**REPUBLIQUE DU SENEGAL**

***\*\*\*\*\* \* \* \*\*\*\*\*\*\*\****

**UNIVERSITE CHEIKH ANTA DIOP DE DAKAR**

****

**ECOLE SUPERIEURE POLYTECHNIQUE**

***DEPARTEMENT GENIE INFORMATIQUE***

**MEMOIRE DE FIN DE CYCLE**

**Pour l’obtention du :**

**DIPLOME D’INGENIEUR DE CONCEPTION EN**

**TELECOMMUNICATIONS ET RESEAUX (DIC)**

**SUJET :**

**Conception et Implémentation d’une Solution de SMS Banking en JEE**

**Lieu de stage**: **AtoS GDC Sénégal Période stage : 02/02/2017 – 30/06/2017**

****

|  |  |  |
| --- | --- | --- |
| ***Présenté et soutenu par***  ***Mamadou Oury DIALLO*** | ***Enseignant encadrant***  ***Dr Mandicou BA*** | ***Maître de stage***  ***Tariq DJIBALENE*** |
| *Mamadou Oury DIALLO* | *Dr. Mandicou Bah* | *M. Tariq DJIBALENE* |
|  |  |

**Année universitaire : 2016 – 2017**

**REPUBLIQUE DU SENEGAL**

***\*\*\*\*\* \* \* \*\*\*\*\*\*\*\****

**UNIVERSITE CHEIKH ANTA DIOP DE DAKAR**

****

**ECOLE SUPERIEURE POLYTECHNIQUE**

***DEPARTEMENT GENIE INFORMATIQUE***

**MEMOIRE DE FIN DE CYCLE**

**Pour l’obtention du :**

**DIPLOME D’INGENIEUR DE CONCEPTION EN**

**TELECOMMUNICATIONS ET RESEAUX (DIC)**

**SUJET :**

**Conception et Implémentation d’une Solution**

**de SMS Banking en JEE**

**Lieu de stage**: **AtoS GDC Sénégal Période stage : 02/02/2017 – 30/06/2017**

****

|  |  |  |
| --- | --- | --- |
| ***Présenté et soutenu par***  ***Mamadou Oury DIALLO*** | ***Enseignant encadreur***  ***Dr Mandicou BA*** | ***Maître de stage***  ***Tariq DJIBALENE*** |
| *Mamadou Oury DIALLO* | *Dr. Mandicou BA* | *M. Tariq DJIBALENE* |

**Année universitaire : 2016 – 2017**

# DEDICACES

GLOIRE à ALLAH, Seigneur des Terres et des Cieux et à son prophète Mohammed (PSL).

Je dédicace ce mémoire :

* A ma chère mère Mariama Korka BA et complice pour l’immense amour, le soutien qu’elle me porte et les sacrifices consentis pour la réussite de mes études
* A mon papa Abdou Karim DIALLO, paix à ton âme et qu’ALLAH notre seigneur par la grâce de son prophète Mohamed (PSL) t’accorde clémence et miséricorde et t’accueille dans son plus haut paradis
* A mon oncle Mamadou Malal DIALLO et ma tante Aïssatou DIALLO pour tous leurs efforts et conseils. Je ne saurais vous remercier assez. Que Dieu vous accorde longue vie
* A mes cousins
* A mes frères et sœurs
* A mes amis
* Aux étudiants de ma promotion DIC de l’ESP
* A tous ceux, qui ont de près ou de loin contribué à la réalisation de ce document

# REMERCIEMENTS

Je rends tout d’abord grâce à ALLAH le Miséricordieux de m’avoir accordé vie et santé jusqu’à la réalisation de ce travail, ainsi que son prophète Mohamed (Paix et salut sur lui).

Ma gratitude va d’abord à mon enseignant encadreur le Docteur Mandicou BAH et à mon maître de stage Tariq DJIBALENE, pour leur confiance, leurs conseils, leur disponibilité et leurs orientations.

Je remercie chaleureusement :

* Le Docteur Idy DIOP, responsable pédagogique de la formation DIC Télécom pour sa disponibilité et son aide durant tout ce cycle
* Le Professeur Moussa DIALLO pour son assistance, ses conseils et son aide
* Le Docteur Alassane BAH, chef de Département Génie Informatique
* Les dirigeants de la structure d’accueil du stage AtoS, les maîtres de stage puis le restent du personnel par ordre hiérarchique
* A Monsieur Kalidou SY pour son soutien et ses conseils
* A Bassirou NGOM mon collègue de chambre, pour son aide précieuse
* Tout autre intervenant dans l’élaboration du mémoire, ordonné, selon l’importance de leur participation
* Toutes les personnes qui de près ou de loin, ont contribuées à la réalisation de ce document
* L’ensemble des professeurs du département Génie informatique

# AVANT-PROPOS

Depuis sa création, l’Ecole Supérieure Polytechnique de Dakar a toujours constitué un véritable pôle d’excellence pour la formation des étudiants du Sénégal, de la sous-région et d’ailleurs. L’ESP est un établissement public qui a fêté ses 50 ans en 2014, après plusieurs mutations :

* 20 mai 1964 : création de l’Institut Polytechnique (IP), établissement technique supérieure dépendant de l’université Cheikh Anta Diop de Dakar (UCAD) et rattachée à la direction de l’enseignement supérieur.
* 15 novembre 1967, les décrets présidentiels numéros 67-12-30 et 67-12-31 créent l’Institut Universitaire de Technologie (IUT) à la place de l’IP.
* 30 avril 1973, la promulgation de la loi 73-17 et le décret 73-387 permettent à l’IUT d’être un établissement public doté de la personnalité juridique et de l’autonomie financière au sein de l’université.
* 1973-1974, des réformes ont permis à l’IUT de dispenser des enseignements en commerce et administration des entreprises. C’est ainsi que ces formations complémentaire ont entrainé la délivrance de Diplôme d’Ingénieur en Technologie (DIT) et de Diplôme d’Etude Supérieure de Commerce et d’Administration de Gestion des Entreprises (DESCAE). Ce qui entraine par conséquent une nouvelle appellation de l’institut, et devient alors l’Ecole Nationale Supérieure Universitaire de Technologie (ENSUT).
* 24 novembre 1994, la loi 94-78 crée un établissement public dénommé Ecole Supérieure Polytechnique (ESP), qui est composé de l’EPT et l’ENSETP.
* En 2005 l’ENSETP a été reconstituée en reprenant sa partie industrielle qui était dans l’ESP et en 2007 l’excentre de Thiès de l’école supérieure polytechnique ex EPT est redevenu l’Ecole Polytechnique de Thiès rattachée à l’université de Thiès et ensuite rattaché directement au ministère chargé de l’enseignement supérieur.

L’école supérieure polytechnique de Dakar est un véritable levier de développement qui est constituée de 06 départements : 05 départements industriels et 01 département de gestion. Parmi ses départements on distingue :

Le département Génie Chimique et Analyse Biologique (GCBA), le département Génie Civil, le département Génie Electrique, le département de Gestion, le département Génie Informatique et le département Génie Mécanique. L’ESP forme des Techniciens à l’issue d’une formation de 02 ans et des Ingénieurs de Conception à l’issue d’une formation de 05 ans. Le département Génie Informatique est piloté par un Chef de département en la personne de Pr Alassane BAH, assisté par des responsables pédagogiques pour chaque classe. Les responsables pédagogiques sont chargés de surveiller le déroulement normal des enseignements en effectuant un contrôle permanent sur les enseignants, les étudiants et les tests organisés en devoir simple et en devoir surveiller (DS), représentant les examens semestriels et faire part de la situation actuelle au chef de département. Les enseignements s’organisent autour des cours théoriques, pratiques et les stages d’entreprises permettant aux étudiants de mettre en œuvre les connaissances acquises. Cela m’a offert la possibilité d’effectuer mon stage durant une période de 05 mois chez AtoS Sénégal.

# Table des matières

[DEDICACES i](#_Toc486600116)

[REMERCIEMENTS ii](#_Toc486600117)

[AVANT-PROPOS iii](#_Toc486600118)

[Table des matières iv](#_Toc486600119)

[Sigles et abréviations vii](#_Toc486600120)

[Table des figures viii](#_Toc486600121)

[Table des tableaux x](#_Toc486600122)

[RESUME xi](#_Toc486600123)

[ABSTRACT xii](#_Toc486600124)

[INTRODUCTION 1](#_Toc486600125)

[CHAPITRE I PRESENTATION GENERALITE 3](#_Toc486600126)

[I.1 Présentation de la structure d’accueil 4](#_Toc486600127)

[I.1.1 AtoS en chiffre 4](#_Toc486600128)

[I.1.2 Historique 4](#_Toc486600129)

[I.1.3 Domaine d’activité 5](#_Toc486600130)

[I.1.4 Partenaires 6](#_Toc486600131)

[I.1.5 GDC Sénégal 6](#_Toc486600132)

[I.2 Présentation du sujet 8](#_Toc486600133)

[I.2.1 Contexte 8](#_Toc486600134)

[I.2.2 Problématiques 9](#_Toc486600135)

[I.2.3 Objectifs 10](#_Toc486600136)

[I.3 Conclusion 10](#_Toc486600137)

[CHAPITRE II GENERALITES 11](#_Toc486600138)

[II.1 Concepts réseaux 12](#_Toc486600139)

[II.1.1 Architecture technique du réseau GSM 12](#_Toc486600140)

[II.1.2 Présentation du SMSC dans le cas des réseaux 2G/3G 13](#_Toc486600141)

[II.1.3 Fonctionnement des SMS : Mode PULL, Mode PUSH 13](#_Toc486600142)

[II.1.4 Principe de mise en place d’un ESME 15](#_Toc486600143)

[II.1.5 Notion de sécurité GSM 18](#_Toc486600144)

[II.2 Intégration continue 19](#_Toc486600145)

[II.2.1 Fondamentaux sur l'intégration continue 20](#_Toc486600146)

[II.2.2 Outils d’intégration continue et de qualimétrie 20](#_Toc486600147)

[II.3 Méthodologie de développement 21](#_Toc486600148)

[II.3.1 Processus de développement 21](#_Toc486600149)

[II.3.2 Les rôles 23](#_Toc486600150)

[II.3.3 SCRUM dans notre projet 23](#_Toc486600151)

[II.3.4 Présentation d’UML 24](#_Toc486600152)

[II.4 Conclusion 25](#_Toc486600153)

[CHAPITRE III SPECIFICATION FONCTIONNELLE 26](#_Toc486600154)

[III.1 Généralités 27](#_Toc486600155)

[III.1.1 Règles générales 27](#_Toc486600156)

[III.1.2 Règles liées à l’accès aux écrans 27](#_Toc486600157)

[III.1.3 Règles liées à la validation des formulaires 27](#_Toc486600158)

[III.1.4 Règles liées aux actions sur les objets 28](#_Toc486600159)

[III.2 Fonctionnalités de l’application 29](#_Toc486600160)

[III.2.1 Fonctionnalité « Envoi et réception de SMS » 29](#_Toc486600161)

[III.2.2 Fonctionnalité « Gestion des groupes bancaires » 31](#_Toc486600162)

[III.2.3 Fonctionnalité « Gestion des banques » 32](#_Toc486600163)

[III.2.4 Fonctionnalité « Gestion des agences » 33](#_Toc486600164)

[III.2.5 Fonctionnalité « Gestion des clients » 34](#_Toc486600165)

[III.2.6 Fonctionnalité « Gestion des comptes » 35](#_Toc486600166)

[III.2.7 Fonctionnalité « Gestion des profils » 36](#_Toc486600167)

[III.2.8 Fonctionnalité « Gestion des utilisateurs » 38](#_Toc486600168)

[III.2.9 Fonctionnalité « Gestion des numéros courts » 39](#_Toc486600169)

[III.2.10 Fonctionnalité « Gestion des services » 40](#_Toc486600170)

[III.2.11 Fonctionnalité « Gestion des contrats » 45](#_Toc486600171)

[III.2.12 Fonctionnalité « Chargement des données » 47](#_Toc486600172)

[III.2.13 Fonctionnalité « Consultation des SMS » 48](#_Toc486600173)

[III.2.14 Fonctionnalité « Statistiques des SMS » 48](#_Toc486600174)

[III.2.15 Fonctionnalité « Monitoring des processus » 48](#_Toc486600175)

[III.3 Conclusion 50](#_Toc486600176)

[CHAPITRE IV CONCEPTION DU SYSTEME 51](#_Toc486600177)

[IV.1 Spécification fonctionnelle détaillée 52](#_Toc486600178)

[IV.1.1 Catalogue 52](#_Toc486600179)

[IV.1.2 Définition des acteurs 52](#_Toc486600180)

[IV.1.3 Etude préliminaire 53](#_Toc486600181)

[IV.1.4 Diagramme de cas d’utilisation et leur descriptions 55](#_Toc486600182)

[IV.1.5 Diagramme de classe 61](#_Toc486600183)

[IV.2 Conception détaillée 63](#_Toc486600184)

[IV.3 Architectures générale du système 67](#_Toc486600185)

[IV.3.1 Communication via un modem GSM 69](#_Toc486600186)

[IV.3.2 Communication via une liaison spécialisée 72](#_Toc486600187)

[IV.4 Architecture technico-fonctionnelles 73](#_Toc486600188)

[IV.4.1 Architecture fonctionnelle 73](#_Toc486600189)

[IV.4.2 Architecture logicielle 76](#_Toc486600190)

[IV.5 Conclusion 77](#_Toc486600191)

[CHAPITRE V IMPLEMENTATION DE LA SOLUTION 78](#_Toc486600192)

[V.1 Choix des outils et des technologies 79](#_Toc486600193)

[V.1.1 Architecture de technique détaillée de la plateforme 79](#_Toc486600194)

[V.1.2 Choix des outils et mise en place l’environnement de développement 79](#_Toc486600195)

[V.1.3 Choix des outils et mise place de l’environnement de production 83](#_Toc486600206)

[V.2 Aperçu des résultats 83](#_Toc486600209)

[V.2.1 Authentification 83](#_Toc486600210)

[V.2.2 Définition d’un contrat 84](#_Toc486600211)

[V.2.3 Planifier un message PUSH 85](#_Toc486600212)

[V.2.4 Historique des messages 86](#_Toc486600213)

[V.3 Conclusion 86](#_Toc486600214)

[Annexes 89](#_Toc486600215)

# Sigles et abréviations

Nous présentons ici certains sigles et abréviations utilisés dans le document.

|  |  |
| --- | --- |
| Abréviation | Signification |
| API | Application Programming Interface |
| BTS | Base transceiver system |
| BSC | Base system Controller |
| DDOS | Distributed Denial of Service |
| ESME | Extended Short Message Entity |
| GMSC | Gateway Short message service center |
| GDC | Global Delivery System |
| GSM | Global System Mobile |
| JEE | Java Edition Enterprise |
| MS | Mobile station |
| MO | Mobile Originating |
| MT | Mobile Terminating |
| SMSC | Short message service center |
| SS7 | Signaling System #7 |
| SMMP | Short Message Peer to Peer. |
| SDCCH | Stand-Alone Dedicated Control Channel |
| UIT | Union Internal of Telecommunication |
| VPN | Virtual Private Network |

# Table des figures

[Figure 1 : Répartition des employés d’ATOS dans le monde 4](#_Toc486661857)

[Figure 2 : Historique du Groupe Atos 5](#_Toc486661858)

[Figure 3 : Les activités d’Atos GDC Sénégal 7](#_Toc486661859)

[Figure 4 : Les secteurs d'activité du GDC Atos Sénégal 7](#_Toc486661860)

[Figure 5 : Organigramme GDC Sénégal 8](#_Toc486661861)

[Figure 6 : Architecture technique du réseau GSM 13](#_Toc486661862)

[Figure 7 : Envoi de message du TE vers le SMSC 14](#_Toc486661863)

[Figure 8 : Envoi de SMS du SMSC vers le mobile destinataire 15](#_Toc486661864)

[Figure 9 : Modem GSM 16](#_Toc486661865)

[Figure 10 : La passerelle SMS reliant 2 SMSC 17](#_Toc486661866)

[Figure 11 : Procédure d'authentification dans le réseau GSM 19](#_Toc486661867)

[Figure 12 : Intégration continue 20](#_Toc486661868)

[Figure 13 : Les contraints d'un projet. 21](#_Toc486661869)

[Figure 14 : Cycle de vie de SCRUM 22](#_Toc486661870)

[Figure 15: Diagramme de Gantt du SMS Banking 25](#_Toc486661871)

[Figure 16 : Diagramme de packages du système SMS Banking 54](#_Toc486661872)

[Figure 17 : Sous-système « Gestion des utilisateurs » 55](#_Toc486661873)

[Figure 18 : Cas d'utilisation "Authentification" 55](#_Toc486661874)

[Figure 19 : Diagramme de séquence du cas « authentification » 57](#_Toc486661875)

[Figure 20 : Sous-système "Gestion des groupe bancaire" 57](#_Toc486661876)

[Figure 21 : Diagramme de séquence "créer un groupe bancaire" 59](#_Toc486661877)

[Figure 22 : Cas d'utilisation "Supprimer groupe bancaire" 60](#_Toc486661878)

[Figure 23 : Diagramme de séquence du cas « Supprimer groupe bancaire » 61](#_Toc486661879)

[Figure 24 : Diagramme de classe 62](#_Toc486661880)

[Figure 25 : Modélisation de la base de données du système.](#_Toc486661881)

[Figure 26 : Modèle relationnel de la base de données 67](#_Toc486661882)

[Figure 27 : Architecture générale du système 68](#_Toc486661883)

[Figure 28 : Liaison entre le SMS-C de l’opérateur et du serveur Kannel via un modem 69](#_Toc486661884)

[Figure 29 : Architecture de la solution première solution 70](#_Toc486661885)

[Figure 30 : Architecture de la deuxième solution 71](#_Toc486661886)

[Figure 31 : Liaison du serveur Kannel au SMSC de l’opérateur via une liaison spécialisée 72](#_Toc486661887)

[Figure 32 : Architecture du SMS en mode PULL 73](#_Toc486661888)

[Figure 33 : Architecture d’un message initié par un utilisateur habilité. 75](#_Toc486661889)

[Figure 34 : Architecture d'un SMS déclenché par un évènement 76](#_Toc486661890)

[Figure 35 : Architecture d'un message déclencher par un évènement 76](file:///C:\Users\A661546\Desktop\Atso_SMS_Banking_Stage_2017\Memoire%20woury\Memoire%20Mamadou%20Oury%20DIALLO%20DIC3_2017_V2.docx#_Toc486661891)

[Figure 36 : Les couches de l'application 77](#_Toc486661892)

[Figure 37:Architecture détaillée de la plateforme 79](#_Toc486661893)

[Figure 38 : Interface d'authentification 84](#_Toc486661894)

[Figure 39 : Création d'un contrat 84](#_Toc486661895)

[Figure 40 : Interface de planification de message PUSH 85](#_Toc486661896)

[Figure 41 : Historique de SMSPULL 86](#_Toc486661897)

[Figure 42 : connexion de Jenkins à github 92](#_Toc486661898)

[Figure 43 : Importation du projet dans Jenkins 93](#_Toc486661899)

# Table des tableaux

[Tableau 1 : Les commandes AT: 16](#_Toc486661900)

[Tableau 2 : Messages d'erreur à envoyer 31](#_Toc486661901)

[Tableau 3 : Caractéristiques d'un groupe bancaire 31](#_Toc486661902)

[Tableau 4 : Caractéristique d'une banque 32](#_Toc486661903)

[Tableau 5 : Caractéristiques d'une agence 33](#_Toc486661904)

[Tableau 6 : Caractéristique d’un client 34](#_Toc486661905)

[Tableau 7 : Caractéristiques d'un compte bancaire 35](#_Toc486661906)

[Tableau 8 : Caractéristiques des profils 36](#_Toc486661907)

[Tableau 9 : Exemple de paramétrage de profil 37](#_Toc486661908)

[Tableau 10 : Caractéristiques d'un utilisateur 38](#_Toc486661909)

[Tableau 11 : Caractéristiques d'un numéro court 39](#_Toc486661910)

[Tableau 12 : Caractéristique des numéros courts 40](#_Toc486661911)

[Tableau 13 : Syntaxe des messages envoyés par le client 41](#_Toc486661912)

[Tableau 14 : Liste des services de base 43](#_Toc486661913)

[Tableau 15 : Attributs de la table gestion de contrat 46](#_Toc486661914)

[Tableau 16 : Acteurs du système 52](#_Toc486661915)

[Tableau 17 : Les acteurs et leurs rôles 52](#_Toc486661916)

[Tableau 18 : Description textuelle "Authentification" 56](#_Toc486661917)

[Tableau 19 : Description textuelle "créer un groupe bancaire" 58](#_Toc486661918)

[Tableau 20 : Description des classes 63](#_Toc486661919)

[Tableau 21 : Passage du modèle objet au modèle relationnel 66](#_Toc486661920)

[Tableau 22 : Tableau récapitulatif 71](#_Toc486661921)

[Tableau 23 : Tableau comparatif SGBD 80](#_Toc486661922)

# RESUME

Ce document est un mémoire de fin de cycle en vue de l’obtention du Diplôme d’Ingénieur de Conception en Télécommunications et Réseaux. Il présente la réalisation d’un projet effectué dans le cadre d’un stage à AtoS Sénégal portant sur la« Conception et l’Implémentation d’une Solution de SMS Banking en JEE».

Nous assistons à un développement fulgurant de la téléphonie mobile qui réduit les contraintes géographiques et les coûts des transactions tout en offrant aux banques ou structures financières une stratégie d’expansion faible. Afin de s'adapter à la révolution générée par ces technologies et systèmes d'information, les structures financières (et notamment les banques) développent de plus en plus de projets d’e*-Banking[[1]](#footnote-1)* ces dernières années*.* On retrouvele ***SMS Banking***qui est une branche qui s’inscrit dans le même contexte que l’*e-Banking* permettant aux clients d’une structure financière[[2]](#footnote-2) d’avoir accès aux informations relatives à leurs comptes. Le ***SMS*** *(Short Message Service) Banking,*dans un cadre de réinvention de l’expérience client et de la transformation digitale, ne cesse de se développer ces dernières décennies.

Conscient de cette expansion, l’entreprise Atos Sénégal a décidé d’exploiter cette nouvelle branche par la mise en place d’une solution offrant des services permettant ainsi aux structures financières de maintenir leur croissance et d’augmenter leur clientèle tout en augmentant la satisfaction de cette dernière.

Ainsi, il sera question dans ce mémoire de concevoir et d’implémenter une solution de *SMS Banking* intégrant la technologie JEE (*Java Entreprise Edition)* et le SMS. La réalisation de cette solution passera par la mise en place :

* D’une application web modulaire, générique et paramétrable (permettant aux structures financières d’offrir une gamme de services à leurs clientèles)
* D’un serveur SMS pour permettre aux clients de ces structures d'effectuer leurs opérations bancaires à partir d'un format SMS bien défini.
* D’un serveur d’intégration continue et de qualimétrie.

Mots clés : JEE, SMS, 2G, 3G, GSM.

# ABSTRACT

This document is a dissertation in order to obtain the diploma of design engineer in telecommunications and networks. It presents a project carried out during an internship in AtoS Senegal on the “Design and implementation of a SMS Banking in JEE”.

We see a rapid development of mobile telephony which reduces geographic constraints and transaction costs while offering banks or financial structures a weak expansion strategy. In order to adapt to the revolution caused by these technologies and information systems, the banking sector or banking structure in recent years is developing more and more e-Banking projects*. SMS Banking* is a branch of e-Banking that allows customer of a bank or financial structure to access their account information. The *SMS Banking*, in reinvention of the customer experience and digital transformation, has not ceased to develop in recent decades. Aware of this expansion, Atos Senegal decided to develop this new service.

In this dissertation, we will discuss the design and implementation of a *SMS Banking* solution using JEE (Java Enterprise Edition) and SMS technology. The realization of this solution will require the implementation of:

* A modular, generic and configurable web application (allowing financial structure to offer a range of services to their customer)
* A SMS server to allow customer of these structures to use their banking operations from a well-defined SMS format
* A server for continuous integration and qualimetry

Keywords: JEE, SMS, 2G, 3G, GSM.

