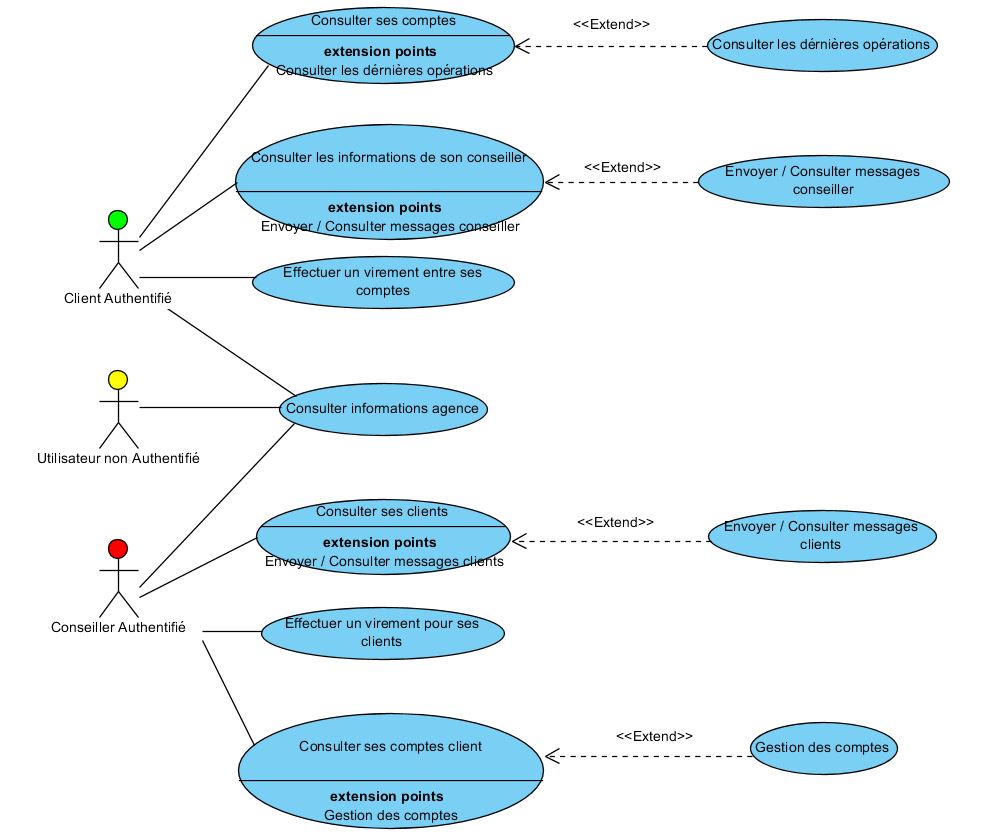


Diagramme de classes :  
(Que vous trouverez ci joint au projet dans une taille adaptée).