# INTRODUCTION

Dans un contexte économique actuellement qualifié de dynamique, la survie des banques se présente dans leur capacité à s’adapter rapidement aux changements et à innover. Il devient ainsi primordial pour les banques ou structures bancaires de développer des stratégies permettant de maintenir leur croissance et d’augmenter leurs clientèles. Ceci se faisant en augmentant la satisfaction de leurs clients qui va conduire à un meilleur taux de rétention.

Il est fort important de constater qu’au cours de ces dernières années les technologies mobiles ont créées des conditions de marché concurrentiel. Les technologies mobiles sont au centre des préoccupations des utilisateurs. Ces dernières, grâce à un accès aux informations en temps réel, quel que soit l’endroit où on se trouve, constitue l’un de produits les plus en résonance avec la tendance de la société.

En effet, selon les estimations de l'International Télécommunication Union (ITU), dans le monde fin 2016, ce sont près de 7,4 milliards d'abonnements mobiles qui étaient souscrits, soit quasiment la totalité de la population mondiale [W1]. Cela correspond ainsi à un taux de pénétration de 99,7%. De plus, 5,8 milliards de ces abonnements ont été souscrits dans des pays en développement.

Au Sénégal, selon le dernier rapport de l’ARTP en juin 2016 [W2], le mobile compte 15 765 524 lignes sur 14 millions d’habitants. Et durant les trois derniers mois, il y a eu un peu plus de 410 millions de SMS (Short Message Service) envoyés. En outre, le phénomène des multi SIM (usagers possédant plusieurs abonnements) fait que le taux de pénétration dépasse les 100 % pour atteindre 116,71 %. Sans oublier que 75 % de la population sénégalaise utilise le téléphone portable pour communiquer via SMS.

Le choix du SMS ne résulte pas du hasard. En fait, il suffit de voir ces statistiques pour savoir qu’aujourd’hui le SMS est un outil performant et est devenu incontournable. Dès lors, en s’appuyant sur la technologie SMS, il est possible de créer aujourd’hui un réseau permanent entre le client et la banque : ***SMS Banking***.

Le ***SMS Banking***  est une branche de l*’e-Banking* qui combine le SMS et le téléphone mobile. Il permet ainsi aux clients d’une banque ou structure financière d’effectuer les différentes opérations bancaires depuis n’importe quel téléphone portable ou tablette et à tout instant sans aucun déplacement. Le client pourra donc effectuer diverses opérations sur un ou plusieurs de ses comptes bancaires par simple envoi de SMS et donc sans être contraint à se déplacer.

Nous essayerons d’atteindre les objectifs qui nous ont été fixés tout en appliquant une démarche rigoureuse d’analyse conceptuelle et de développement dans ce document qui s’articule autour des chapitres suivants :

* Un premier chapitre « PRESENTATION GENERALE » dans lequel nous présentons la structure d’accueil, le contexte, la problématique ainsi que les objectifs du sujet.
* Un deuxième chapitre « GENERALITES » définissant les différents concepts généraux apparaissant dans notre architecture ainsi que les outils utilisés dans la réalisation du sujet.
* Un troisième chapitre « SPECIFICATION FONCTIONNELLE » dans lequel nous présentons les fonctionnalités du système, leur cas d’utilisation ainsi que les règles métiers qui sont associés. Ce chapitre constitue la colonne vertébrale de notre sujet.
* Un quatrième chapitre « CONCEPTION DU SYSTEME » qui présente notre travail d’analyse et de conception à travers des spécifications des modèles nécessaires à chaque étape. Il présente également les différentes architectures du système.
* Un cinquième chapitre « REALISATION DE LA SOLUTION » qui présente les outils que nous avons utilisés pour la réalisation du système *SMS Banking* ainsi le résultat de la solution proposée.

# PRESENTATION GENERALE

**Résumé** :

*Dans ce premier chapitre, nous présentons la structure au sein de laquelle nous avons effectué notre projet de fin de formation : AToS Sénégal. Ensuite, nous traitons du sujet et définissons le contexte ainsi que la problématique dans lesquels ce travail a été réalisé et terminons par présentant les différents objectifs à atteindre pour ce projet.*

**Sommaire**

I.1 Présentation de la structure d’accueil 4

I.1.1 AtoS en chiffre 4

I.1.2 Historique 4

I.1.3 Domaine d’activité 5

I.1.4 Partenaires 6

I.1.5 GDC Sénégal 6

I.2 Présentation du sujet 8

I.2.1 Contexte 8

I.2.2 Problématiques 9

I.2.3 Objectifs 10

I.3 Conclusion 10

### Présentation de la structure d’accueil

#### AtoS en chiffre

Atos est l'un des dix plus grands acteurs des Entreprises de Services du Numérique (ESN) au niveau mondial. Avec un chiffre d'affaires annuel de 8,8 milliards d'euros, actuellement la société emploie environ 86 600 personnes dans 66 pays, principalement en France, en Allemagne aux Pays-Bas, en Espagne, au Royaume-Uni, en Inde, au Maroc et depuis décembre 2014 au Sénégal. La *Figure* 1 nous donne une idée de la répartition des salariés d’Atos à travers le monde.



Figure 1 : Répartition des employés d’ATOS dans le monde

#### Historique

L’histoire du groupe Atos a été rythmée par des acquisitions et des fusions et l’une des dernières en date est l’acquisition de la société française Bull comme l’illustre la *Figure* 2.

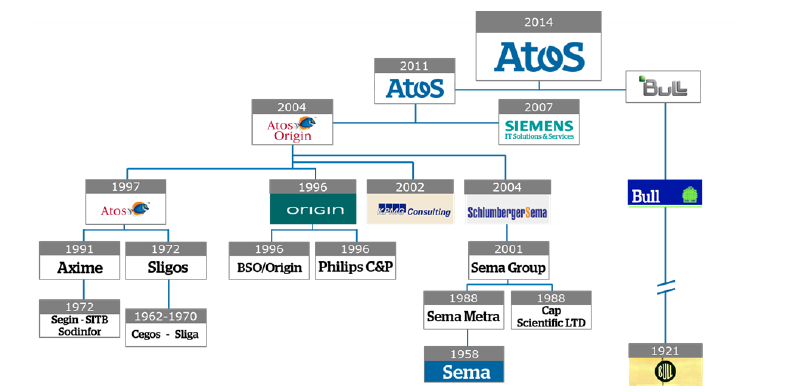


Figure 2 : Historique du Groupe Atos

#### Domaine d’activité

Côté sur le marché Euronext Paris, le Groupe exerce ses activités sous les marques:

* Le conseil et les services technologiques :

Atos accompagne ses clients en les aidant à gagner en rentabilité et en efficacité grâce aux technologies et à l’innovation.

* L’intégration de systèmes :

Consiste à réunir au sein d'un même système d'information, des parties développées de façon séparées. Atos fournit des systèmes parfaitement intégrés.

* L’infogérance:

Consiste à déléguer à un prestataire extérieur de la gestion, l’exploitation, l’optimisation et la sécurisation du système d’information d’une société. Atos gère et transforme ainsi toutes les opérations informatiques de ses clients, y compris l’ensemble de leurs systèmes informatiques et de traitement de données.

* Services :

Transactionnels et de paiement : Worldline est la filiale d’Atos consacrée aux services de paiement en ligne. Les activités de Worldline sont organisées autour de trois axes :

* Merchant Services et Terminaux,
* Mobility et e-Transactional Services et
* Financial Processing et Software Licensing.

En 2013, les activités de Worldline, au sein du Groupe Atos, ont généré un chiffre d’affaires de 1,1 milliard d’euros.

#### Partenaires

Atos s'appuie sur un réseau de partenaires stratégiques afin de combiner leurs expertises et fournir à ses clients des solutions innovantes.

Les partenaires d’Atos sont les suivants :

* **EMC**² : Une technologie largement utilisée pour équiper leurs clients de services d'infrastructure robustes et fiables.
* **Microsoft** : La combinaison de solutions souples et innovantes pour anticiper et s'adapter aux évolutions des activités des entreprises.
* **Oracle** : Des solutions basées sur les systèmes d'entreprise Oracle et les processus business qu'ils supportent.
* **SAP** : Atos est partenaire de SAP Global Services & Hosting.
* **Siemens** : un fabricant de technologie
* **VMware** : Solutions cloud.
* **HP** : Des solutions pour aider ses clients dans le développement de CA, la réduction des coûts et le gain d'efficacité.
* **IBM** : Des solutions couvrant Mainframes et serveurs Unix & Intel et leur association à des solutions logicielles et de stockage.

#### GDC Sénégal

Le Global Delivery Center du Sénégal (GDC SN) a été mis en place en décembre 2014. C’est un centre de compétence technique capable de fournir des services informatiques à des entreprises d’Afrique Subsaharienne et plus globalement à tous les clients Atos qui sont répartis dans les quatre coins du monde.

Ce phénomène connu sous le terme d’offshoring consiste à délocaliser des services informatiques d’une entreprise vers un pays à bas coûts salariaux. Concrètement les entreprises des pays occidentaux font exécuter une partie de leurs travaux liée à des sujets technologiques ou des sujets de production par des ressources qui sont dans des pays où les coûts sont moindres.

Ce centre s’activera dans l’intégration de solutions informatiques, du développement et la maintenance de systèmes informatiques, comme l’illustre la *Figure* 3 :

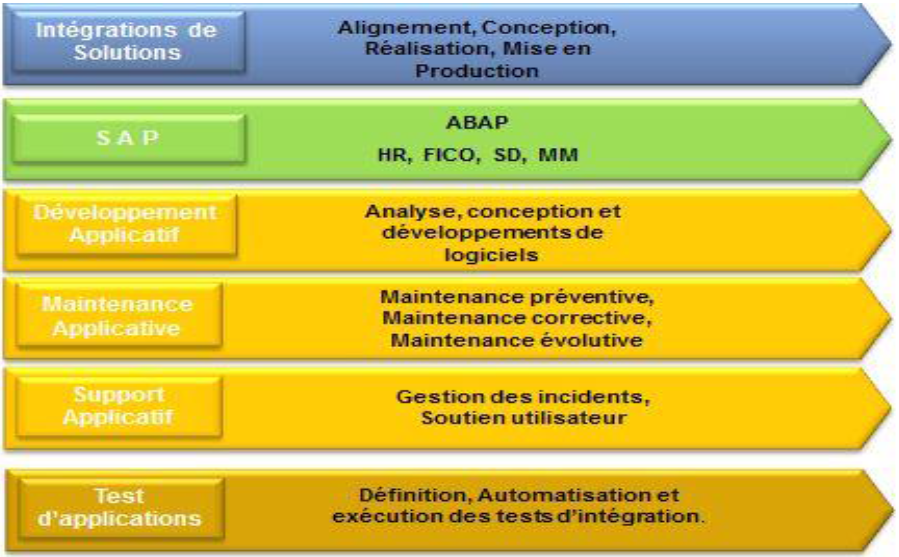


Figure 3 : Les activités d’Atos GDC Sénégal

Ses métiers cibles sont les télécoms, l’industrie, la banque, l’assurance, le secteur public et la distribution. La *Figure 4* suivante présente les secteurs d’activités du GDC Sénégal.

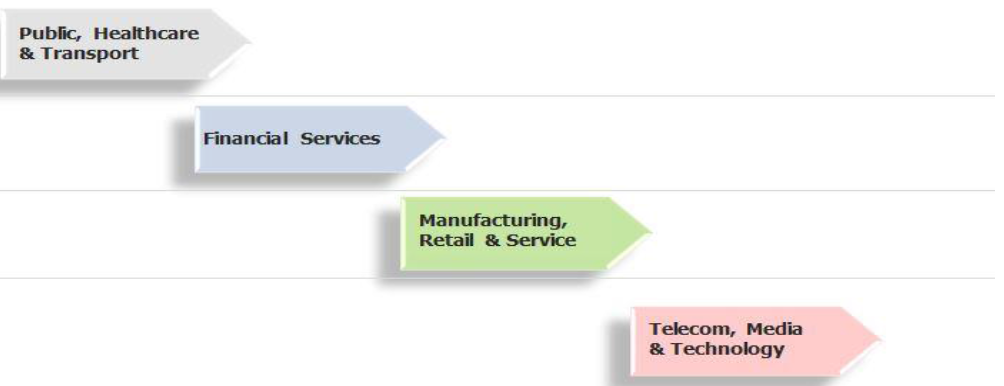


Figure 4 : Les secteurs d'activité du GDC Atos Sénégal

Le GDC Sénégal a une hiérarchie et est organisé comme le montre la *Figure 5* suivante :

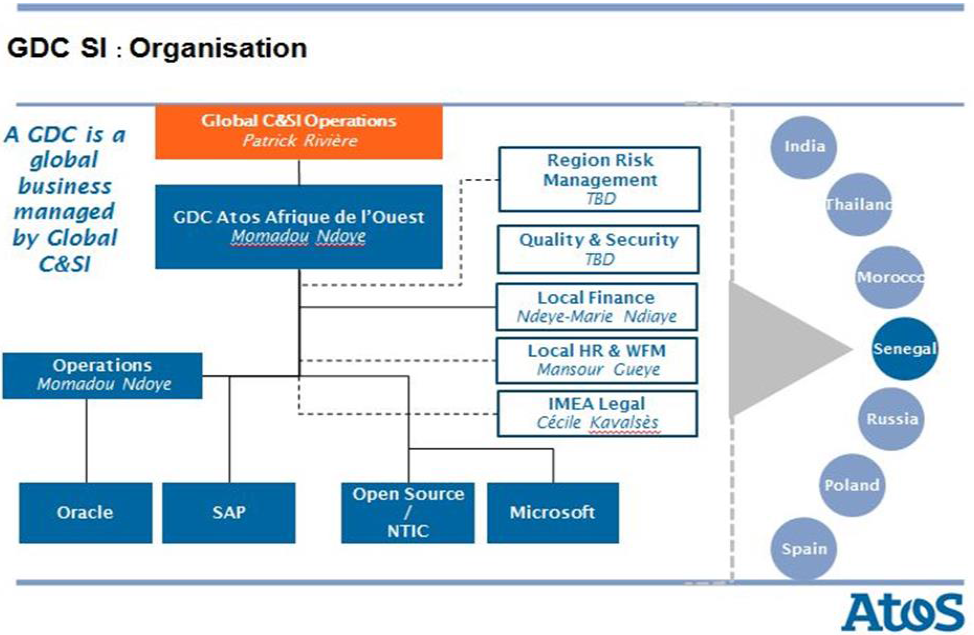


Figure 5 : Organigramme GDC Sénégal

### Présentation du sujet

Dans cette section, nous présentons explicitement le sujet de notre mémoire s’intitulant « **Conception et Implémentation d’une solution de *SMS Banking* en JEE**». Il sera question de présenter d’abord le contexte du projet ensuite énumérer la problématique et enfin exposer les différents objectifs fixés.

#### Contexte

Dans le cadre du processus de bancarisation[[3]](#footnote-3), de digitalisation[[4]](#footnote-4) et du renouvèlement de l’expérience client, de plus en plus de banques tendent à se doter de moyens leur permettant d’accroître leur clientèle, notamment dans les zones ne disposant pas d’agences à proximité pour fournir leurs services. C’est ainsi qu’est apparue la solution d’*e-Banking* qui vient répondre à ce besoin car, elle offre la possibilité d’accéder à ces services via internet et donc évite les clients à se déplacer. Malheureusement cette solution présente des limites dont principalement la non couverture des zones ne disposant tout simplement pas d’accès à Internet. C’est alors que plusieurs entreprises se sont tournées vers des solutions utilisant le SMS : ***Le SMS Banking****. En effet, ce* choix présente de nombreux avantages que sont :

* La convenance : le *SMS Banking* offre plus de souplesse aux clients qui effectuent les transactions bancaires via leur téléphone mobile à temps réel sans se déplacer ou faire la queue.
* Le traitement instantané des demandes des clients de n’importe où et à n’importe quel moment,
* Le gain de temps : les clients ne sont pas obligés de se déplacer donc moins de file d’attente au niveau des agences.
* L’augmentation de la satisfaction des clients donc meilleurs taux de rétention,
* La simplification de l’interaction des clients avec la banque,
* Le SMS est lu presque instantanément par les destinataires,
* De nouveaux revenus grâce à la facturation de ces nouveaux services,
* Coûts abordables de la communication via SMS favorisant son utilisation.

Il faut noter que malgré ces avantages, le *SMS Banking* présente des limites liées à la couverture réseaux.

Compte tenu du développement que connaît aujourd’hui le marché de la téléphonie, ainsi que du développement de solutions *de SMS Banking* par de nombreuses organisations, ajouter à cela les nombreux avantages du *SMS Banking,* AtoS Sénégal a décidé de développer de telles solutions et d’être parmi les pionniers de la sous-région.

C’est dans ce cadre qu’il nous a été confié la mise en place d’une solution de *SMS Banking* pour permettre aux banques ou structures financières de répondre aux besoins de ses clients et satisfaire ainsi au mieux leurs exigences.

#### Problématiques

Dans un but de favoriser l’inclusion bancaire, les banques ne cessent de produire des solutions de plus en plus sophistiquées. La plupart des clients de ses banques, dont le nombre ne cesse de croître, se trouve dans des zones rurales dont l’accès aux agences est difficile (causé par des aléas de la nature). Il faut également noter que de nombreux clients de ces banques passent un temps important dans les trajets « domiciles-agence » ce qui se traduit par une perte de temps énorme.

Certains clients des banques se trouvent dans des zones où ils n’ont pas accès à internet pour bénéficier de la panoplie des services bancaires offerts (Web Banking[[5]](#footnote-5), etc.). Ils se sentent ainsi défavorisés, commencent à ressentir du mécontentement qui, à la longue les amène à se tourner vers d’autres banques, etc. Dès lors la question qui se pose est de savoir comment s’appuyer sur la technologie SMS pour maintenir et renforcer la relation « banque -clients » ?

#### Objectifs

Nous avons comme objectif la mise en place d’une application permettant aux banques ou structures financières d’offrir une gamme de services à leurs clientèles. Ces dits services permettront d’effectuer des opérations bancaires à partir d'un format SMS bien défini.

Ainsi, les principaux objectifs sont :

* En Back-end, mettre en place tous les services permettant le traitement des SMS reçus (PULL) ou à envoyer (PUSH).
* En Front-end, développer une interface web en JEE générique permettant le paramétrage et l’administration de l’application.
* Mettre en place un serveur d’intégration continue et de qualimétrie.

### Conclusion

Ce premier chapitre, nous a permis de présenter la structure d’accueil, de contextualiser le sujet et de dégager la problématique ainsi que les objectifs à atteindre.

Dans le chapitre suivant, nous allons présenter le *SMS Banking* et les différents concepts utilisés dans le document.

# GENERALITES

**Résumé :**

*Dans ce second chapitre, nous présentons tout d’abord les différents concepts réseaux utilisés dans le cadre de ce projet, puis présentons la méthodologie de développement ainsi le langage de modélisation adoptées.*

**Sommaire**

II.1 Concepts réseaux 12

II.1.1 Architecture technique du réseau GSM 12

II.1.2 Présentation du SMSC dans le cas des réseaux 2G/3G 13

II.1.3 Fonctionnement des SMS : Mode PULL, Mode PUSH 13

II.1.4 Principe de mise en place d’un ESME 15

II.1.5 Notion de sécurité GSM 18

II.2 Intégration continue 19

II.2.1 Fondamentaux sur l'intégration continue 20

II.2.2 Outils d’intégration continue et de qualimétrie 20

II.3 Méthodologie de développement 21

II.3.1 Processus de développement 21

II.3.2 Les rôles 23

II.3.3 SCRUM dans notre projet 23

II.3.4 Présentation d’UML 24

II.4 Conclusion 25

### Concepts réseaux

Dans cette partie, nous abordons les concepts techniques du SMS dans l’ordre suivant :

* Nous présentons tout d’abord l’architecture technique du réseau GSM
* Puis définissons le processus de l’envoi du SMS

#### Architecture technique du réseau GSM

Les opérateurs réseaux utilisent un certain nombre d’équipements pour assurer la transmission des appels téléphoniques, des SMS, des MMS, etc. Les différents éléments techniques sur lesquels les opérateurs se basent sont au nombre de trois. Il s’agit entre autre:

* Du terminale de l’abonné appelé «MS» (*Mobile Station*).Le « MS » permet d’effectuer un ensemble de tâches, mais pour ce qui nous concerne, il peut envoyer et recevoir des SMS.
* Du réseau d’accès composé d’antennes (BTS, *Base Tranceiver Station*) et de contrôleurs d’antenne (BSC, *Base Station Controler*) qui permettent aux terminaux de se connecter (d’accéder) au réseau par radio.
* Du réseau cœur : Il se connecte au réseau d’accès et reçoit les demandes de l’utilisateur. Il est composé de plusieurs MSC (*Mobile Switching Center*), chacune couvrant une zone géographique bien définie et de base de données techniques **[W3]**. Il assure la vérification des droits d’accès des abonnés aux différents services en interrogeant les éléments tiers. Il assure également l’itinérance et dans certains cas, délègue-la réalisant du service demandé à une plate-forme tierce (par exemple le SMS-C dans le cadre du SMS).

Dans la suite nous allons étudier le SMSC (*SMS Center*) qui constitue l’élément phare dans le traitement des SMS au sein du réseau de l’opérateur. La *Figure 6* suivant montre l’architecture technique du réseau GSM.

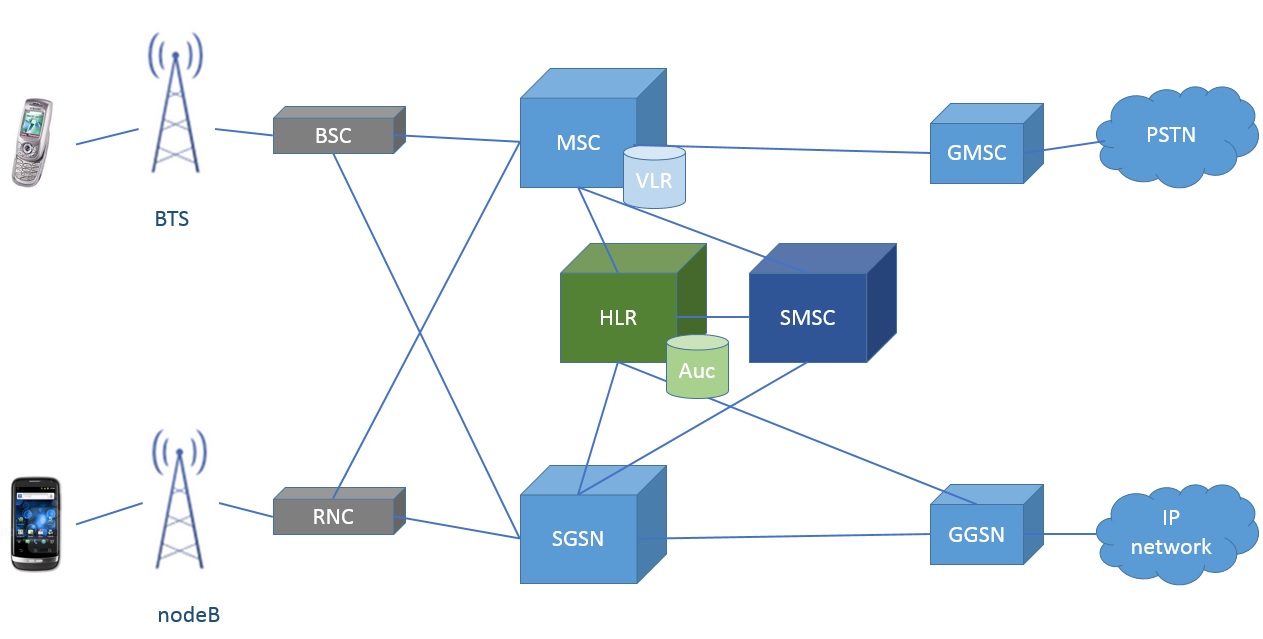


Figure 6 : Architecture technique du réseau GSM

#### Présentation du SMSC dans le cas des réseaux 2G/3G

Dans le réseau 2G/3G il apparait un composant dénommé SMSC pour Short Message service Center (Centre de service des messages courts en français) qui permet de gérer le transfert de messages(SMS) limités à 160 caractères entre téléphones mobiles.

Le SMSC est connecté au MSC en utilisant le protocole de signalisation SS7.Il est identifié dans le réseau par un numéro de téléphone afin que le MSC puisse lui transmettre les messages provenance des abonnés **[W4].** Il fonctionne en mode « *store and forward* ».Il existe cependant un seul SMSC par réseau GSM.

Le SMSC peut être connecté à des applications pouvant recevoir ou envoyer des SMS, les services SMS. Le protocole a utilisé dans ce cas est SMPP (*Short Message Peer to Peer*).

#### Fonctionnement des SMS : Mode PULL, Mode PUSH

Il existe trois classes de SMS **[W5]** :

* Les messages venant d'un mobile (Mode PULL) : l'abonné envoie un message. On parle alors de SMS-MO, MO pour Mobile Originating.
* Les messages arrivant à un mobile (Mode PUSH): l'abonné reçoit un message. On parle alors de SMS-MT, MT pour Mobile Terminating.
* Les messages en diffusion (broadcast) : l'opérateur envoie le même message à un ensemble d'abonnés localisés dans une zone géographique.

1. Mobile Originated (MO)

Lors de l’envoi d’un SMS (SMS-MO) le terminal code le corps du message et l’envoi au GMSC avec deux informations supplémentaires que sont : le numéro du destinataire et le numéro du SMS-C à utiliser. Ce dernier est configuré dans le téléphone et permet au GSMC de relayer le SMS au SMS-C. Si le message n’a pas pu être délivré au SMSC, le réseau envoie un acquittement négatif. Si la première tentative échoue, le message est stocké et d’autres tentatives sont réalisées. Au bout d’un certain nombre d’échec ou après un certain délai, le message est effacé. La *Figure 7* suivante illustre ce mécanisme.

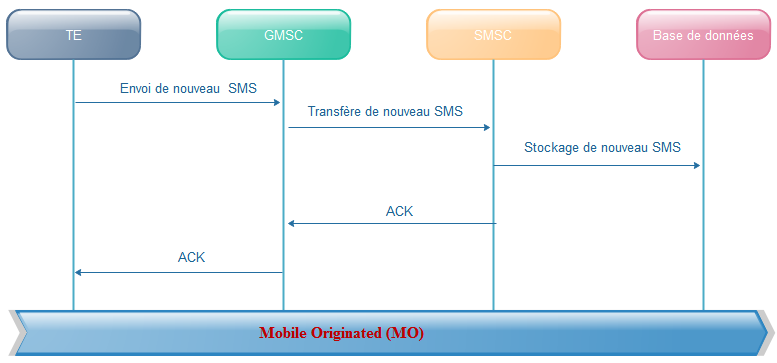


Figure 7 : Envoi de message du TE vers le SMSC

1. Mobile Terminated(MT)

Le SMSC ayant reçu le message l’enregistre dans sa base de données puis tente ensuite de l’envoyer au destinataire par le biais du GLMSC. Le mobile récepteur après avoir reçu le message, envoie un acquittement au SMSC lui notifiant qu’il a bien reçu le message et ce dernier efface alors le message de sa base de données. La *Figure 8* ci-dessous le principe d’envoi de SMS du SMS-C vers le client.

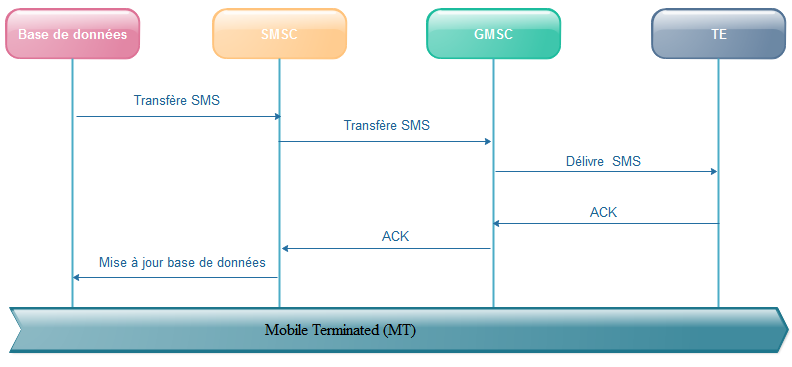


Figure 8 : Envoi de SMS du SMSC vers le mobile destinataire

#### Principe de mise en place d’un ESME

Il existe plusieurs façons de concevoir une application SMS en général, soit en utilisant un dispositif sans fil, utilisé une API ou se connecter directement au SMSC de l’opérateur (qui est la méthode utilisée lors de notre déploiement).

1. Utiliser un dispositif sans fil

Le moyen le plus rentable de mettre en place des ESME légers dans les organisations est d’utiliser un équipement tel qu’un modem GSM qui est connecté au serveur d’application via le port série et qui permet de transférer les données sur le réseau GSM **[B1]**. L’avantage de cette solution est sa modularité : Si le ESME tombe en panne alors le système de la banque ne sera pas affecté. L’inconvénient est que le matériel ne supporte pas un grand flux de données.

1. Se connecter au SMSC directement

Le moyen le plus rapide de transférer les SMS via le réseau GSM est de se connecter directement au SMSC ou à la passerelle SMS **[B2]**. Cette méthode exige que l’application SMS soit intégrée au réseau de l’opérateur téléphonique. Le lien de communication entre le logiciel d’application et le SMSC peut s’assurer par un réseau TCP/IP ou une connexion X.25[[6]](#footnote-6). L’avantage est la rapidité de la transmission des SMS. L’inconvénient est que le SMSC est ainsi exposé à toute connexion externe.

1. Utiliser les API

Le moyen le plus répandu pour développer des ESME est l’implémentation des API (*Application Programming Interface*) dans le logiciel de l’application **[B3]**. Une connexion Internet est établie à partir de l’application (le serveur SMS) qui permet de transférer les données du client à la passerelle SMS du provider qui offre le service puis au réseau GSM. L’avantage est que les API permettent l’envoi d’un grand flux de SMS à la fois. L’inconvénient est que les API qui offrent une bonne qualité de services sont payantes.

1. Les modems GSM

Un modem GSM est un équipement de communication sur le réseau GSM qui se comporte comme un modem dial-up. La différence principale entre eux est qu'un modem dial-up envoie et reçoit des données par une ligne téléphonique fixe alors qu'un modem GSM envoie et reçoit des données par les ondes radioélectriques **[B4]**. Il s'agit d'un modem GSM fonctionnant dans les bandes de fréquences 900MHz et 1800MHz. Pour nos différents tests et démos, on a utilisé une clé orange pour faire office de modem GSM. Ce type de modem supporte les commandes AT (*ATtention*.). La *Figure 9* suivante illustre un modem GSM.



Figure 9 : Modem GSM

1. Les commandes AT

Les commandes AT sont définies dans la norme GSM 07.07 (pour les SMS cf. GSM 07.05) **[W6]**. Elles ont été conçues par Hayes pour piloter les modems. Ce sont des commandes qu’on peut directement envoyer au modem. Chaque commande est envoyée sous la forme d'une ligne de texte encodée en ASCII, terminée par le caractère \r seul (code ASCII 13). Toutes les commandes doivent être précédées de AT. Le tableau suivant donne quelques commandes AT. Le *Tableau 1* suivant donne quelques commandes AT.

Tableau 1 : Les commandes AT:

|  |  |
| --- | --- |
| **Commande AT** | **Fonctionnement** |
| **T+CMGF** | Permet de régler le modem GSM en mode texte ou en mode SMS PDU. |
| **AT+CMGS** | Permet d’envoyer le message au numéro précisé. |
| **AT+CMGL** | Permet de lister les messages reçus en mode texte. |
| **AT+CGSN** | Affiche l’identifient IMEI. |

1. Les passerelles GSM

La passerelle est un système matériel et logiciel sécurisé permettant de faire la liaison entre deux réseaux, afin de faire l'interface entre les protocoles de réseaux différents.

Un problème majeur de la messagerie SMS réside dans le fait que les SMSC développés par des sociétés différentes utilisent leur propre protocole de communication, qui est pour la plupart, propriétaire **[W7]**. Par exemple, un SMSC de type NOKIA offrant une interface gérée par le protocole CIMD ne saurait échanger directement avec un SMSC d'un autre vendeur dont l'interface est gérée par le protocole EMI. Pour juguler ce problème, une passerelle SMS est placée entre ces deux SMSCs. Ce dernier agit comme un convertisseur de protocoles entre non seulement des SMSCs, mais aussi avec des dispositifs du protocole TCP/IP comme le montre la *Figure 10* ci-dessous.



Figure 10 : La passerelle SMS reliant 2 SMSC

1. Le protocole de communication SMPP

Le protocole SMPP (*Short message Peer-to-Peer*) est un protocole de l'industrie de télécommunication pour échanger des messages SMS entre entités paires comme des centres de service de messages courts (SMSC) ou entre les SMSC et l’ESME **[W8]**. Il est utilisé pour envoyer et recevoir des messages (SMS-MT) sur les mobiles et les réponses depuis ces mobiles (SMS-MO) en utilisant le réseau GSM, GPRS, etc. Il se base sur le protocole TCP pour la transmission (*Transmitter*) et le protocole IP pour la réception (*Receiver*). La transmission et la réception peuvent s’effectuer simultanément en TCP/IP avec une connexion *Transceiver*.

#### Notion de sécurité GSM

La gestion de la sécurité est primordiale dans toute application. Dans le cadre de notre projet, la majeure partie de la sécurité est au soin de l’opérateur, ainsi nous montrons la façon dont ce dernier le gère ainsi que les potentielles failles qu’il présente.

1. Sécurité du réseau GSM

Le réseau GSM utilise divers algorithmes localisés aussi bien au niveau du MS qu’au niveau du réseau cœur. Les algorithmes utilisés sont ***[B5]*** :

* Dans L’AUC (*Authentification* User Center) :
  + L’algorithme d’authentification A3,
  + L’algorithme de génération de la clé de chiffrement A8,
  + Des clés Ki de client du réseau.
* Dans la BTS (*Base Transceiver System*) :
  + L’algorithme de chiffrement A5 pour le chiffrement des données utilisateur et des données de signalisation.
* Dans le MS :
  + L’algorithme d’authentification A3,
  + L’algorithme de génération de la clé de chiffrement A8,
  + La clé Ki individuel de l’usager.

La sécurité GSM est adressée sur deux plans : authentification et chiffrement. La *Figure 11* suivante montre le processus d’authentification utilisée dans le réseau GSM.

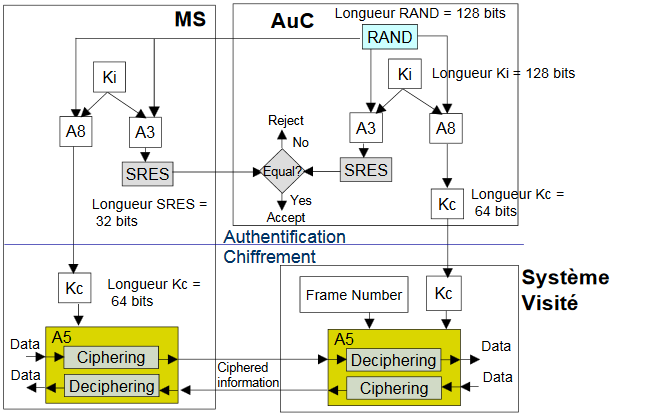


Figure 11 : Procédure d'authentification dans le réseau GSM

1. La vulnérabilité du réseau GSM

Nous étudions ici la partie concernant notre projet à savoir le SMSC.

* **Du point de vue du SMSC**

Les SMSC gèrent les messages SMS en les disposant en file d’attente dont la politique et la taille sont gérées de manière spécifique par chaque opérateur. Cette file d’attente peut être beaucoup plus importante que la capacité mémoire des téléphones mobiles. Lorsque cette mémoire dédiée est saturée, le message reste sur le SMSC jusqu’à ce que l’utilisateur l’efface.

Les SMS sont transmis à travers un canal de contrôle SDCCH (Stand-Alone Dedicated Control Channel) qui lui est alloué pendant une durée de 4 à 5 secondes **[W9]**. Ainsi un type d’attaque DDOS (*Distributed Denial of Service*) peut être utilisé par un pirate pour inonder le buffer par des SMS non signifiants expédiés à un numéro de mobile cible. Cette inondation de SMS entraine le rejet des SMS en surplus du buffer à cause de la limitation de l’espace de la file. Ainsi, le pirate peut récupérer les SMS rejetés, car ils ne sont pas cryptés.

**NB** : Dans le réseau GSM les données sont cryptées uniquement entre le MS et la BTS.

### Intégration continue

Au cours de notre stage, on a eu à travailler dans une équipe chargée de mettre en place une « usine logicielle pour Atos ». Nous avions comme principale tâche d’automatiser l’installation de Jenkins et de sonarQube dans une machine virtuelle (Vagrant) en utilisant l’outil Ansible. L’installation de ces outils va nous permettre d’assurer la non-régression de notre code.

##### Fondamentaux sur l'intégration continue

L'intégration Continue, dans sa forme la plus simple, se compose d'un outil qui surveille les modifications de code dans un gestionnaire de configuration. Dès qu'un changement est détecté, il va automatiquement compiler et tester l’application. Si la moindre erreur arrive alors l'outil va immédiatement avertir les développeurs afin qu'ils puissent tout de suite corriger le problème. L'intégration Continue peut aussi suivre la santé d’un projet en surveillant la qualité du code et les métriques de couverture et ainsi aider à maintenir la dette technique à un niveau bas et à abaisser les coûts de maintenance. L'intégration permet également de réduire les risques en fournissant des retours rapides **[B6].**

La *Figure 12* montre comment nous pouvons représenter le cheminement d'un développement avec et sans intégration continue :

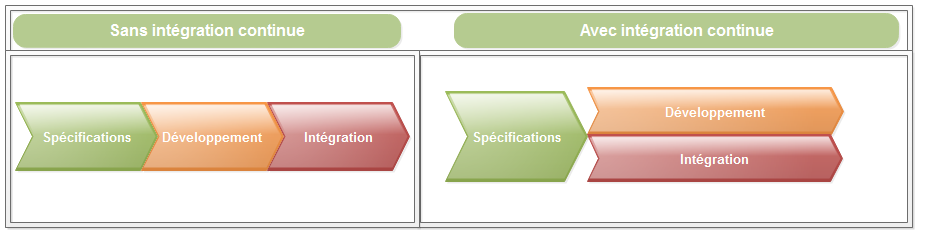


Figure 12 : Intégration continue

. L’intégration continue présente les avantages suivants **[W10]**:

* Centralisation : Permettre à un ensemble de personnes de bénéficier d’information commune.
* Automatisation : permettre d’effectuer l’ensemble des phases de production (compilation, tests, déploiement …) sans ou avec très peu d’intervention humaine.
* Historisation : permettre de garder les productions précédentes et voir l’évolution des productions.

##### Outils d’intégration continue et de qualimétrie

De nombreux produits répondent aux critères que requières l’intégration continue et de qualimétrie. Parmi ces outils, on peut citer : Hudson/Jenkins, Bamboo (Atlassian), Continuum (Apache), Cruise Control, Luntbuild, Anthill, TeamCity, BuildForge,SonarQube et sûrement d'autre.

L’utilisation de Jenkins ainsi que de SonarQube parmi cette panoplie d’outils est justifié par le fait que l’équipe chargée de la mise en place d’une « usine logicielle » au sein d’AtoS utilisait ce dernier.

### Méthodologie de développement

Dans cette partie il sera question de parler des processus de développement et présenter le processus que nous avons adopté ainsi que son application dans notre projet. La deuxième partie sera consacrée à la présentation du langage de modélisation UML.

#### Processus de développement

Un processus de développement définit un ensemble d’activités et leur enchaînement. Une activité comprend des tâches, des contraintes, des ressources et une démarche. Pour satisfaire le client, ces trois paramètres doivent être pris en compte.

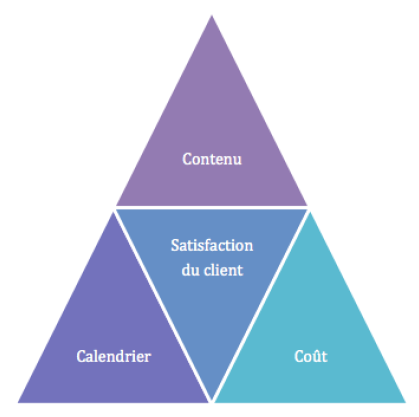
****

Figure 13 : Les contraints d'un projet.

Il existe deux grandes familles de méthodes de gestions de projet que sont les méthodes classiques et les méthodes agiles. Le principal défaut des méthodes classiques est effet tunnel qui peut être très néfaste et conflictuel, on constate souvent un déphasage entre le besoin initial et l’application réalisée. On se rapporte alors aux spécifications validées et au contrat. Les méthodes agiles quant à elles, se présentent comme une alternative qui se propose de réduire considérablement voir complètement cet effet tunnel en impliquant le client dans le projet.

1. La méthode agile

Les méthodes se fondent sur quarte valeurs décrites dans un manifeste qui a découlé de ce rassemblement **[W9]** :

* Les individus et leurs interactions avant les processus et les outils ;
* Des fonctionnalités opérationnelles avant la documentation ;
* Collaboration avec le client plutôt que contractualisation des relations ;
* Acceptation du changement plutôt que conformité aux plans.

Les méthodes agiles utilisent un principe de développement itératif et incrémental qui consiste à découper le projet en sous projets. Les différentes itérations sont définies avec le client et la fin de chaque itération donne un « incrément » qui est un exécutable du produit final.

1. Présentation de SCRUM

*SCRUM* s'appuie sur le découpage d'un projet en boîtes de temps nommés « sprints ». Les sprints peuvent durer entre quelques heures et un mois (avec une préférence pour deux semaines). Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé. Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective. Cette technique analyse le déroulement du sprint achevé, afin d'améliorer ses pratiques. L'adaptation et la réactivité de l'équipe de développement est facilitée par son auto-organisation **[W10]**.

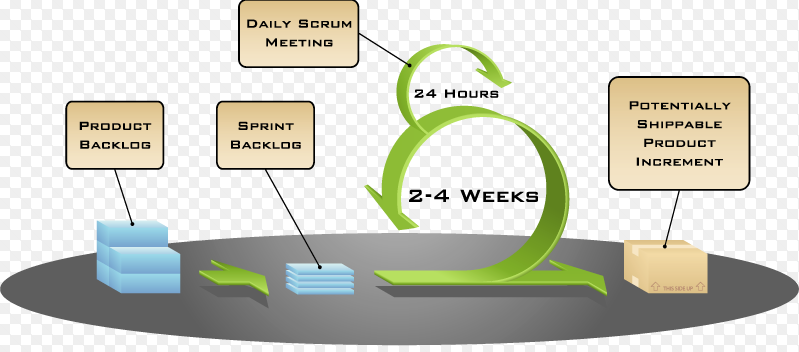


Figure 14 : Cycle de vie de SCRUM

*SCRUM* se fonde sur trois piliers que sont :

* La transparence : *SCRUM* met l’accent sur le fait d’avoir un langage commun entre l’équipe et le management pour permettre une bonne compréhension du projet ;
* L’inspection : À intervalle régulier, *SCRUM* propose de faire le point sur les différents artéfacts produits, afin de détecter toute variation indésirable. Ces inspections ne doivent pas être faites trop fréquemment, ou par un inspecteur mal formé : cela nuirait à l'avancement du projet ;
* L’adaptation : Si une dérive est constatée pendant l'inspection, le processus doit alors être adapté. *SCRUM* fournit des rituels, durant lesquels cette adaptation est possible. Il s'agit de la réunion de planification de sprint, de la mêlée quotidienne, de la revue de sprint ainsi que de la rétrospective de sprint.

#### Les rôles

Dans *SCRUM,* on distingue les différents rôles suivant :

* **Le Product Owner**

Il est responsable de la définition du contenu du produit et de la gestion des priorités pour son développement. Il représente le client final et communique la vision globale du produit à l’équipe.

* **Le Scrum Master**

Il agit en tant que facilitateur entre le Product Owner et les membres de l’équipe. Il doit identifier les obstacles mis en évidence par la méthode et les éliminer.

* **L’Equipe**

Elle est responsable de la réalisation opérationnelle en développant un incrément à chaque sprint. Elle est auto-organisée et doit avoir les compétences nécessaires pour accomplir le travail.

#### SCRUM dans notre projet

Dans le cadre de notre projet, on a adopté la méthode *SCRUM* dans la mesure où l’on devait rendre régulièrement des livrables afin que les maîtres les valident pour pouvoir passer à la tâche suivante.

* **Répartition de rôle**

Les rôles définis dans *SCRUM* ont été attribués aux personnes engagées dans le projet en tenant compte de leur spécialisation. Par conséquent, le Rôle de « *PRODUCT OWNER* » a été attribué à TARIQ DJIBALENE.

Les rôles « *SCRUM MASTER* » et « EQUIPE » ont été attribués comme suit :

* *SCRUM MASTER* : Tariq DJIBALENE
* L’EQUIPE DE DEVELLOPEMENT: Mamadou Oury DIALLO

Des réunions ont été organisées toutes les deux ou trois semaines pour voir l’état d’avancement du projet et pour redéfinir les sprints (durée, tâches, etc.)

#### Présentation d’UML

Dans cette section, nous présentons l’historique d’UML ainsi que ces différents types de diagramme.

1. Définition et historique

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue **[W11]**.

1. Les diagrammes

UML 2 s’articule autour de treize types de diagrammes, chacun d’eux étant dédié à la représentation des concepts particuliers d’un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes :

* Six diagrammes structurels que sont : Diagramme de Diagramme d’objets, Diagramme de paquetages, Diagramme de structure composite, Diagramme de composants, Diagramme de déploiement.
* Sept diagrammes comportementaux :
* Diagramme de cas d’utilisation, Diagramme de vue d’ensemble des interactions, Diagramme de séquence, Diagramme de communication, Diagramme de temps, Diagramme d’activité et les Diagrammes d’états.

1. Planning

Ayant adopté *SCRUM* dans le cadre de ce projet, nous avons utilisé le logiciel GanttProject pour établir notre diagramme deGantt. Le diagramme de Gantt, est un outil qui permet de représenter visuellement l’état d’avancement des différentes activités (tâches) qui constituent un projet.

La *Figure 15* ci-dessous montre la planification des différentes tâches du projet *SMS Banking* a effectuées ainsi qu’une estimation de leur durée de réalisation.

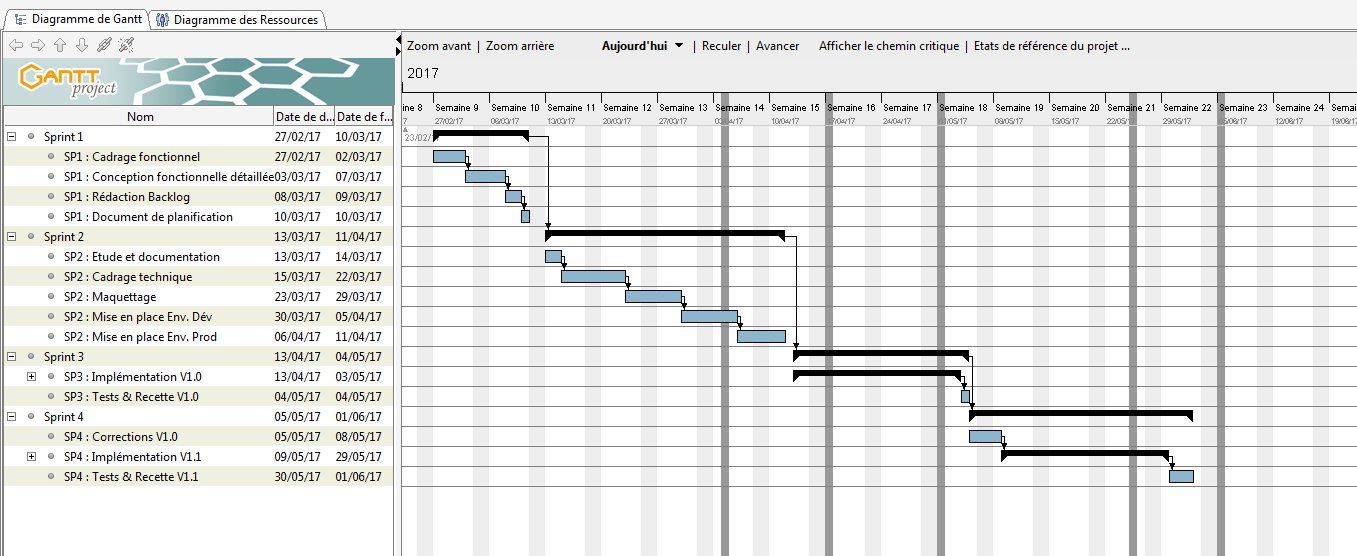


Figure 15: Diagramme de Gantt du SMS Banking

### Conclusion

Dans ce chapitre, nous avons abordé les différentes notions importantes utilisées tout au long du document plus précisément dans l’architecture. On a présenté le ***SMS Banking*** ainsi que les diverses technologies en rapport avec ce dernier pour sa mise en œuvre. Le chapitre suivant sera consacré à la spécification fonctionnelle du système à développer.

# SPECIFICATION FONCTIONNELLE

***Résumé :***

*Dans ce chapitre, nous présentons l’ensemble des fonctionnalités de l’application « SMS Banking » qui, de façon générale, sont scindées en deux parties : Fonctionnalités en Back-End et les Fonctionnalités en Front-End ainsi que les règles métiers liées aux différents cas d’utilisation de ces fonctionnalités.*

**Sommaire**

III.1 Généralités 27

III.1.1 Règles générales 27

III.1.2 Règles liées à l’accès aux écrans 27

III.1.3 Règles liées à la validation des formulaires 27

III.1.4 Règles liées aux actions sur les objets 28

III.2 Fonctionnalités de l’application 29

III.2.1 Fonctionnalité « Envoi et réception de SMS » 29

III.2.2 Fonctionnalité « Gestion des groupes bancaires » 31

III.2.3 Fonctionnalité « Gestion des banques » 32

III.2.4 Fonctionnalité « Gestion des agences » 33

III.2.5 Fonctionnalité « Gestion des clients » 34

III.2.6 Fonctionnalité « Gestion des comptes » 35

III.2.7 Fonctionnalité « Gestion des profils » 36

III.2.8 Fonctionnalité « Gestion des utilisateurs » 38

III.2.9 Fonctionnalité « Gestion des numéros courts » 39

III.2.10 Fonctionnalité « Gestion des services » 40

III.2.11 Fonctionnalité « Gestion des contrats » 45

III.2.12 Fonctionnalité « Chargement des données » 47

III.2.13 Fonctionnalité « Consultation des SMS » 48

III.2.14 Fonctionnalité « Statistiques des SMS » 48

III.2.15 Fonctionnalité « Monitoring des processus » 48

III.3 Conclusion 50

### Généralités

Dans cette partie, nous présentons les fonctionnalités du système, leur cas d’utilisation ainsi que les règles métiers liées à ces derniers.

***Remarque*** :

*Dans toute la suite, les termes « SMS » et « message » seront utilisés indifféremment. L’expression « utilisateur habilité » fait référence à un utilisateur ayant les droits nécessaires afin d’effectuer l’action concernée*.

#### Règles générales

Cette section présente l’ensemble des règles valables pour toutes les fonctionnalités. Cela évite de répéter chacune de ces règles au niveau de chaque cas d’utilisation. Les règles métiers propres à chaque cas (lorsqu’il y en a) seront précisées lors de la description de ce cas.

#### Règles liées à l’accès aux écrans

Un utilisateur ne peut accéder à un écran que s’il a les habilitations dessus : cela est défini par son profil. Lorsqu’il n’a pas accès à un écran, tous les liens menant vers cet écran doivent être masqués (et non affichés et désactivés).

#### Règles liées à la validation des formulaires

1. Champs obligatoires

Lorsque l’utilisateur valide un formulaire sans renseigner un ou plusieurs champs obligatoires :

* Un message d’erreur apparait : « Veuillez renseigner les champs obligatoires »,
* Les champs obligatoires non renseignés sont mis en évidence,
* Les champs saisis ne sont pas réinitialisés (sauf mention contraire).

1. Champs uniques

Lorsque l’utilisateur valide un formulaire alors qu’une règle d‘unicité est violée :

* Un message d’erreur apparait : « Le champ « X » doit être unique ! »,« X » étant le nom du champ en erreur,
* Le champ en erreur est mis en évidence,
* Les champs saisis ne sont pas réinitialisés (sauf mention contraire).

1. Format des champs

Les zones de saisie doivent empêcher la saisie de valeurs au format incorrect (contrôles en Front-end).

1. Violation de règles métiers

Lorsque l’utilisateur valide un formulaire alors qu’une règle métier est violée (contrôles en Back-end) :

* Un message d’erreur apparait indiquant la règle métier violée,
* Les champs saisis ne sont pas réinitialisés (sauf mention contraire).

#### Règles liées aux actions sur les objets

1. Règles sur l’ajout

Lorsque l’opération d’ajout réussit :

* Un message de succès apparait afin d’en informer l’utilisateur
* L’écran d’ajout est réinitialisé

De même, lorsque l’opération d’ajout échoue :

* Un message d’erreur apparait afin d’en informer l’utilisateur et contient une description précise de la (es) cause(s) de cet échec
* L’écran d’ajout n’est pas réinitialisé

1. Règles sur la recherche

Sauf mention contraire, les critères de recherche peuvent ne pas être renseignés et dans ce cas la liste de tous les objets est retournée, De même, lorsque le(s) critère(s) est (sont) renseigné(s), alors la recherche doit retourner uniquement les objets correspondant à ce(s) critère(s).

Lorsqu’aucun résultat n’est retourné :

* Le tableau résultat de la recherche est tout de même affiché
* Le message suivant apparait dans ce tableau : « Aucun résultat trouvé »

1. Règles sur la modification

Lorsque l’utilisateur tente de modifier un objet qui n’existe pas (notamment si la liste affichée des objets n’est pas actualisée), le message d’erreur suivant doit être affiché :

« L’opération ne peut être effectuée, car cet élément n’existe pas ! ».

1. Règles sur la suppression

Toute opération de suppression doit être précédée de l’affichage d’un message de confirmation avant application. Ce message sera libellé ainsi : « Êtes-vous sûr de vouloir effectuer cette suppression ? ». Lorsque l’utilisateur tente de supprimer un objet qui n’existe pas (notamment si la liste affichée des objets n’est pas actualisée), le message d’erreur suivant doit être affiché : « L’opération ne peut être effectuée, car cet élément n’existe pas ! »,

La suppression en cascade n’est pas autorisée, ainsi lorsqu’un utilisateur tente de supprimer un objet lié à un ou plusieurs objets, le message d’erreur suivant doit être affiché : « L’opération ne peut être effectuée, car cet élément est lié à d’autres objets ! ».

### Fonctionnalités de l’application

Cette section présente l’ensemble des fonctionnalités de l’application « *SMS Banking* » sur la base du cahier des charges. Pour chacune des fonctionnalités, il sera listé :

* Les cas d’utilisations liées à cette fonctionnalité
* La description des objets (attributs et leurs contraintes)
* Les règles métiers lorsqu’il y en a

#### Fonctionnalité « Envoi et réception de SMS »

Cette fonctionnalité concerne l’échange par SMS entre une banque et ses clients. Deux cas sont à distinguer :

* Mode PUSH : ce mode correspond au cas où le serveur envoie un SMS à un ou plusieurs clients (pas de réponse du client).
* Mode PULL : ce mode correspond au cas où le client envoie un SMS à la banque dans le but de solliciter un service particulier (consulter le solde d’un compte, vérifier la disponibilité d’un chéquier, etc.). Une réponse lui est alors envoyée.

1. Cas d’utilisation « Envoyer un SMS à un client (mode PUSH) »

* Description

La banque peut envoyer des messages à ses clients, selon une fréquence paramétrée au sein de l’application. Ces messages peuvent être liés à des services (solde, crédit, débit, etc.) ou contenir d’autres types d’informations (promotions, indisponibilité de guichets, etc.). Ce mode d’envoi correspond au Mode « PUSH ».

La périodicité (fréquence d’envoi) peut être paramétrée à un jour et une heure donnés. En plus de cela, elle peut être :

* Périodicité « Mensuelle » : Envoi d’un message à un jour du mois bien défini.
  + Exemple : envoi du solde au client le 05 de chaque mois à 07h du matin.
* Périodicité « Hebdomadaire » : Envoi d’un message un jour de semaine bien défini.
  + Exemple : envoi des 5 dernières opérations sur compte chaque samedi à 10h du matin.
* Périodicité « Journalière » : Envoi d’un message chaque jour.
  + Exemple : envoi de la dernière opération de crédit chaque jour à 18h.
* Périodicité « Instantanée » : Envoi d’un message instantané dès qu’une opération a lieu. Cela suppose bien entendu que cette opération est remontée par la banque immédiatement au niveau du SI de l’application.
  + Exemple : envoi de la dernière opération (crédit ou débit) dès qu’elle a lieu.
* Règle(s) métier

Un client reçoit des notifications liées un de ses comptes si et seulement si :

* Le service correspondant à cette notification est souscrit, est associé à ce compte et est actif.
* La même notification n’a pas déjà été envoyée au client (dans le cas d’un envoi instantané : « Périodicité instantanée »).

1. Cas d’utilisation « Recevoir un SMS du client (mode PULL) »

* Description

Un client peut envoyer à partir de son téléphone portable, à tout moment, un message à la banque lié à un service auquel il a souscrit : consultation de solde, transfert d’argent, demande de chéquier, etc. Une fois le message reçu (par la banque) et le client authentifié, sa demande est traitée et un message de réponse lui est retourné. Ce message de réponse peut contenir l’information demandée ou correspondre à un message d’erreur, selon le succès ou l’échec de l’opération. Le format du message reçu doit respecter un format bien défini.

Ce mode d’envoi correspond au Mode « PULL ».

Le tableau ci-dessous liste les messages d’erreur à renvoyer (il faut prévoir bien entendu leur paramétrage dans l’application) :

Tableau 2 : Messages d'erreur à envoyer

|  |  |
| --- | --- |
| **Cause de l’erreur** | **Message d’erreur** |
| **Syntaxe incorrecte ou service non reconnu** | Désolé, la syntaxe est incorrecte. Nous ne pouvons pas traiter votre message. Envoyer **AIDE** pour plus de détails. |
| **Client non authentifié** | Désolé, nous n’avons pas pu vous authentifier. Merci de vérifier votre message. Envoyer **AIDE** pour plus de détails. |
| **Service non souscrit** | Désolé, vous n’avez pas souscrit à ce service. Merci de vous rapprocher de votre agence. |
| **Service inactif** | Désolé, ce service est momentanément indisponible. Merci de réessayer ultérieurement. |

#### Fonctionnalité « Gestion des groupes bancaires »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les groupes bancaires au sein du système, il pourra ainsi : ajouter un groupe bancaire, rechercher un groupe bancaire, modifier un groupe bancaire, supprimer un groupe bancaire et obtenir les détails sur un groupe bancaire. Un groupe bancaire est défini par les attributs suivants :

Tableau 3 : Caractéristiques d'un groupe bancaire

|  |  |  |  |
| --- | --- | --- | --- |
| **Attributs** | **Type** | **Obligatoire** | **Contraintes** |
| **Code groupe** | Chaîne de caractères | Oui | Alphanumérique sur 5 caractères |
| **Libellé groupe** | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |

1. Cas d’utilisation « Ajouter un groupe bancaire »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un groupe bancaire dans le système. Le code groupe est unique.

1. Cas d’utilisation « Rechercher un groupe bancaire »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un groupe bancaire dans le système. Les critères de recherche sont les suivants : Code groupe, Libellé groupe.

Le résultat de la recherche doit présenter un tableau avec les colonnes suivantes : Code groupe, Libellé groupe.

1. Cas d’utilisation « Modifier un groupe bancaire »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un groupe bancaire dans le système. Lors de la modification, si le code groupe est modifié, alors sa valeur doit être unique.

1. Cas d’utilisation « Supprimer un groupe bancaire »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un groupe bancaire dans le système.

#### Fonctionnalité « Gestion des banques »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les banques au sein du système, il pourra ainsi : ajouter une banque, rechercher une banque, modifier une banque, supprimer une banque, afficher les détails d’une banque. Une banque est définie par les attributs suivants :

Tableau 4 : Caractéristique d'une banque

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attributs | Type | Obligatoire | | Contraintes |
| Code banque | Chaîne de caractères | | Oui | Alphanumérique sur 5 caractères |
| Libellé banque | Chaîne de caractères | | Oui | Alphanumérique sur 50 caractères |
| Code pays | Chaîne de caractères | | Oui | Alphabet sur 2 caractères |
| Adresse | Texte | | Non | Texte sur 255 caractères |

1. Cas d’utilisation « Ajouter une banque »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter une banque dans le système. Les règles métiers à respecter sont :

* Le code banque est unique,
* Une banque peut ne pas être associée à un groupe bancaire et associée au plus à un seul groupe bancaire.

1. Cas d’utilisation « Rechercher une banque »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher une banque dans le système. Les critères de recherche sont les suivants : code banque, libellé banque, groupe bancaire. Le résultat de la recherche doit présenter un tableau avec les colonnes suivantes : code banque, libellé banque et groupe bancaire.

1. Cas d’utilisation « Modifier une banque »

Ce cas d’utilisation permet à un utilisateur habilité de modifier une banque dans le système. Lors de la modification, les contraintes suivantes sont à respecter :

* Si le code banque est modifié, alors sa valeur doit être unique
* Une banque associée à un groupe bancaire peut en être dissociée
* Inversement, une banque associée à aucun groupe bancaire peut en être associée

1. Cas d’utilisation « Supprimer une banque »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer une banque dans le système.

#### Fonctionnalité « Gestion des agences »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les agences au sein du système, il pourra ainsi : ajouter une agence, rechercher une agence, modifier une agence, supprimer une agence.

Tableau 5 : Caractéristiques d'une agence

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Code agence | Chaîne de caractères | Oui | Alphanumérique sur 5 caractères |
| Libellé agence | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |
| Adresse | Texte | Non | Texte sur 255 caractères |

1. Cas d’utilisation « Ajouter une agence »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter une agence dans le système. L’ajout d’une agence doit respecter les contraintes suivantes :

* Le code agence est unique
* Une agence doit obligatoirement être associée à une et une seule banque

1. Cas d’utilisation « Rechercher une agence »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher une agence dans le système. Les critères de recherche sont les suivants : code agence, libellé agence, code banque.

Le résultat de la recherche doit présenter un tableau avec les colonnes suivantes : code agence, libellé agence, code banque.

1. Cas d’utilisation « Modifier une agence »

Ce cas d’utilisation permet à un utilisateur habilité de modifier une agence dans le système. Lors de la modification :

* Si le code agence est modifié, alors sa valeur doit être unique
* L’agence peut être associée à nouveau à une autre banque

1. Cas d’utilisation « Supprimer une agence »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer une agence dans le système.

#### Fonctionnalité « Gestion des clients »

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les clients au sein du système, il pourra ainsi : ajouter un client, rechercher un client, modifier un client, supprimer un client.

Tableau 6 : Caractéristique d’un client

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Nom | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |
| Prénom(s) | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |
| Numéro(s) de téléphone | Numérique | Oui | Numérique sur 12 à 13 digits |
| Code secret | Numérique | Oui | Numérique sur 4 à 6 digits |

1. Cas d’utilisation « Ajouter un client »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un client dans le système. Les contraintes suivantes sont à respecter lors de l’ajout d’un client dans le système:

* Au moins un numéro de téléphone doit être fourni pour valider l’ajout d’un client
* Un client est associé à aucun ou plusieurs comptes (cela va permettre de définir ses comptes dans un second temps).

1. Cas d’utilisation « Rechercher un client »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un client dans le système. Les critères de recherche sont les suivants : nom, prénom(s), numéro de téléphone et le code secret. Le résultat de la recherche doit présenter un tableau avec les colonnes suivantes : nom, prénom(s), numéro de téléphone et son code secret.

1. Cas d’utilisation « Modifier un client »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un client dans le système.

Si le numéro de téléphone est modifié, alors sa valeur doit être unique.

1. Cas d’utilisation « Supprimer un client »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un client dans le système.

#### Fonctionnalité « Gestion des comptes »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les comptes au sein du système, il pourra ainsi : ajouter un compte, rechercher un compte, modifier un compte, supprimer un compte et afficher les détails du compte. Un compte est défini par les attributs suivants :

Tableau 7 : Caractéristiques d'un compte bancaire

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Numéro de compte | Numérique | Oui | 20 chiffres ? |
| Solde | Numérique | Oui | Montant |

1. Cas d’utilisation « Ajouter un compte »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un compte dans le système. Lors de l’ajout de ce compte :

* Le Numéro de compte est unique
* Un compte est associé à un et un seul client
* Un compte est associé à une et une seule agence
* L’utilisateur doit obligatoirement associer un client et une agence au compte à ajouter

1. Cas d’utilisation « Rechercher un compte »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un compte dans le système. Les critères de recherche sont les suivants : numéro de compte, nom client, prénom(s) client, code banque, code agence et le numéro de téléphone. Le résultat de la recherche doit présenter un tableau avec les colonnes suivantes : numéro de compte, nom client, prénom(s) client, code banque, code agence et le numéro de téléphone.

1. Cas d’utilisation « Modifier un compte »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un compte dans le système. Lorsqu’on modifie un compte :

* Si le numéro de compte est modifié, alors sa valeur doit être unique
* Le compte peut être associé à nouveau à un autre client ou à une autre agence

1. Cas d’utilisation « Supprimer un compte »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un compte dans le système.

#### Fonctionnalité « Gestion des profils »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les profils au sein du système, il pourra ainsi : ajouter un profil, rechercher un profil, modifier un profil et supprimer un profil. Un profil est défini par les attributs suivants :

Tableau 8 : Caractéristiques des profils

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Libellé profil | Chaine de caractères | Oui | Alphanumérique sur 50 caractères |

En plus de(s) attribut(s) ci-dessus, lors de la définition d’un profil, il faudra également lister tous les écrans (ou fonctionnalités) auxquels un utilisateur avec ce profil peut accéder.

Exemple : le tableau ci-dessus indique certains profils (à titre d’exemple) avec leurs autorisations / restrictions :

Tableau 9 : Exemple de paramétrage de profil

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Libellé profil | Gestion banque | Gestion compte | Gestion client | Gestion contrat | Consultation SMS | Stat. Coût des SMS |
| Admin. Système |  |  |  |  |  |  |
| Agent souscription |  |  |  |  |  |  |
| Financier |  |  |  |  |  |  |

1. Cas d’utilisation « Ajouter un profil »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un profil dans le système. Lors de la création d’un profil, en plus des attributs, l’utilisateur doit sélectionner l’ensemble des écrans (ou fonctionnalités) accessibles.

* Prévoir un profil « Profil Super Admin » qui a accès à tout le système sans exception :
* Ce profil est paramétré de base et est disponible à l’installation du système
* Ce profil n’est pas modifiable.

Lors de la création d’un profil qui a exactement les mêmes accès écrans (ou fonctionnalités) qu’un profil existant, afficher un message d’avertissement à l’utilisateur. La création d’un tel profil reste tout de même possible.

1. Cas d’utilisation « Rechercher un profil »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un profil dans le système.

Les critères de recherche sont les suivants : libellé profil, liste des écrans.

1. Cas d’utilisation « Modifier un profil »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un profil dans le système. Lors de la modification d’un profil qui a exactement les mêmes accès écrans (ou fonctionnalités) qu’un profil existant, afficher un message d’avertissement à l’utilisateur. La modification d’un tel profil reste tout de même possible.

1. Cas d’utilisation « Supprimer un profil »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un profil dans le système.

Tous les profils peuvent être supprimés à l’exception du profil de base « Profil Super Admin ».

#### Fonctionnalité « Gestion des utilisateurs »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les utilisateurs au sein du système, il pourra ainsi : ajouter un utilisateur, rechercher un utilisateur, modifier un utilisateur et supprimer un utilisateur. Un utilisateur est défini par les attributs suivants. Le *Tableau 10* suivant donne les caractéristiques d’un utilisateur :

Tableau 10 : Caractéristiques d'un utilisateur

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Nom | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |
| Prénom(s) | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |
| Image | Chaîne de caractères | Non | Correspond au chemin relatif de l’image de l’utilisateur |

1. Cas d’utilisation « Ajouter un utilisateur »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un utilisateur dans le système et de lui associer un profil. Les contraintes à respecter lors de l’ajout d’un utilisateur sont :

* Un utilisateur est associé à un et un seul profil.
* Un profil est obligatoire lors de la création d’un utilisateur.
* Le système doit prévoir un utilisateur pré configuré « Super Admin » avec le profil « Profil Super Admin ».
* Un seul utilisateur tout au plus peut posséder le profil « Profil Super Admin ».

1. Cas d’utilisation « Rechercher un utilisateur »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un utilisateur dans le système. Les critères de recherche sont les suivants : prénom(s), nom, profil.

1. Cas d’utilisation « Modifier un utilisateur »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un utilisateur dans le système. Les règles métiers pour ce cas sont :

* L’affectation d’un profil est obligatoire lors de la création d’un utilisateur.
* L’utilisateur « Super Admin » ne peut pas être modifié.
* Un seul utilisateur tout au plus peut posséder le profil « Profil Super Admin ».

1. Cas d’utilisation « Supprimer un utilisateur »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un utilisateur dans le système. Les contraintes de suppression sont :

* Tous les utilisateurs peuvent être supprimés à l’exception de l’utilisateur de base « Super Admin ».
* Un utilisateur connecté ne peut être supprimé. Cela implique alors de :
* Fixer une durée pour sa session : une fois la session expirée, l’utilisateur sera déconnecté et la suppression devient alors possible.
* Mettre en attente cette opération de suppression jusqu’à déconnexion de l’utilisateur.
* Empêcher un utilisateur ayant une opération de suppression en attente de se connecter et la suppression devient alors possible.
* Permettre de lever (annuler) l’opération de suppression en attente et la connexion de l’utilisateur devient alors possible.

#### Fonctionnalité « Gestion des numéros courts »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les numéros courts au sein du système, il pourra ainsi : ajouter un numéro court, rechercher un numéro court, modifier un numéro court, supprimer un numéro court, activer un numéro court et désactiver numéro court. Un numéro court est le numéro vers lequel un client de la banque enverra un SMS afin de solliciter un service donné. Il est défini par les attributs du *Tableau 11* suivants :

Tableau 11 : Caractéristiques d'un numéro court

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Numéro | Numérique | Oui | 10 chiffres max |
| Code pays | Chaîne de caractères | Oui | Alphabet sur 2 caractères |
| Coût SMS | Numérique | Oui | Montant |
| Code Devise | Chaîne de caractères | Oui |  |
| Description | Texte | Non |  |

1. Cas d’utilisation « Ajouter un numéro court »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un numéro court dans le système. Le numéro court est unique et doit être actif par défaut.

1. Cas d’utilisation « Rechercher un numéro court »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un numéro court dans le système. Les critères de recherche sont les suivants : numéro court.

1. Cas d’utilisation « Modifier un numéro court »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un numéro court dans le système.

1. Cas d’utilisation « Supprimer un numéro court »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un numéro court dans le système.

1. Cas d’utilisation « Activer un numéro court »

Ce cas d’utilisation permet à un utilisateur habilité d’activer un numéro court dans le système.

Le numéro court à activer doit être à l’état désactivé.

1. Cas d’utilisation « Désactiver un numéro court »

Ce cas d’utilisation permet à un utilisateur habilité de désactiver un numéro court dans le système. Le numéro court à désactiver doit être à l’état activé et les messages envoyés sur un numéro court désactiver, ne doivent pas être traités : l’utilisateur recevra une notification dans ce sens (cf. « (Cas d’utilisation « Recevoir un SMS du client (mode PULL) »).

#### Fonctionnalité « Gestion des services »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les services au sein du système, il pourra ainsi : ajouter un service, rechercher un service, modifier un service, supprimer un service, activer un service, désactiver service. Il est défini par les attributs du *Tableau 12* suivants :

Tableau 12 : Caractéristique des numéros courts

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Libellé | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |
| Description | Texte | Non |  |
| Mot clé | Chaîne de caractères | Oui | Alphabet sur 10 caractères  Séparateur « \_ » autorisé |

L’attribut « Mot clé » correspond au mot que le client doit indiquer dans son message afin de référencer le service sollicité.

1. Syntaxe d’un message lié à un service

* Syntaxe d’un message envoyé par le client :

Chaque message envoyé par un client doit respecter une syntaxe bien définie afin d’être pris en compte. Sinon, un message d’erreur lui est retourné.

Cette syntaxe dépend du service sollicité, mais de façon générale elle apparait sous cette forme (chevrons non inclus) :

**<MOT\_CLE> <ESPACE> <CODE\_ACCES> <ESPACE> <ARGUMENTS>**

Le *Tableau 13* ci-après en fournit une description détaillée :

Tableau 13 : Syntaxe des messages envoyés par le client

|  |  |
| --- | --- |
| Elément | Description |
| MOT\_CLE | Il correspond à l’attribut « Mot clé » d’un service (voir plus haut) et permet d’indiquer donc le service sollicité.  Exemples :  **SOLDE** : correspondrait au service « Solde » permettant de consulter le solde d’un compte  **CREDIT** : correspondrait au service « Crédit » permettant de consulter la dernière opération de crédit |
| ESPACE | En principe un espace de séparation entre les éléments **« MOT\_CLE »** et **« ARGUMENTS »**.  Plusieurs espaces peuvent être tolérés. |
| CODE\_ACCES | Cette partie représente le code d’accès du client. Elle est isolée de la section suivante **« ARGUMENTS »** afin de simplifier le repérage et l’extraction du code d‘accès qui est utile à la phase d’authentification |
| ARGUMENTS | Cette partie représente les informations utiles au traitement du service sollicité et peuvent varier d’un service à un autre. Dans tous les cas, si elles existent, ces informations sont séparées par **« ESPACE »**.  **Exemples :**   * Pour le service « Solde » de consultation des soldes de comptes, le message pourrait être ainsi : **SOLDE** **1234** * Pour le service « Solde » de consultation d’un solde d’un compte particulier, le message pourrait être ainsi : **SOLDE** **1234** **001122334455** * Pour le service « Chéquier » de vérification de la disponibilité d’un chéquier, le message pourrait être ainsi : **CHEQUIER 4321** * Pour le service « Virement » de virement d’un compte vers un autre, le message pourrait être ainsi : **VIREMENT** **4567** **001122334455 009988776655**   Ces différents exemples montrent que la validation de la syntaxe du message dépend du service sollicité (liste variable d’arguments). |

* Syntaxe d’un message envoyé par la banque :

Que ce soit en mode « Pull » ou en mode « Push » un message envoyé par la banque est :

* Lié obligatoirement à un service.
* Paramétré dans le système (lors de la création du service lié). Le message peut alors être de deux types :
  + Statique : un simple texte envoyé tel qu’il a été saisi par l’utilisateur
  + Dynamique : un texte contenant des sections appelées « Paramètres » avant envoi par des informations liées par exemple à la banque, à une agence, à un client, à un compte, etc.

Exemple :

Le message de réponse au service « Solde » (voir « Services de base ») peut être paramétré sous cette forme :

Le solde de votre compte {NUMERO\_COMPTE} est de {SOLDE\_COMPTE} FCFA

Et à l’envoi, les paramètres « {NUMERO\_COMPTE} » et « {SOLDE\_COMPTE} » remplacés par les informations relatives à un des comptes du client ayant envoyé un message :

Le solde de votre compte 00112233445566 est de 10.475.250 FCFA

1. Services de base

Le système doit offrir un certain nombre de services de base pré configurés. Bien entendu, ils obéissent aux mêmes règles relatives à la syntaxe du message à envoyer (voir plus haut).

Ces services de base sont décrits dans le *Tableau 14* suivant :

Tableau 14 : Liste des services de base

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Service | Mot clé | Push | Pull | Description |
| Aide | AIDE |  |  | Service d’aide permettant de recevoir les mots clés disponibles.  Par défaut le message de réponse est :  Envoyer AIDE suivi du mot clé lié au service. Les mots clés disponibles sont : SOLDE, CREDIT, DEBIT, CHEQUIER, CB, SEUIL, MVT, VIR |
| Solde | SOLDE |  |  | Service permettant de connaître le solde d’un ou plusieurs comptes :   * Si l’utilisateur envoie SOLDE 1111, alors le solde de chaque compte est retourné * Si l’utilisateur envoie SOLDE 1111 001122334455, alors solde du compte N° 001122334455 est retourné |
| Crédit | CREDIT |  |  | Service permettant de connaître les N dernières opérations de crédit :   * Si l’utilisateur envoie CREDIT 1111, alors la dernière opération de crédit est retournée * Si l’utilisateur envoie CREDIT 1111 3, alors les 3 dernières opérations de crédit sont retournées |
| Débit | DEBIT |  |  | Service permettant de connaître les N dernières opérations de débit :   * Si l’utilisateur envoie DEBIT 1111, alors la dernière opération de débit est retournée * Si l’utilisateur envoie DEBIT 1111 3, alors les 3 dernières opérations de débit sont retournées |
| Chéquier | CHEQUIER |  |  | Service permettant de vérifier la disponibilité du chéquier |
| Carte bancaire | CB |  |  | Service permettant de vérifier la disponibilité de la carte bancaire |
| Seuil | SEUIL |  |  | Service permettant de recevoir une alerte en cas de dépassement de seuil sur l’un de ses comptes  Lors du paramétrage d’un tel service, il faut donc prévoir un moyen de renseigner le montant du seuil et en min ou en max |
| Mouvement | MVT |  |  | Service permettant de connaître les N dernières opérations (crédit ou débit) :   * Si l’utilisateur envoie MVT 1111, alors la dernière opération est retournée * Si l’utilisateur envoie MVT 1111 3, alors les 3 dernières opérations sont retournées |
| Virement | VIR |  |  | Service permettant d’effectuer un virement d’un compte vers un autre |

1. Cas d’utilisation « Ajouter un service »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un service dans le système. Lors de l’ajout d’un service dans le système :

* Le mot clé est unique.
* Un service peut être défini pour le mode « Pull » et/ou le mode « Push » :
  + Si le service n’est pas défini en mode « Pull », lorsque le client envoie un message lié à ce service, il reçoit le message d’erreur suivant (qui peut être modifié par la suite) : « Désolé, ce service n’est pas disponible en mode Pull. Nous ne pouvons pas traiter votre demande. Merci ».
  + Si le service n’est pas défini en mode « Push », il ne doit pas être pris en compte lors de l’envoi des messages Push par la banque.
* Tout service doit au moins être actif pour l’un des deux modes « Pull » et « Push ».
* Le service est par défaut disponible en mode « Pull » et « Push » (sauf pour certains services de base).
* Le service est par défaut actif.

1. Cas d’utilisation « Rechercher un service »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un service dans le système.

Les critères de recherche sont les suivants : libellé, mot clé, mode Pull, mode Push

1. Cas d’utilisation « Modifier un service »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un service dans le système. Lors de la modification, Si le mot clé est modifié, alors sa valeur doit être unique.

Cette modification est logiquement prise en compte systématiquement dans le contenu du message de réponse du service « Aide » (qui liste tous les mots clés).

1. Cas d’utilisation « Supprimer un service »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un service dans le système.

1. Cas d’utilisation « Activer un service »

Ce cas d’utilisation permet à un utilisateur habilité d’activer un service dans le système.

Le service à activer doit être à l’état désactivé

1. Cas d’utilisation « Désactiver un service »

Ce cas d’utilisation permet à un utilisateur habilité de désactiver un service dans le système. Le service à désactiver doit être à l’état activé. Les messages envoyés sur un service désactivé, ne doivent pas être traités : l’utilisateur recevra une notification dans ce sens (cf. Cas d’utilisation « Recevoir un SMS du client (mode PULL) »).

#### Fonctionnalité « Gestion des contrats »

* Description

Cette fonctionnalité permet à un utilisateur habilité de paramétrer les contrats au sein du système, il pourra ainsi: ajouter un contrat, rechercher un contrat, modifier un contrat, supprimer un contrat, activer un contrat, désactiver contrat.

Un contrat est l’accord entre la banque et le client stipulant que ce dernier a souscrit à un ensemble de services. Il est défini par les attributs du *Tableau 15* suivants :

Tableau 15 : Attributs de la table gestion de contrat

|  |  |  |  |
| --- | --- | --- | --- |
| Attributs | Type | Obligatoire | Contraintes |
| Numéro contrat | Numérique | Oui | 10 chiffres max |
| Libellé contrat | Chaîne de caractères | Oui | Alphanumérique sur 50 caractères |
| Description | Texte | Non |  |
| Date début | Date et heure | Oui | **Format :** YYYY-MM-JJ HH:MM:SS |
| Date fin | Date et heure | Oui | **Format :** YYYY-MM-JJ HH:MM:SS |

Un contrat peut être renseigné de deux façons :

* Un utilisateur habilité le paramètre dans le système (cf. « Cas d’utilisation » ci-dessous)
* La banque fournit toutes les informations nécessaires au système et celui-ci l’intègre via la procédure de chargement des données (cf. « Cas d’utilisation « Charger les données de souscription aux services »)

1. Cas d’utilisation « Ajouter un contrat »

Ce cas d’utilisation permet à un utilisateur habilité d’ajouter un contrat dans le système.

Le numéro du contrat est unique et un contrat est associé à au moins un compte. Pour chaque compte, il indique les services souscrits et dans quel(s) mode(s) ils le sont :

* Un service disponible en mode « Pull » et « Push » peut être paramétré pour n’être disponible qu’en un seul mode sur un compte
* Un service disponible en un seul mode est forcément paramétré uniquement pour ce mode sur un compte

Pour chaque compte, au moins un service doit être souscrit et un numéro de téléphone du client est défini (numéro parmi ceux préenregistrés).Il faut noter que le même numéro peut être utilisé pour tous les comptes, est actif par défaut et un contrat a une date de début et une date de fin :

* Si la date de début n’est pas encore atteinte (ce qui ne devrait en principe pas arriver) ou si la date de fin est atteinte (expiration de contrat), aucun message ne doit être traité relativement à ce contrat quel que soit le mode. Le client reçoit alors un message d’erreur dans ce sens.

1. Cas d’utilisation « Rechercher un contrat »

Ce cas d’utilisation permet à un utilisateur habilité de rechercher un contrat dans le système. Les critères de recherche sont les suivants : le numéro contrat, le libellé du contrat, la date début, Date fin du contrat.

1. Cas d’utilisation « Modifier un contrat »

Ce cas d’utilisation permet à un utilisateur habilité de modifier un contrat dans le système. Si le numéro contrat est modifié, alors sa valeur doit être unique .La souscription aux services peut être modifiée.

1. Cas d’utilisation « Supprimer un contrat »

Ce cas d’utilisation permet à un utilisateur habilité de supprimer un contrat dans le système.

1. Cas d’utilisation « Activer un contrat »

Ce cas d’utilisation permet à un utilisateur habilité d’activer un contrat dans le système.

Le contrat à activer doit être à l’état désactivé.

1. Cas d’utilisation « Désactiver un contrat »

Ce cas d’utilisation permet à un utilisateur habilité de désactiver un contrat dans le système. Le contrat à désactiver doit être à l’état activé. Les messages envoyés relativement à un contrat désactivé, ne doivent pas être traités : l’utilisateur recevra une notification dans ce sens (cf. Cas d’utilisation « Recevoir un SMS du client (mode PULL) »).

#### Fonctionnalité « Chargement des données »

Cette fonctionnalité permet d’alimenter la base de données du système avec les données fournies par la banque. Le chargement de données est découpé en 3 parties :

* Charger les données clients et comptes.
* Charger les données de souscription aux services (contrats).
* Charger les opérations sur les comptes.

Les différents cas d’utilisation de cette fonctionnalité sont :

* Cas d’utilisation « Charger des données clients et comptes »
* Cas d’utilisation « Accéder aux logs de chargement »
* Cas d’utilisation « Charger les données de souscription aux services »
* Cas d’utilisation « Charger les opérations sur les comptes »
* Cas d’utilisation « Purger les opérations »
* Cas d’utilisation « Purger les logs de chargement »

#### Fonctionnalité « Consultation des SMS »

Cette fonctionnalité permet à un utilisateur habilité de consulter les messages entrants et/ou sortants, messages sans réponse et éventuellement sur une période donnée. Cette fonctionnalité est citée juste à titre indicatif et n’est pas encore décrite/détaillée ni implémentée.

#### Fonctionnalité « Statistiques des SMS »

Cette fonctionnalité permet à un utilisateur habilité de consulter les statistiques SMS : graphiques et coûts. Cette fonctionnalité est citée juste à titre indicatif et n’est pas encore décrite/détaillée ni implémentée.

#### Fonctionnalité « Monitoring des processus »

* Description

Cette fonctionnalité permet à un utilisateur habilité d’effectuer la surveillance et la mesure de performance du système.

En effet, plusieurs processus liés au système (dont le chargement de données, l’envoi de message, la réception de messages, etc.) seront lancés et il faudra alors surveiller leur état d’exécution et leur temps de réponse.

La disponibilité de l’application (interface d’administration) et le serveur de messagerie sont également concernés. Cela permet d’alerter l’administrateur du système (ou toute autre personne habilitée) en cas de disfonctionnement (par sms ou par mail) et également de pouvoir construire des graphes afin de visualiser les performances du système.

1. Cas d’utilisation « Monitorer les processus en tâche de fond »

Ce cas va permettre à l’utilisateur habilité de surveiller les processus qui tournent en tâche de fond. Le système doit permettre de lister l’ensemble des processus critiques afin de les monitorer. Les informations utiles sont :

* Le statut d’exécution
* La date du dernier lancement
* Le nombre de redémarrage subi
* Le temps écoulé depuis la dernière exécution
* Le résultat des derniers traitements
* Le temps de traitements (s’il y en a)

1. Cas d’utilisation « Afficher l'état des processus en tâche de fond »

Ce cas va permettre à l’utilisateur habilité d’afficher l’état des processus qui tournent en tâche de fond. Le système doit permettre de lister l’ensemble des processus critiques afin de les monitorer. Les informations utiles sont :

* Le statut d’exécution
* La date du dernier lancement
* Le nombre de redémarrage subi
* Le temps écoulé depuis la dernière exécution
* Le résultat des derniers traitements
* Le temps de traitements (s’il y en a)

1. Cas d’utilisation « Arrêter un processus »

Ce cas va permettre à l’utilisateur habilité d’arrêter un processus qui tourne en tâche de fond. Le processus à arrêter doit être en cours d’exécution. L’arrêt de certains processus peut être interdit s’ils ont des traitements en cours. Toutefois, s’il s’agit d’un processus qui tourne indéfiniment et qui bloque des traitements, alors dans ce cas il peut être arrêté.

* L’action d’arrêt doit être tracée :
* L’auteur (l’utilisateur qui a effectué l’action)
* La date et heure
* Le commentaire de l’auteur
* Le nombre de tentatives jusqu’au succès

1. Cas d’utilisation « Démarrer un processus »

Ce cas va permettre à l’utilisateur habilité de démarrer un processus en tâche de fond.

* Le processus à démarrer doit être à l’arrêt
* L’action de démarrage doit être tracée :
* L’auteur (l’utilisateur qui a effectué l’action)
* La date et heure
* Le commentaire de l’auteur
* Le nombre de tentatives jusqu’au succès

### Conclusion

Dans ce chapitre, nous avons décrit le processus de développement adopté dans le cadre de ce projet ainsi que la manière dont on l’a appliqué à notre projet. Ensuite on a vu les différentes fonctionnalités du système *SMS Banking*, leur cas d’utilisation et les contraintes à respecter. Dans le chapitre suivant, nous présentons la conception du système *SMS Banking* en détaille tout en proposant différentes types d’architectures.

# CONCEPTION DU SYSTEME

**Résumé :**

*Dans ce chapitre, nous entamons l’étude conceptuelle. Cette phase vise à trouver une solution informatique pour mettre en œuvre et construire le système analysé. Dans cette phase les fonctionnalités sont illustrées à l’aide de diagramme UML. A cet effet, nous utilisons les cas d'utilisations qui permettent d’illustrer les besoins fonctionnels. La description textuelle et la description séquentielle (diagramme de séquence) de certains cas jugés importants ont été faites. Cela va nous permettre de délimiter le système pour pouvoir prendre en compte toutes les contraintes et mesures nécessaires à la réussite et à la production d’une application de qualité, qui répond aux besoins du client*.

**Sommaire**

IV.1 Spécification fonctionnelle détaillée 52

IV.1.1 Catalogue 52

IV.1.2 Définition des acteurs 52

IV.1.3 Etude préliminaire 53

IV.1.4 Diagramme de cas d’utilisation et leur descriptions 55

IV.1.5 Diagramme de classe 61

IV.2 Conception détaillée 63

IV.3 Architectures générale du système 67

IV.3.1 Communication via un modem GSM 69

IV.3.2 Communication via une liaison spécialisée 72

IV.4 Architecture technico-fonctionnelles 73

IV.4.1 Architecture fonctionnelle 73

IV.4.2 Architecture logicielle 76

IV.5 Conclusion 77

### Spécification fonctionnelle détaillée

#### Catalogue

Les acteurs suivants sont les utilisateurs du système *SMS Banking*. Il faut préciser que cette liste n’est pas exhaustive car, elle dépendra de l’organigramme de la banque ou structure bancaire souhaitant implémenter l’outil. Le *Tableau 16* suivant en donne un aperçu.

Tableau 16 : Acteurs du système

|  |  |  |
| --- | --- | --- |
| **Id** | **Nom** | **Use Case (Acteur principal)** |
| A\_01 | Admin. Système | Tous |
| A\_03 | Utilisateur habilité | Use case assignés |
| A\_02 | Client de la banque | SMS PULL |

#### Définition des acteurs

Le *Tableau 17* suivant décrit les acteurs du système et leurs rôles.

Tableau 17 : Les acteurs et leurs rôles

|  |  |  |
| --- | --- | --- |
| Acteur | Description | Rôle |
| A\_01 | L’admin système est un utilisateur qui a tous les droits sur le système. | Paramétré tout le système |
| A\_02 | Un utilisateur habilité est un utilisateur ayant les droits nécessaires pour effectuer une tâche particulière. | Son rôle principal est d’accomplir les tâches qui lui sont assignées. |
| A\_03 | C’est l’acteur externe dans le système ; le bénéficiaire du service ‘*SMS Banking*’. Il reçoit par SMS des alertes de type Push envoyées automatiquement par le système ou bien il émet par SMS des requêtes de type Pull et reçoit les réponses après le traitement de sa demande par le système. | Son rôle principal est d’interagir avec le système. |

#### Etude préliminaire

Les diagrammes de cas d’utilisation permettent d’exprimer les besoins des utilisateurs du système, ils sont orientés utilisateur de ce besoin. Nous allons scinder notre système en diagramme de packages et chaque diagramme de package est vu comme un « sous-système » de notre plateforme.

Les différents diagrammes de packages représentent les fonctionnalités définies dans l’étude conceptuelle. Dans ce chapitre, vu que nous utilisons la méthodologie *SCRUM*, nous allons réaliser les diagrammes de packages des fonctionnalités suivantes (pour la version 1.0 du système) : Gestion des groupes bancaires, Gestion des banques, Gestion des agences, Gestion des clients, Gestion des comptes, Gestion des Contrats, Gestion des utilisateurs, Gestion des profils, Gestion des services, Gestion des numéros courts, Gestion des SMS. Le *Figure 13* ci-dessous montre le diagramme de package du système.

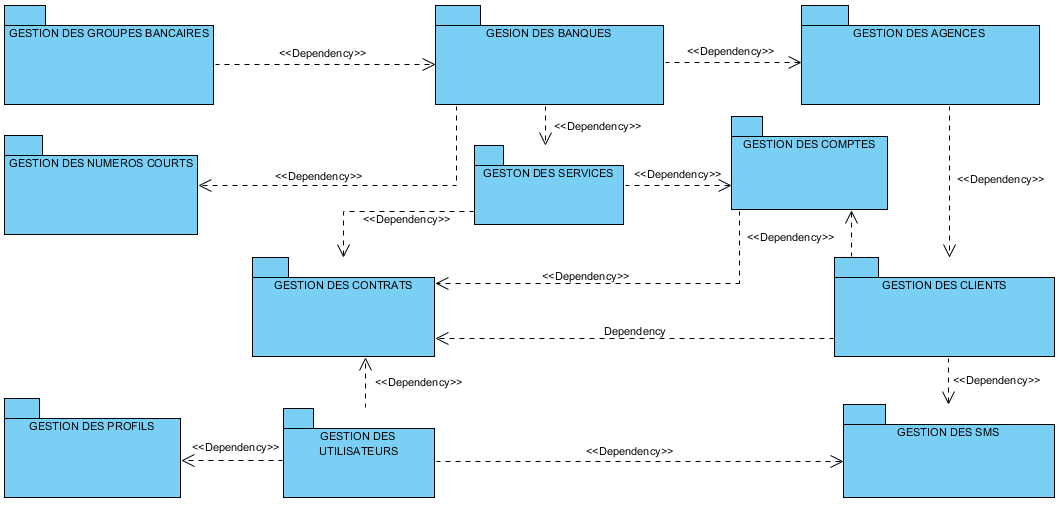


Figure 16 : Diagramme de packages du système SMS Banking

Vu que nous avons identifié les différents acteurs du système et définit les différents packages, dans les sections suivantes nous allons établir les diagrammes de cas d’utilisation de chaque sous-système (package), la description et le diagramme de séquence de certains cas.

#### Diagramme de cas d’utilisation et leur descriptions

La représentation du diagramme de cas d’utilisation de chaque package sera suivie d’une description textuelle des différents cas d’utilisation de cette dernière.

###### Le sous-système « Gestion des utilisateurs»

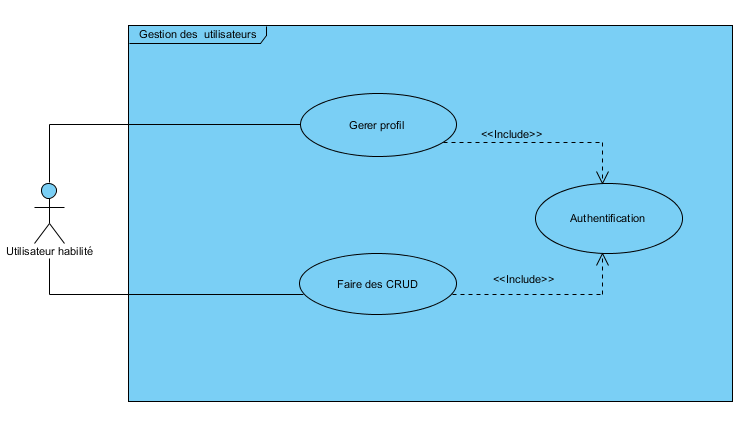


Figure 17 : Sous-système « Gestion des utilisateurs »

1. Cas d’utilisation « Authentification »



Figure 18 : Cas d'utilisation "Authentification"

* **Description textuelle**

Tableau 18 : Description textuelle "Authentification"

|  |  |
| --- | --- |
| **Titre** | **Authentification** |
| **Description** | Ce cas d’utilisation permet à un utilisateur de donner son mot de passe et son nom d’utilisateur afin d’avoir accès à son profil et aux différentes fonctionnalités. |
| **Acteur(s)** | Utilisateurs habilités |
| **Pré condition** | 1 L’utilisateur n’est pas connecté.  2 L’utilisateur accède à l’interface d’authentification. |
| **Scénario nominal** | 1 Le système demande le mot de passe et le login de L’utilisateur.  2 L’utilisateur rempli le formulaire d’authentification et valide.  3 Le système vérifie la validité du login et du mot de passe.  4 Le système redirige l’utilisateur vers son tableau de bord. |
| **Scénario alternatif** | **1.a.**  Le login ou le mot de passe est incorrect. Le système affiche le message suivant : «**Login ou Mot de passe incorrect** »,  le système reprend le scénario 1. |
| **Post condition** | L’utilisateur est connecté et accède à son profil. |

* **Diagramme de séquence**

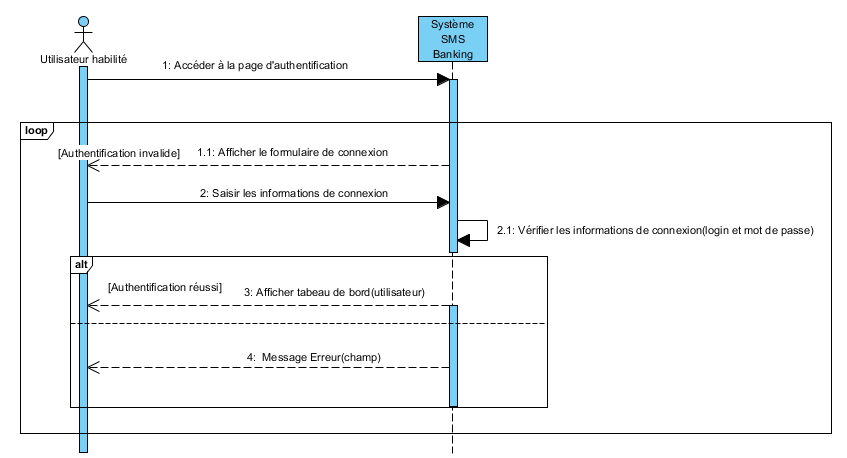


Figure 19 : Diagramme de séquence du cas « authentification »

###### Sous-système « Gestion des groupes bancaires »

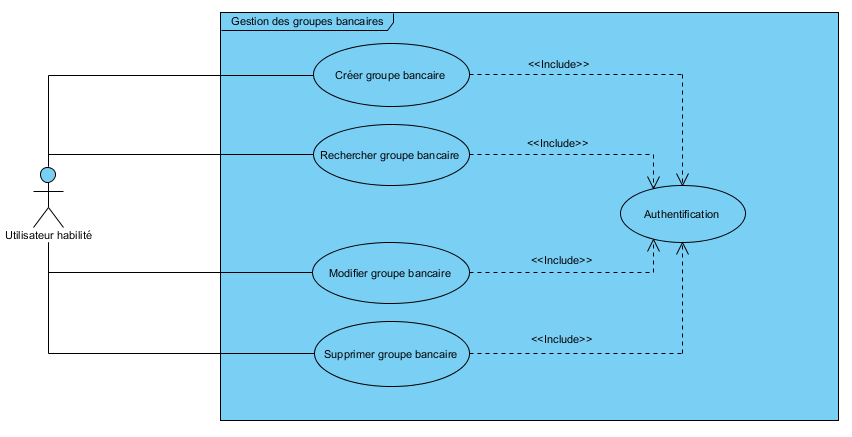


Figure 20 : Sous-système "Gestion des groupe bancaire"

1. Cas d’utilisation « Créer groupe bancaire »

* **Description textuelle**

Tableau 19 : Description textuelle "créer un groupe bancaire"

|  |  |
| --- | --- |
| **Titre** | **Créer groupe bancaire** |
| **Description** | Ce cas permet à un utilisateur habilité de créer un groupe bancaire. |
| **Acteur(s)** | Utilisateur(s) habilité(s) |
| **Pré condition** | 1. Etre connecté.  2. Accéder à l’interface de « **Gestion de groupes bancaires** » |
| **Scénario nominal** | 1. L’utilisateur clique sur le bouton « **Créer groupe bancaire ».**  3. Le système affiche un formulaire de saisie.  4. L’utilisateur remplit les différents champs du formulaire.  5. L’utilisateur clique sur le bouton « **Valider »**.  6. Le système crée le groupe bancaire.  7. Le système notifie l’utilisateur avec le message « **groupe bancaire créée avec succès »**. |
| **Scénario alternatif** | **4a**- Un ou plusieurs champ(s) obligatoire(s) ne sont pas rempli.   1. Le cas d’utilisation reprend à l’étape 1 du scénario nominale sans réinitialisation des champs saisis. 2. Le système affiche le message suivant devant le(s) champ(s) concerné(s) **« Erreur, champ obligatoire ».**   **4b.** Le nom du groupe bancaire existe déjà.   1. Le cas d’utilisation reprend à l’étape 1 du scénario nominal sans réinitialisation des champs saisis. 2. Le système affiche le message **« le code du groupe bancaire doit être unique »** devant le champconcerné.   **4c**. L’utilisateur annule la création du groupe bancaire.   1. Le système revient sur l’affichage du formulaire de saisi et le cas d’utilisation se termine en échec. |
| **Post condition** | 1. Le système affiche un message de succès. 2. Le système réinitialise l’écran de saisi. |

* **Diagramme de séquence**

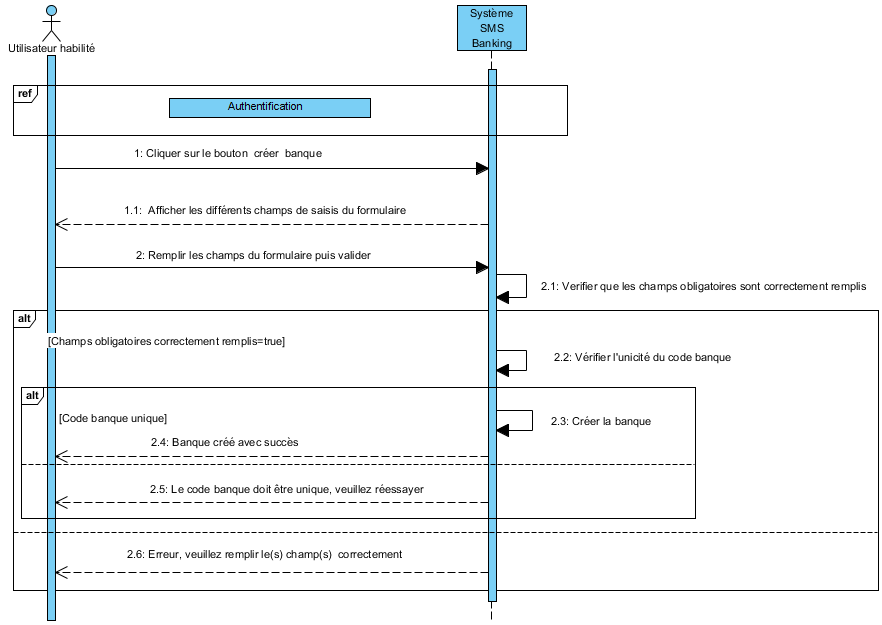


Figure 21 : Diagramme de séquence "créer un groupe bancaire"

1. Cas d’utilisation « Supprimer groupe bancaire »

* **Description textuelle**

|  |  |
| --- | --- |
| **Titre** | **Supprimer groupe bancaire** |
| **Description** | Ce cas permet à un utilisateur habilité de supprimer un groupe bancaire. |
| **Acteur(s)** | Utilisateur(s) habilité(s) |
| **Pré condition** | 1. Etre connecté. 2. Accéder à la liste des groupes bancaires. |
| **Scénario nominal** | 1. L’utilisateur localise le groupe bancaire à supprimer puis clique sur le bouton « **supprimer** ». 2. Le système affiche le message **« Voulez-vous vraiment supprimer le groupe bancaire**   **« NOM\_GROUPE\_BANCAIRE »?**   1. L’utilisateur confirme la suppression du groupe bancaire. 2. Le système supprime le groupe bancaire. 3. Le système notifie l’utilisateur avec le message suivant « **Groupe bancaire supprimé avec succès** » |
| **Scénario alternatif** | 2a. L’utilisateur annule la suppression.   1. Le système termine le cas d’utilisation en échec. 2. Le système affiche la liste des groupes bancaires.   **3a.** Le groupe bancaire est lié à d’autres objets.   1. Le système affiche le message suivant « **L’opération ne peut être effectuée car cet élément est lié à d’autres objets !**» |
| **Post condition** | 1. Le système affiche la liste des groupes bancaires. |

Figure 22 : Cas d'utilisation "Supprimer groupe bancaire"

* **Diagramme de séquence**

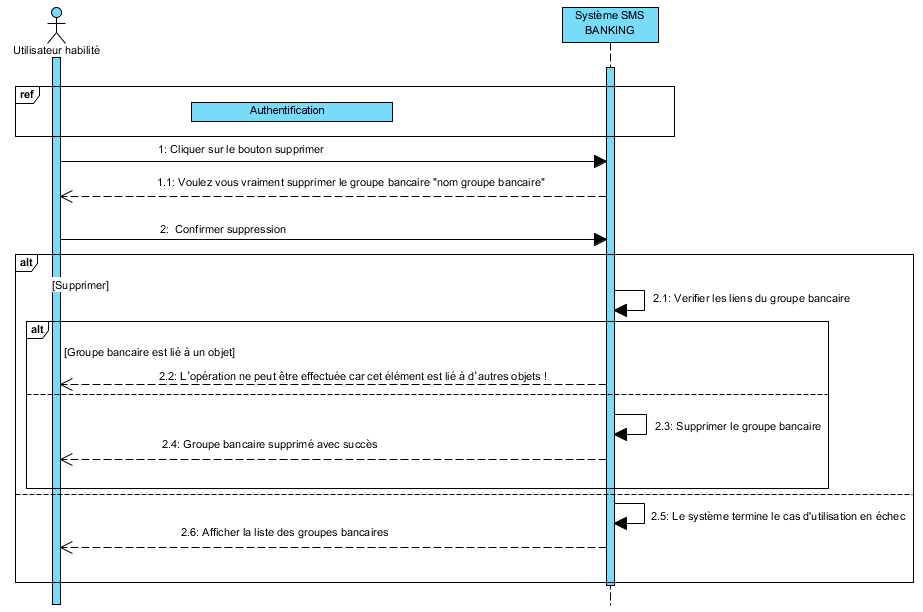
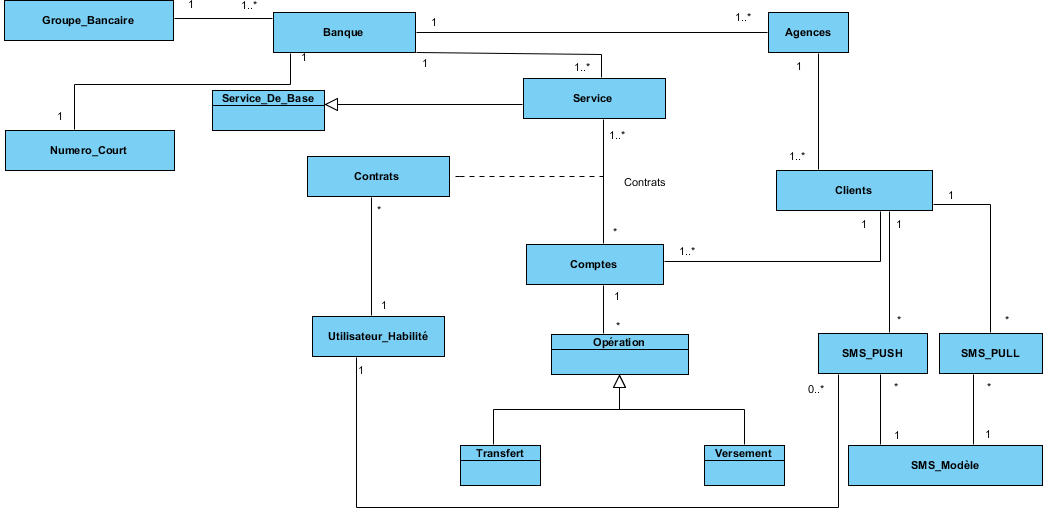


Figure 23 : Diagramme de séquence du cas « Supprimer groupe bancaire »

#### Diagramme de classe

La *Figure* 21 suivant présente le diagramme de classe de notre système.

 Figure : *Diagramme de classe*

### Conception détaillée

Au niveau de la conception détaillée, toutes les questions relatives aux détails de la solution doivent être modélisées. Dans cette phase nous allons :

* Décrire les classes.
* Décrire les associations.
* Décrire les méthodes.
* Passer au modèle relationnel.
* Dresser le dictionnaire de données.

###### Description des classes

Dans le *Tableau 21* ci-dessous, nous décrivons les attributs et les méthodes des différentes classes de notre système.

Tableau 20 : Description des classes

|  |  |  |
| --- | --- | --- |
| Classes | Attributs | Méthodes |
| Groupe Bancaire | Id\_Grb  Libelle\_Grb  Code\_Grb | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| Banque | Id\_Bq  Code\_Bq  Libelle\_Bq  Code\_Pys | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| Agence | Id\_Ag  Libelle\_Ag  Code\_Ag  Adresse\_Ag | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| Client | Id\_Cli  Nom\_Cli  Prenom\_Cli  Tel\_Cli  Code\_Cli | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| Compte | Id\_Cmpt  Num\_Cmpt  Solde\_Cmpt | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| Service | Id\_Serv  Libelle\_Serv  Desc\_Serv  Format\_Serv | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| Contrat | Id\_Ctr  Libelle\_Ctr  Desc\_Ctr  Date\_Cr  Date\_Mod  Date\_Debut  Date\_Fin  Activé | Créer()  Rechercher()  Detail()  Modifier()  Supprimer()  Imprimer() |
| Numéro court | Id\_Ncrt  Numéro\_Ncrt  Code\_Pys  Cout\_SMS  Code\_Devis  Desc\_Ncrt  Activé | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| Opération | Id\_Op  Date\_Op  Montant  Type\_Op | Créer()  Rechercher()  Detail() |
| SMS\_PUSH | Id\_Push  Msg\_Push  Date\_Envoi  Heur\_Envoi  Status\_Push | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| SMS\_PULL | Id\_Pull  Msg\_Recu  Msg\_reponse  Date\_Pull  Heur\_Pull | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |
| SMS\_Model | Id\_Mod  Libelle\_Mod  Msg\_Model | Créer()  Rechercher()  Detail()  Modifier()  Supprimer() |

###### Le passage du modèle objet au modèle relationnel

L’utilisation des SGBDR impose un changement de représentation entre la structure des classes et la structure des données relationnelles. Pour le passage vers le modèle relationnel, nous présentons les règles de passage suivantes :

Tableau 21 : Passage du modèle objet au modèle relationnel

|  |  |
| --- | --- |
| **Modèle Objet** | **Modèle Relationnel** |
| **Classe Table** | Attributs de type simple Colonne |
| **Attributs de type simple** | Colonne |
| **Attributs de type composés** | Colonne ou clé étranger |
| **Instance** | T- Uplet |
| **Objet identifié** | Clé primaire |
| **Association 1 à plusieurs** | Clé étrangère |
| **Association plusieurs à plusieurs** | Table de liens |
| **Héritage** | Clé primaire identique sur plusieurs tables |
| **Méthode** | Ne figure pas |

La *Figure 25* représente la modélisation de la base de données du système. Elle a été réalisée grâce à l’outil JDL-Studio.

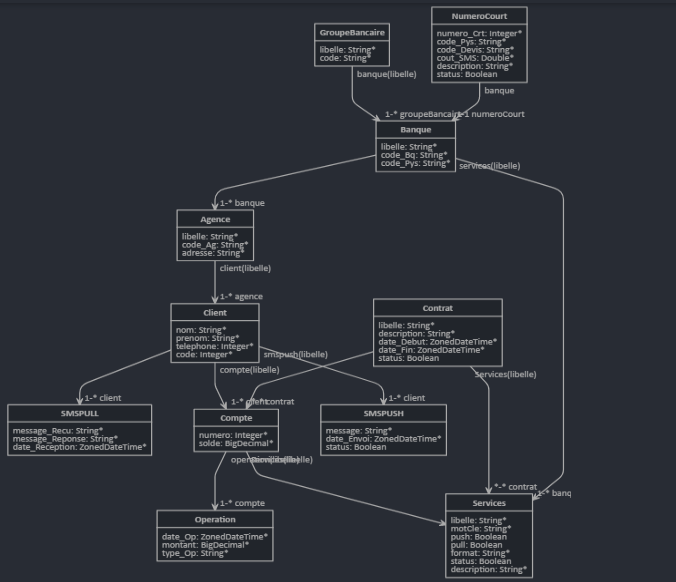
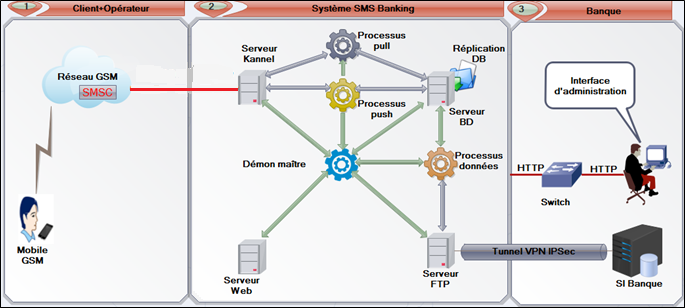


Figure 26 : Modèle relationnel de la base de données

### Architectures générale du système

La *Figure 27* ci-dessous présente l’architecture générale de notre système. Elle est composée de trois grands blocs que sont :

* Client et Opérateur : ce bloc représente la partie composée des différents clients de la banque ou structure financière disposant d’un abonnement chez un quelconque opérateur ainsi que le réseau de ce dernier.
* Système *SMS Banking*: ce bloc représente notre système, il est constitué d’un ensemble de serveurs et de processus ayant des rôles bien définis. C’est la synergie entre ces différents composants qui assure le fonctionnement du système. On retrouve entre autre : un serveur de SMS Kannel, un serveur web, un processus pull, un processus push, un processus de données, un serveur de base de données, une réplication de la base de données, un serveur SFTP et un démon maître.
* Le bloc banque qui représente la partie informatique de la banque ou structure financière.



Canal de communication

Figure 27 : Architecture générale du système

Le système SMS Banking est connecté à la banque à travers un tunnel VPN s’il est héberger au niveau de AtoS et est connecté à l’opérateur réseaux via un canal de communication qui peut être : un modem ou une liaison spécialisée.

Ainsi, dans les différentes architectures proposées ci-dessous, il faut noter qu’il nous faudra toujours :

* Un modem GSM ou une liaison spécialisée qui permet à la banque de se connecter au réseau d'un opérateur de téléphonie en utilisant les cartes SIM ;
* Les téléphones portables dont doivent disposer les clients qui veulent s'abonner à ce service ;
* Une application serveur comprenant toutes les fonctionnalités du *SMS Banking* et qui sera déployée au niveau des serveurs de la banque ;

Les SMS sont acheminés par le réseau de téléphonie mobile des clients vers le serveur et vice-versa. Nous proposons deux façons d’établir une communication entre le serveur de messagerie et le SMSC de l’opérateur :

* Une communication via un modem GSM et
* Une communication via une liaison spécialisée,

Les requêtes-réponses entre le SMS-C de l’opérateur et notre serveur de messagerie sont acheminées différemment, selon le mode utilisé.

#### Communication via un modem GSM

Dans ce type de communication, le modem joue le rôle de passerelle entre notre serveur de messagerie et réseau de l’opérateur. La *Figure 28* suivante présente l’architecture de la communication via un modem GSM.

* **Architecture**

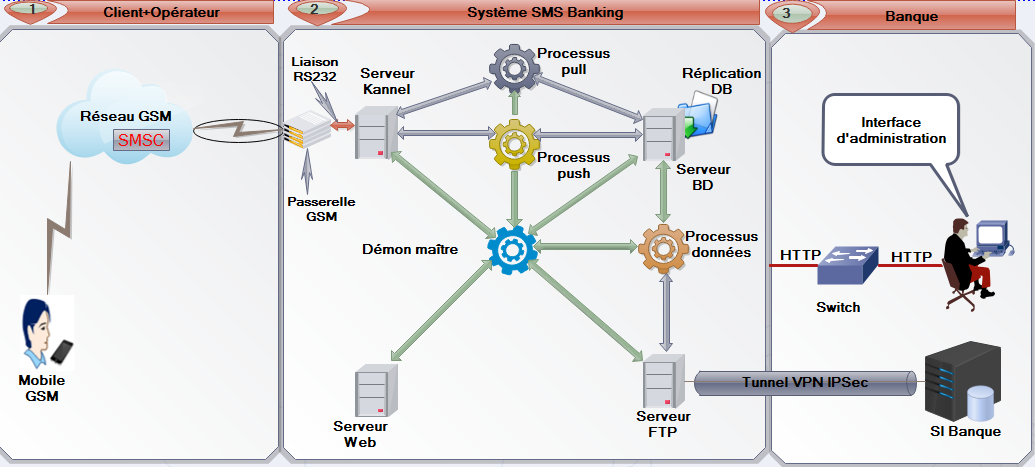


Figure 28 : Liaison entre le SMS-C de l’opérateur et du serveur Kannel via un modem

* **Coût de la solution**

Afin de proposer une optimale, nous évaluons le coût de la solution proposée côté banque, côté client et côté opérateur.

* **Côté banque ou structure bancaire**

Le coût côté banque ou structure bancaire va dépendre du type d’architecture choisie. On propose deux solutions pour cela.

**Solution 1: Serveur Web avec base de données et FTP sur un seul serveur dédié**

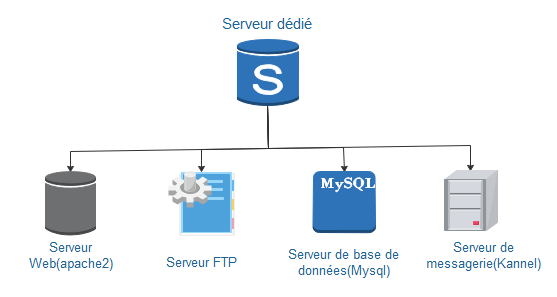


Figure 29 : Architecture de la solution première solution

Cette option est la plus simple en matière d’architecture. Elle consiste à choisir un serveur unique dédié pour héberger le serveur de base de données, le serveur de web, le serveur ftp et les différentes scripts (démons). Dans ce cas, la banque n’aura à acheter qu’un seul serveur et un modem GSM. L’inconvénient de cette solution est que si le serveur dédié tombe, tout le système Banking est paralysé.

Elle se caractérise par :

* Architecture simple
* Coût le plus bas
* Moins extensible

**Solution2 : Serveur Web avec base de données et FTP dissocié**

La *Figure 30* ci-dessous présente l’architecture de la solution avec un serveur web, une base de données et un serveur SFTP dissociés

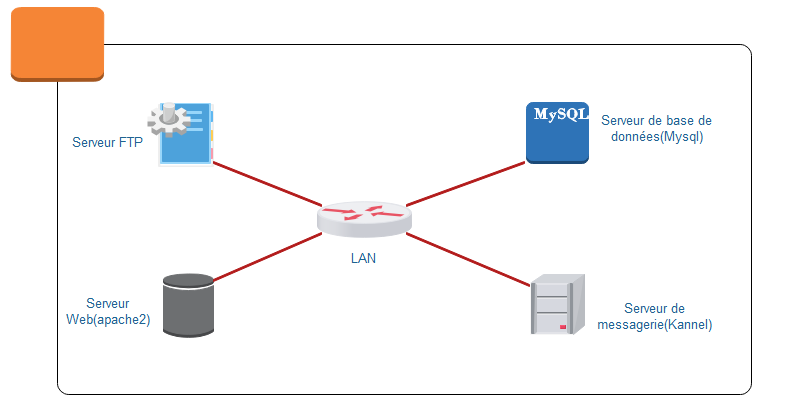


Figure 30 : Architecture de la deuxième solution

Cette option est la plus complexe en matière d’architecture. Elle consiste à choisir un serveur physique pour héberger chaque type de serveur (serveur de base de données, le serveur de web, le serveur ftp). La structure bancaire dépense plus avec ce type d’architecture. Elle achètera quatre serveurs physiques et un modem GSM. Cette solution est caractérisée:

* Architecture complexe.
* Coût le plus élevé.
* Extensible.

Le *Tableau 23* ci-dessous est un récapitulatif des deux solutions proposées :

Tableau 22 : Tableau récapitulatif

|  |  |  |
| --- | --- | --- |
|  | Solution 1 | Solution 2 |
| Nombre de serveur | 1 | 4 |
| Nombre de modem | 1 | 1 |
| Coût | Moins élevé | Plus élevé |

Le choix d’une solution au profit d’une autre dépendra des moyens de la banque ou structure bancaire.

* **Coût côté client**

Pour bénéficier des services de *SMS Banking*, le client de la banque ou structure bancaire effectuera au minium l’achat :

* D’un téléphone portable.
* Montant de la souscription au service défini par la banque.
* **Côté opérateur**

L’opérateur du réseau n’a aucune dépense à effectuer.

#### Communication via une liaison spécialisée

* **Architecture**

La Figure 31 suivant montre l’architecture de la communication via une liaison spécialisée.

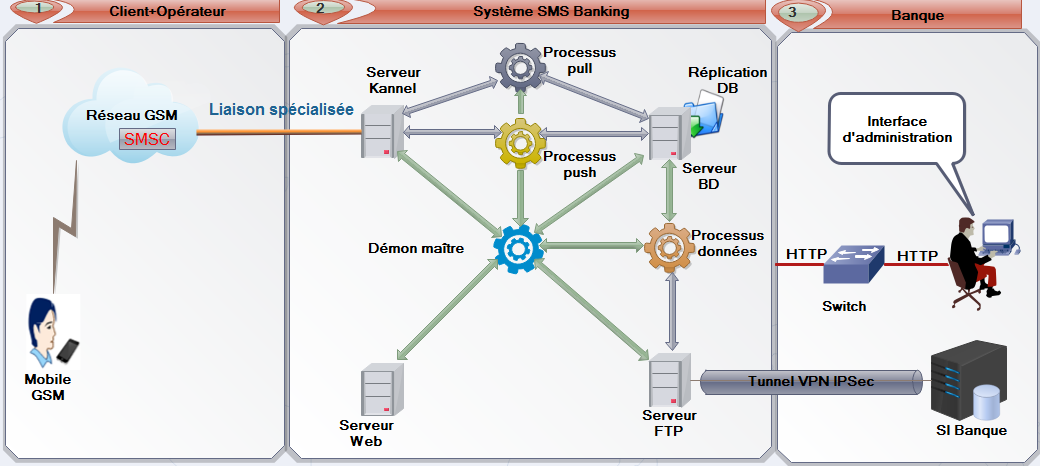


Figure 31 : Liaison du serveur Kannel au SMSC de l’opérateur via une liaison spécialisée

* **Coût de la solution**

Comme pour la première solution, nous évaluons de cette dernière également côté opérateur, côté banque ou structure bancaire et du côté client.

* **Côté banque ou structure bancaire**

Pour cette seconde architecture, le modem est remplacé par une liaison spécialisée. Ainsi le coût du modem est remplacé par le coût de la liaison spécialisée.

* **Coût côté client**

Le coût côté client reste le même.

### Architecture technico-fonctionnelles

#### Architecture fonctionnelle

La communication par SMS entre MS et notre système s’effectue suivant deux modes à savoir le mode PULL et le mode PUSH. Chacun de ces modes présente son propre architecture.

###### Mode PULL

Le SMS en mode PULL est bidirectionnelle. Il est préludé par le client de la banque ou structure bancaire qui veut consulter son compte bancaire, demander un chéquier, etc. Son architecture est représentée par la figure ci-dessous. La *Figure 32* suivant présente l’architecture en mode pull du système.

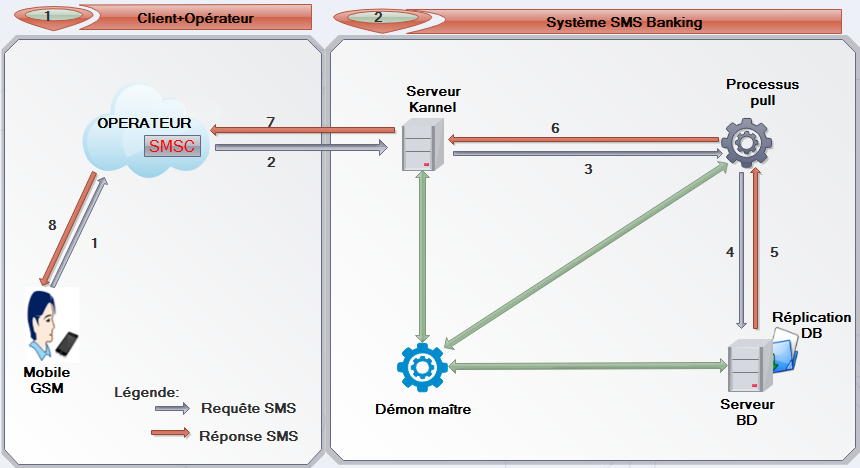


Figure 32 : Architecture du SMS en mode PULL

* **Processus des messages échangés**

Les différentes interactions entre le(s) client(s) et le système *SMS Banking* sont définis comme suit :

1. Le client expédie le message (avec le format prédéfini) en spécifiant le service escompté au numéro court correspondant.

2. Le SMS-C de l’opérateur traite le(s) message(s) reçu(s) puis les envoie soit au modem GSM relié au serveur kannel ou directement au serveur kannel selon l’implémentation choisie.

3. Le serveur kannel déclenche le « processus pull».

4. Le « processus pull » qui tourne en permanence récupère la requête de l’utilisateur du serveur kannel et interroge le serveur de base de données en fonction des informations reçus.

5. Le serveur de base de données renvoi les informations correspondant à la requête au « processus pull ».

6. Le « processus pull » envoie à son tour le message au serveur de messagerie kannel.

7. Le serveur kannel formate la requête reçue au format SMS puis l’envoi au SMS-C de l’opérateur.

8. Le SMS-C de l’opérateur renvoi le message à l’utilisateur final.

###### SMS en mode PUSH

Un message PUSH est un message initié par la banque (ou structure bancaire) et envoyé aux clients de cette dernière. Ce message est peut être une notification prédéfinie ou un message publicitaire.

Le message(SMS) PUSH est envoyé aux clients suivant deux cas :

* Les différentes périodes (Date/Heure) de notification sont planifiées à l’avance par un utilisateur habilité,
* L’utilisateur habilité envoie un message(SMS) de publicité (ou à titre informatif) aux clients ou un message personnalisé à un client particulier.

Dans deux cas, un « démon push » qui tourne en permanence scrute la base de données pour notifier les différents clients.

* **Cas1 : Message initié par un utilisateur habilité**

La *Figure 33* ci-dessous présente le processus suivi par le message initié par un utilisateur habilité.

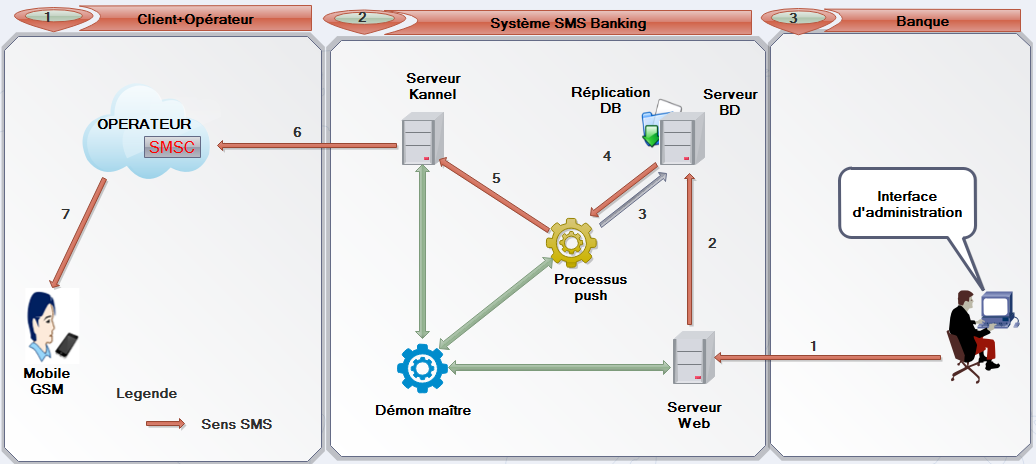


Figure 33 : Architecture d’un message initié par un utilisateur habilité.

* **Description**

L’acheminement du message(SMS) de l’application au client destinataire se déroule de la manière suivante :

1. L’utilisateur habilité renseigne le message à envoyer dans l’interface d’administration puis l’envoie au serveur web.

2. Le serveur web (ou d’application) se charge de stocker le message dans la base de données choisie.

3. Le « processus push » qui tourne en permanence vient scruter la base de données à la recherche de message(s) à envoyer.

4. La base de donné renvoie le(s) message(s) et le(s) destinataire(s) au « processus push ».

5. Le « processus push » envoie les informations reçues de la base de données au serveur kannel.

6. Le serveur kannel formate les informations reçues puis les envoie soit au modem GSM ou directement au SMS-C de l’opérateur, selon l’implémentation (utilisation d’un modem GSM ou d’une ligne spécialisée).

7. Le SMS-C de l’opérateur téléphonique se charge d’envoyer le(s) message(s) au(x) destinataire(x) final(s).

**Cas2 : Message(SMS) déclenché par un évènement**

La *Figure 28* suivante décrit l’architecture d’un message déclenché suite à un évènement (par exemple virement sur compte).

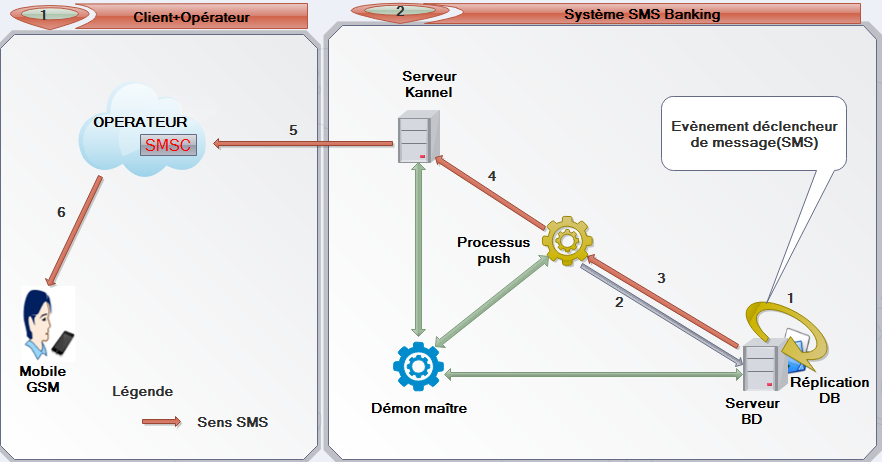
****

Figure 34 : Architecture d'un SMS déclenché par un évènement

* **Description**

Figure 35 : Architecture d'un message déclencher par un évènement

Les notifications envoyées au(s) client(s) par fonction d’une Date/heure prédéfinie ou les messages reçus suite à un mouvement (par exemple virement) sur leurs comptes bancaires suit le processus suivant :

1. Déclenchement de l’évènement suit à un changement d’état sur un compte.

2. Le «processus push » constate le déclenchement de l’évènement en scrutant la base de sonnées puis envoie des requêtes pour récupérer les différentes informations à notifier au(x) client(s).

3. La base de données envoie au « processus push » les informations correspondant à la requête.

4. Le « processus push » envoie les informations reçues de la base de données au serveur kannel.

5. Le serveur kannel formate les informations reçus puis les envoie soit au modem GSM ou directement au SMS-C de l’opérateur, selon l’implémentation (utilisation d’un modem GSM ou d’une ligne spécialisée).

6. Le SMS-C de l’opérateur téléphonique se charge d’envoyer le(s) message(s) au(x) destinataire(x) final(s).

#### Architecture logicielle

Notre proposons une solution basée sur le modèle 3-tiers qui consiste à diviser l’application en couches séparées : **Couche base de données**, **Couche persistance**, **Couche middle** Et la **couche front**. La  *Figure* 36 montre l’architecture de l’application.

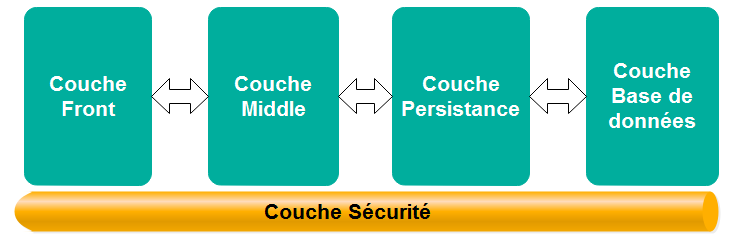


Figure 36 : Les couches de l'application

Cette architecture est simpliste, elle sera affinée pour être fidèle à l’architecture finale de notre application après le choix des outils et technologies.

## Conclusion

Dans ce chapitre, nous avons énuméré l’ensemble de fonctionnalités de notre système. Nous avons également décri leur cas d’utilisation ainsi l’étude textuelle de ces derniers. Ensuite, nous avons conçu les différentes architectures pour la mise en place d’un système performant. Dans le dernier chapitre suivant nous les différentes technologie utilisées ainsi que la solution proposée.

# IMPLEMENTATION DE LA SOLUTION

***Résumé****:*

*Dans dernier ce chapitre, nous abordons trois points essentiels que sont : le choix des outils et technologique, l’architecture détaillée de la plateforme, ainsi que la présentation de l’aperçu de l’application IHM[[7]](#footnote-7). Nous présentons d’abord l’architecture en couche de l’application, effectuons le choix des outils et technologie puis terminons avec la présentation de la solution proposée.*

Sommaire

V.1 Choix des outils et des technologies 79

V.1.2 Architecture de technique détaillée de la plateforme 79

V.1.3 Choix des outils et mise en place l’environnement de développement 79

V.1.4 Choix des outils et mise place de l’environnement de production 83

V.2 Aperçu des résultats 83

## Choix des outils et des technologies

Dans cette section il s’agit de présenter les différents outils et technologies utilisés dans la conception du système *SMS Banking*. Le choix des technologies de développement découle de plusieurs éléments tels que les enjeux du projet et la culture de l'entreprise.

### Architecture de technique détaillée de la plateforme

Nous définissons ici l’architecture en couche détaillée de notre l’application. Elle présente les différents blocs de la plateforme avec les différentes technologies utilisées. La *Figure 31* suivant donne un aperçu de cette architecture.

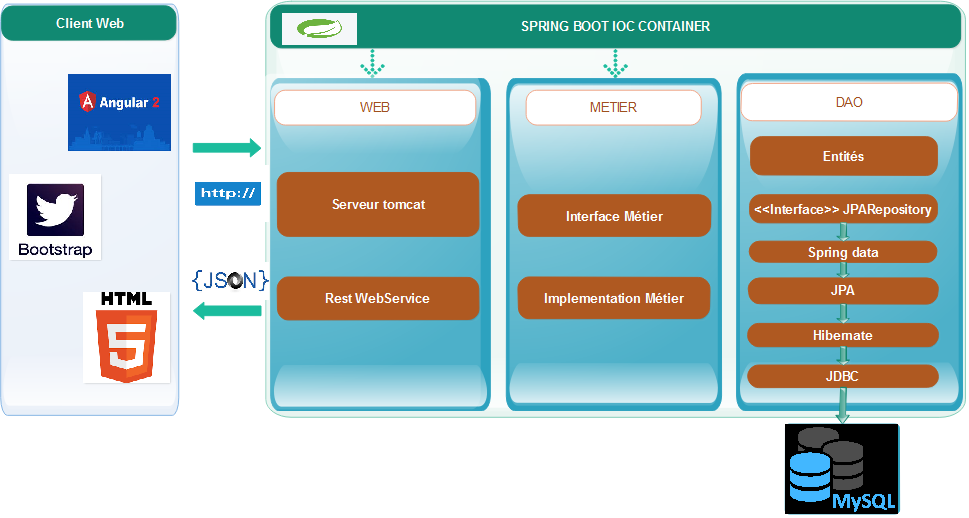


Figure 37:Architecture détaillée de la plateforme

### Choix des outils et mise en place l’environnement de développement

Cette sous-section a pour objectif de présenter les outils dans l’environnement de développement.

#### CentOS

*CentOS* est une distribution Linux orientée Entreprises, basée sur les [sources de Red Hat Enterprise Linux, disponibles de manière libre et gratuite](ftp://ftp.redhat.com/pub/redhat/linux/enterprise/). Chaque version de CentOS est supportée pendant 10 ans (par des mises à jour de sécurité**) [W14].** On utilise le système CentOS pour l’environnement de développement et de production.

#### Base de données

 *MySQL* pour la conservation des données de la plateforme, l’utilisation d’une base de données est indispensable. Il existe plusieurs Systèmes de Gestion de Base de Données (SGBD) dont les plus présents sur le marché sont MySQL, PostgreSQL et Oracle. Nous utiliserons la base de données MySQL vue la comparaison effectuée suite à la demande de l’entreprise et aussi cette comparaison est venue consolidée le choix de cet outils qui fait partie des exigences de l’entreprise. Ce choix est plus justifié par le *Tableau 24* suivant en notant que l’entreprise privilégie les solutions Open Source.

Tableau 23 : Tableau comparatif SGBD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MySQL/PostgreSQL | IBM DB2 | MS SQL | Oracle SQL |
| Cluster | Répliqué | Distribué | Partagé | Partagé |
| Réplication | C:\Users\A661546\Desktop\archi1\téléchargement.jpg | C:\Users\A661546\Desktop\archi1\téléchargement.jpg | C:\Users\A661546\Desktop\archi1\téléchargement.jpg | C:\Users\A661546\Desktop\archi1\téléchargement.jpg |
| Procédures  Stockées | C:\Users\A661546\Desktop\archi1\téléchargement.jpg | C:\Users\A661546\Desktop\archi1\téléchargement.jpg | C:\Users\A661546\Desktop\archi1\téléchargement.jpg | C:\Users\A661546\Desktop\archi1\téléchargement.jpg |
| Type de sauvegarde | Complète | Complète  Différentielle  Incrémentielle | Complète  Différentielle  Incrémentielle | Complète  Différentielle  Incrémentielle |
| Type de support | Communautaire | Entreprise | Entreprise | Entreprise |
| OS Compatible | Windows/MAC/Linux | Windows/MAC/Linux | Windows | Windows/MAC/Linux |
| Licence | Gratuit | Cœur | Utilisateur/Coeur | Utilisateur/Coeur |

#### Git

 *Git* est un logiciel de gestion de versions décentralisé dont la principale tâche est de gérer l’évolution du contenu d’une arborescence. Il permet entre autres :

* De suivre l’évolution du code source (modifications effectuées sur les fichiers sources) et ainsi être capable de revenir sur une version précédente en cas de problème.
* De travailler à plusieurs sur un même projet sans conflits de modifications.

Nous nous sommes servis de ce logiciel pour faire des *commits* du travail effectués chaque jour sur les différentes branches.

#### phpMyAdmin

 *phpMyAdmin* est une application web pour les systèmes de gestion de base de données. Nous l’utilisons pour assurer une meilleure la gestion de notre base de données MySQL.

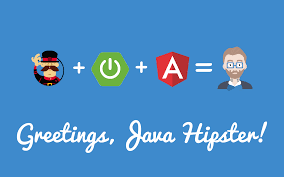
#### Visual paradigme

C:\Users\A661546\Desktop\archi1\paradigme.PNG *Visual Paradigm* est un éditeur qui propose des outils pour la conception et la gestion lors de développement de systèmes informatiques. Parmi cette suite d’outils figure Visual ParadigmFor UMLque nous avons utilisé pour créer nos modèles*.* Il inclut tous les diagrammes UML 2.0 et offre une interface très intuitive aux modélisateurs. Nous l’avons utilisé pour la modélisation de nos différentes fonctionnalités.

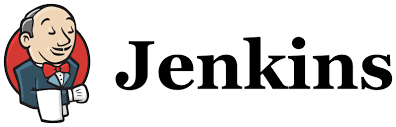
#### Angular2

Ici nous avons utilisé le Framework JavaScript Angular Js dans sa version 2 pour le développement de la partie FrontEnd de notre application. Angular Js est un Framework JavaScript pour les SPA (Single page application) créé par Google en 2010 pour simplifier la création d’application web.

#### Jhipster

* JHipster* est un assemblage d’outils pour faciliter la création/initialisation du projet, le développement et fournir des services standards d’une web app(authentification, [mvvm](http://fr.wikipedia.org/wiki/Mod%C3%A8le-Vue-VueMod%C3%A8le)). Pour utiliser JHipster il faut au minimum installer **Nodejs**, **Yeoman**, **Maven** ou **Gradle** et un JDK. Il existe également une image Docker pour tester facilement **[W15].** *JHipster* nous a permis de générer un projet de base pour aller plus vite dans le développement.

#### Jenkins

 *Jenkins* est un serveur d'automatisation open source. Jenkins gère

et contrôle les processus du cycle de vie du développement de toutes sortes, y compris la construction, le document, le test, le paquet, le déploiement, le déploiement, l'analyse statique et beaucoup d'autre **[W16]**. Cette pratique nous a permis d’intégrer le travail fréquemment et d’être sûr que les modifications du code n’ont pas affectées le bon fonctionnement du logiciel. Pour sa mise en place il faut installer :

* un serveur web : **Apache2**
* un serveur de base de données : **MySQL**
* un serveur d’application : **tomcat7**
* un outil de versionning : **Git**

Nous avons choisi ces différents serveurs dans le cas de notre projet.

#### SonarQube

SonarQube (précédemment Sonar) est un logiciel qui permet de mesurer la qualité d’un code en continue. Cet outil permet entre autres de détecter des erreurs d'inattention telles que des variables oubliées, des paramètres de fonction inutilisés, ou encore des parties de codes qui ne s'exécutent jamais. Il nous a permis d’assure la qualité du code produit.

#### Pencil

 ***Pencil*** est un logiciel de création de maquettes libre et gratuit développé par **Evolution Solutions** (Evolus). Il est utilisé afin de créer des [diagrammes](https://fr.wikipedia.org/wiki/Diagramme) et des maquettes d'interface graphique de logiciels. Cet outil nous a permis de faire tout la maquette de notre application.

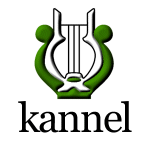
### Choix des outils et mise place de l’environnement de production

Dans cette section, nous déterminons les différents outils nécessaires à la mise en place de l’environnement de production.

#### Base de données

Nous utilisons la base de données mySQL comme système de gestion de base de données dans l’environnement de production. Son administration se fera à partir de la console.

#### Kannel

 Kannel est un logiciel open source développé en langage C qui utilise une passerelle WAP et SMS. En outre, il permet également de gérer le push SMS (c’est-à-dire d’envoyer des messages aux téléphones portables), le pull SMS (la réception d'un SMS ou la réception de requête) ou pull-push SMS (l'envoi de la réponse après traitement à partir de la plateforme). Concernant les fonctionnalités SMS, la passerelle Kannel supporte les principaux protocoles SMS .La passerelle Kannel est un outil très intéressant car nous permettant de développer des applications utilisant le serveur Web Apache ou d’application (tomcat, JBoss). [W17]. Nous avons opté pour l’étude de l’architecture de kannel ainsi que son installation (cf. ***Annexe A)***.

## Aperçu des résultats

Nous allons maintenant présenter le travail effectué à l’aide de quelques captures d’écran accompagnées de description au besoin.

### Authentification

Nous présentons l’interface d’authentification qui est une phase primordiale pour accéder à l’interface d’administration. L’interface d’accueil après authentification diffère, selon le profil de ce dernier. *La Figure 31* ci-dessous montre l’interface d’authentification.

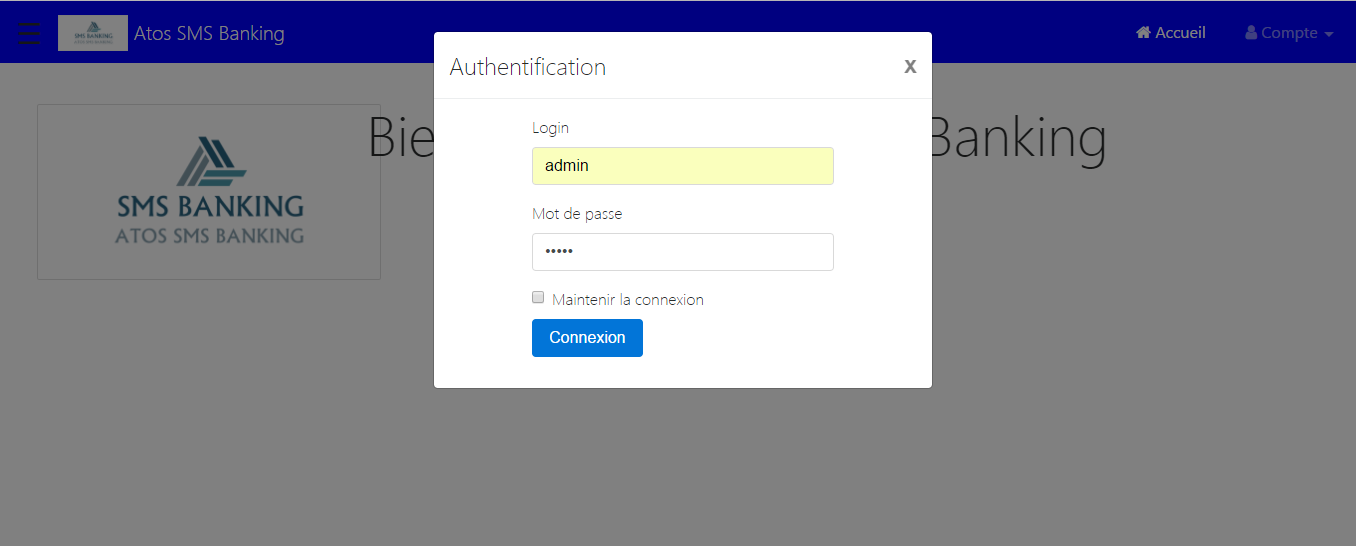


Figure 38 : Interface d'authentification

### Définition d’un contrat

La souscription à un contrat est la première phase à effectuer pour bénéficier des différents services offerts par le système. Nous proposons une interface de souscription de contrat dans la mesure où la banque voudrait utiliser l’application pour effectuer cette tâche. La *Figure 37* suivant présente cette interface.

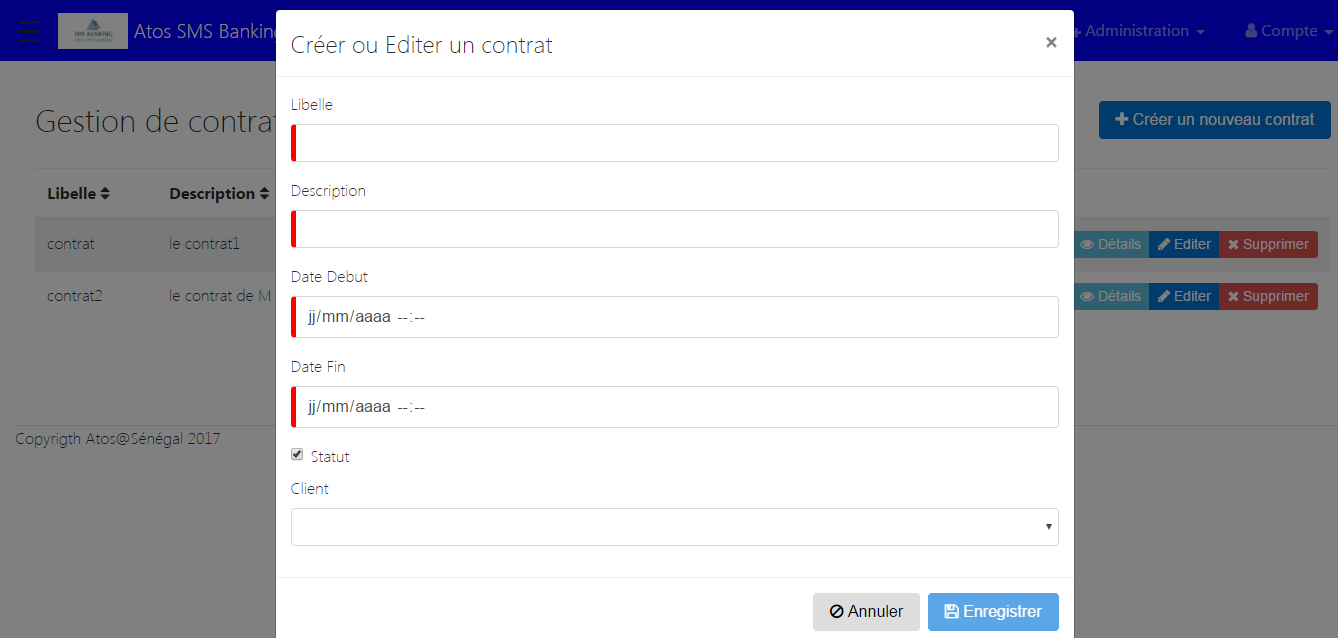
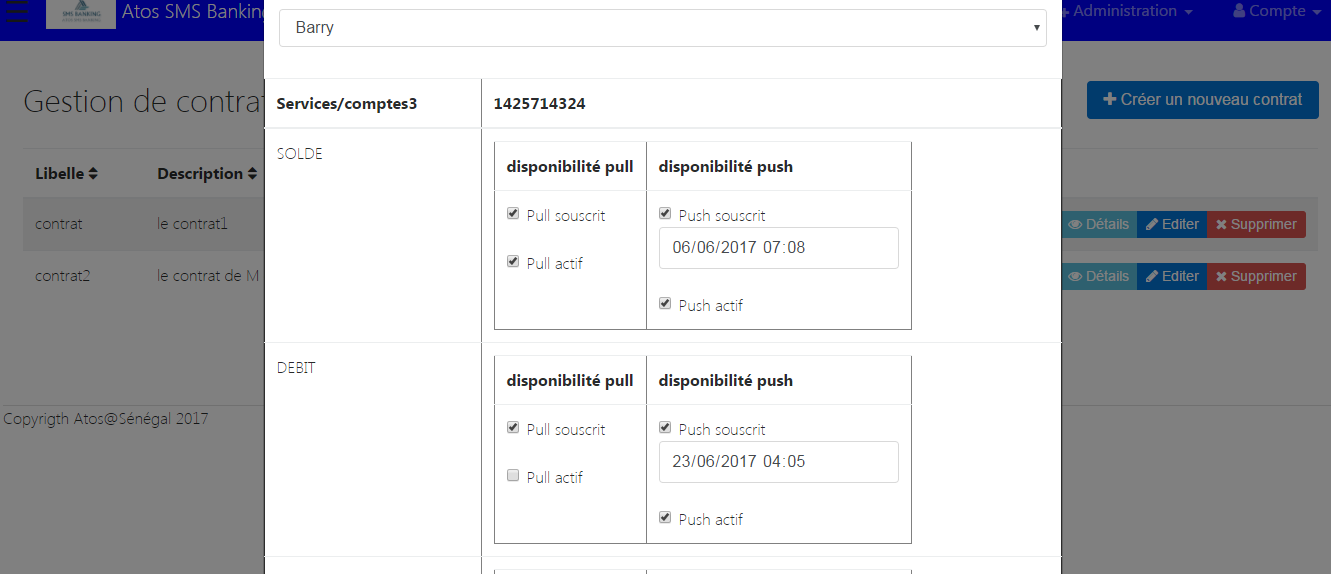


Figure 39 : Création d'un contrat

Après avoir choisie le client auquel associer au contrat, l’interface suivante apparaît pour le paramétrage des services.



### Planifier un message PUSH

Cette interface permet à l’administrateur de la banque ou structure financière de paramétré l’envoi d’un message à un client ç une heure précise. La *Figure 38* ci-dessous met en exergue cette interface.

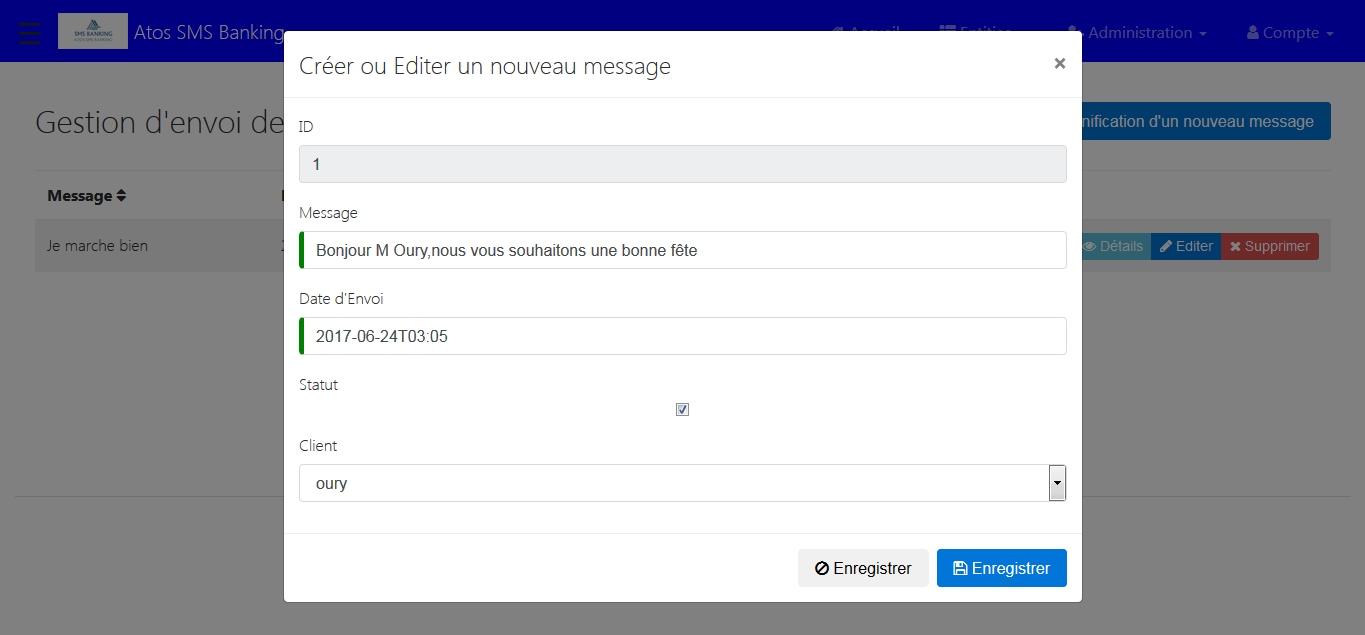


Figure 40 : Interface de planification de message PUSH

### Historique des messages

L’interface La *Figure 39* ci-dessous affiche l’historique des différents messages échangés entre le serveur de messagerie et les clients.

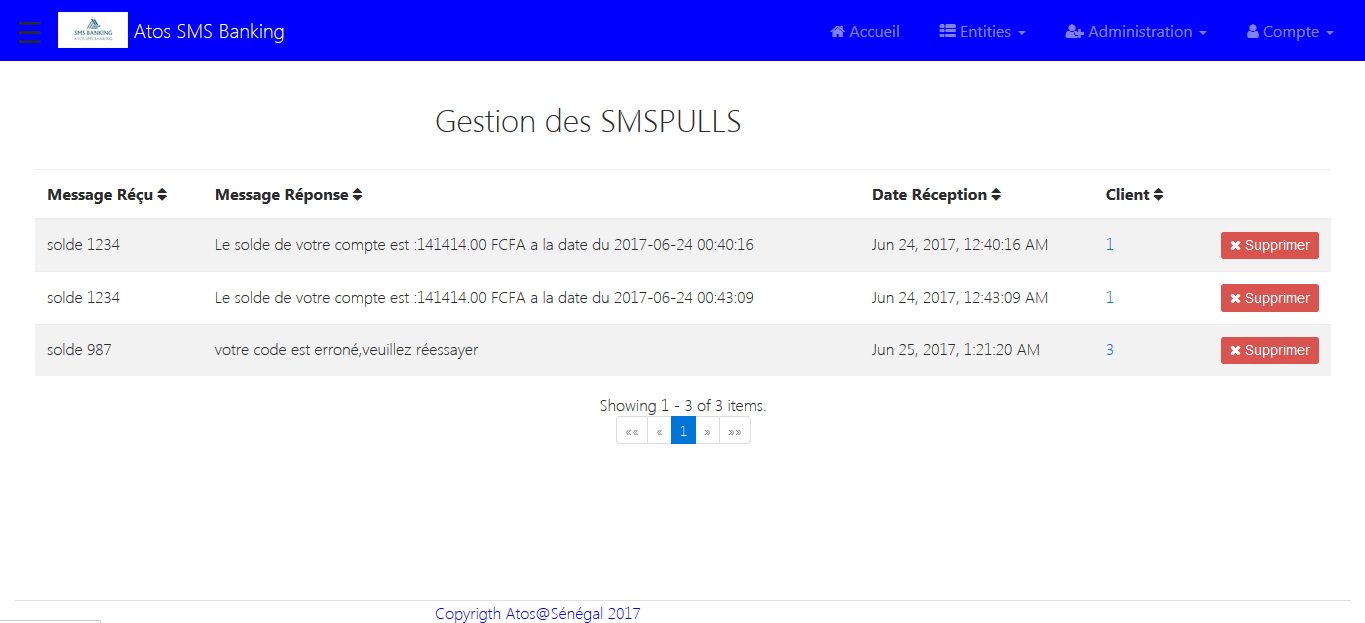


Figure 41 : Historique de SMSPULL

## Conclusion

Ce dernier chapitre nous a permis de présenter les différentes technologies pour la réalisation de la solution proposée. On y a également présenté quelques interfaces du résultat obtenu après le développement de la solution.

***CONCLUSION et PERSPECTIVES***

La conception et l’implémentation d’une solution de *SMS Banking* a été l’objectif du travail que nous avons effectué tout au long de ces cinq derniers mois au sein de la prestigieuse entreprise AtoS Sénégal. Le *SMS Banking* est un sujet d’actualité s’adressant à toute institution bancaire désireux d’offrir à ses clients une pléthore de services basée sur la technologie SMS.

Ce sujet nous a donné l’occasion d’aborder la conception du système SI en adoptant une méthode orientée objet permettant d’utiliser le langage de modélisation UML. Il nous a également permis de mieux appréhender la technologie de SMS, les protocoles de communication des serveurs de SMS, la gestion de projet, le langage angular2, la technologie JEE, la technologie Jhipster ainsi que les technologies d’intégrations continues(Jenkins et SonarQube). Pour atteindre les objectifs fixés, il a fallu associer des compétences en télécommunication et en informatique.

Le système développé lors notre stage permettra à AtoS Sénégal d’offrir aux banques ou structures bancaires le service de *SMS Banking* en deux volets : SMS PUSH et SMS PULL, permettant d’augmenter la satisfaction de leurs clients qui vont conduire à un meilleur taux de rétention grâce aux fonctionnalités offertes par le système:

* Communiquer avec les clients grâce au mode personnalisé ;
* Permettre aux clients de consulter leur comptes à tout instant ;
* Permettre aux clients d’être notifier par SMS pour tout mouvement sur leur comptes ;
* etc.

Cependant, il faut noter que notre système dépend fortement du réseau de l’opérateur mobile.

***« Si on croit être arrivé c’est qu’on n’avait pas l’intention d’aller très loin. »***

Arriver à une solution n’est pas une fin en soi. A long terme, nous prévoyons:

* D’implémenté la fonctionnalité transfert sécurisée grâce à l’établissement d’une connexion crypté depuis le client jusqu’au serveur du système *SMS Banking*. Pour cela nous envisageons de développer une application mobile qui sera exigé pour tous les clients de la banque.
* De mettre en place un système de mobile Banking.
* D’utiliser l’USSD pour la PULL et le SMS pour la partie PUSH de notre système.

***Bibliographie/Webographie***

***Bibliographie***

**[B1] :** memoire-sms-banking.pdf, date de dernière consultation le 28/05/2017

**[B2] :** memoire-sms-banking.pdf, date de dernière consultation le 28/05/2017

**[B3] :** memoire-sms-banking.pdf, date de dernière consultation le 28/05/2017

**[B4]** : memoire-sms-banking.pdf, date de dernière consultation le 28/05/2017

**[B5]** : <http://www.efort.com/r_tutoriels/SECURITE_MOBILE_EFORT.pdf,consulter> le 28/05/2017

**[B6]** : Titre de l’ouvrage, « continuous-integration-with-hudson.pdf », date de dernière consultation le 28/052017.

***Webographie***

**[W1]** : <http://www.journaldunet.com/ebusiness/internet-mobile/1009553-monde-le-nombre-d-abonnes-au-telephone-mobile/>, date de dernière consultation le 25/05/2017

**[W2]:** <http://www.artpsenegal.net/images/documents/Rapport_Observatoire_T2_2016.pdf>, date de dernière consultation le 26/05/2017

**[W3]** : <http://connect.ed-diamond.com/GNU-Linux-Magazine/GLMF-091/Developpement-de-services-SMS>, date de dernière consultation le 16/05/2017.

**[W4]** : http://connect.ed-diamond.com/GNU-Linux-Magazine/GLMF-091/Developpement-de-services-SMS, date de dernière consultation le 16/05/2017.

**[W5]** : <http://connect.ed-diamond.com/GNU-Linux-Magazine/GLMF-091/Developpement-de-services-SMS>», date de dernière consultation le 16/05/2017.

**[W6] :** <http://www.technologuepro.com/gsm/commande_at.htm> , date de dernière consultation le 17/05/2017.

**[W7]** : http://www.memoireonline.com/06/11/4548/m\_Conception-et-implementation-dun-systeme-de-gestion-des-centres-dacces-a-internet--rechargemen8.html, date de dernière consultation le 28/05/2017.

**[W8] :** http://communication.clever.fr/sms-smpp-envoyer-des-sms-client-smpp/, date de dernière consultation le 28/05/2017.

**[W9] :** http://www.lb.refer.org/memoires/669814JosephHADDAD.pdf, date de dernière consultation le 29/05/2017.

**[W10]** : http://igm.univ-mlv.fr/~dr/XPOSE2010/Lecharpentier\_Jenkins/purpose.html, date de dernière consultation le 30/05/2017

**[W11]** : https://www.supinfo.com/articles/single/2976-testez-qualite-votre-code-avec-sonarqube, date de dernière consultation le 30/05/2017

[W9] : http://xavier.lienart.pagesperso-orange.fr/gdp/def\_projet.html, date de dernière consultation le 30/05/2017

**[W10]** : http://www.access-dev.com/access-dev/la-gestion-de-projet-methodes-classiques-vs-methodes-agiles/, date de dernière consultation le 01/06/2017

**[W11]** : <https://fr.scribd.com/doc/49697489/Processus-de-Developpement-Y-Processus-2TUP>, date de dernière consultation le 01/06/2017

**[W14]** :  <https://wiki.centos.org/fr>, date de dernière consultation le 01/06/2017

**[W15]** : <http://blog.ippon.fr/2014/11/19/jhipster-dans-la-pratique/>, date de dernière consultation le 01/06/2017

**[W16] :** <https://www.cloudbees.com/jenkins/about>, date de dernière consultation le 01/06/2017

**[W17] :** <https://dourma.wordpress.com/2012/04/18/kannel-pour-lenvoi-de-sms-via-une-page-web/>, date de dernière consultation le 03/06/2017

# Annexes

1. Présentation de Kannel

Kannel est une passerelle WAP et SMS Open Source (source libre). Lancé en mars 1999, le projet est à l’initiative de la compagnie Finlandaise WAPIT. La passerelle possède trois interfaces chacune ne pouvant communiquer qu’avec un type d’équipement spécifique :

* Les centres SMS (SMSC), utilisant divers protocoles.
* Les serveurs HTTP, pour les contenus WAP et SMS et fournissent le contenu des « WAP push ». HTTP est utilisé pour le « pull », et PAP pour le push.
* Les terminaux WAP, implémentant la pile de protocole WAP et (pour le « push ») WAP Push client.

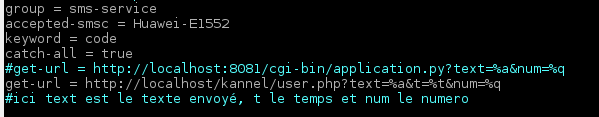
1. Les caractéristiques de Kannel : « BOXES »

Kannel divise ses différentes fonctions selon trois processus appelés « boxes » basés essentiellement sur le type d’équipement externe avec lequel il veut dialoguer.

* **Le Bearerbox** implémentant le niveau porteur du WAP. Il permet entre autre la connexion aux différents SMSC.
* **Le SMSbox** implémente l’essentiel des fonctionnalités de la passerelle SMS. Il reçoit les messages texte depuis le Bearerbox et les interprète comme des requêtes vers des services et y répondent en utilisant le chemin approprié.
* **Le Wapbox** implémente la pile de protocole WAP et le WAP Push (protocole de niveau applicatif). Il n’est possible de mettre en place qu’un seul bearerbox tandis qu’il est possible de disposer de plusieurs Wapbox et SMSboxe.

1. Exemple de paramétrage d’un service

La figure ci-dessous montre un exemple de paramétrage d’un service au sein de serveur de de SMS Kannel.



Le mot clé qui déclenche ce service. On définit autant de mot clé qu’il y a de service.

Requête a exécutée lorsque le mot clé code est reçu par le serveur.

1. Connexion de Jenkins à github

La Figure ci-dessous montre la connexion de Jenkins à github.

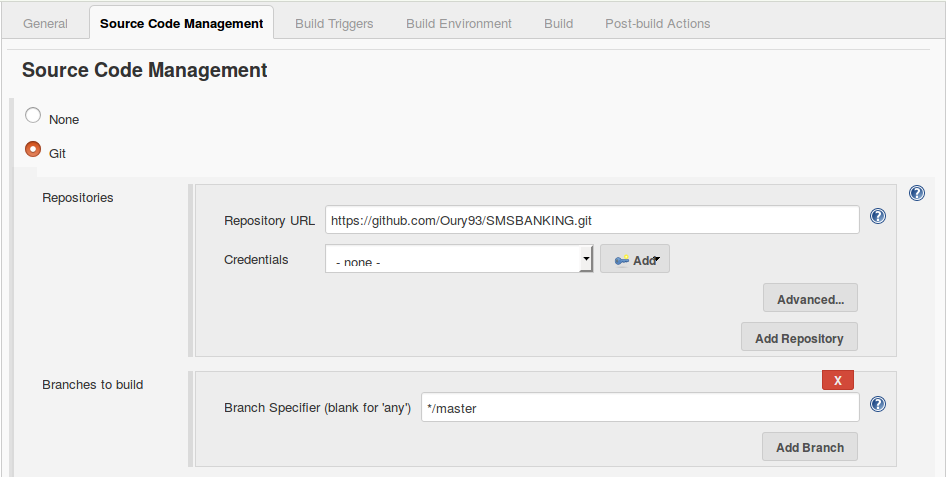


Figure 42 : connexion de Jenkins à github

Après avoir connecté Jenkins à github, le projet à superviser est automatiquement importé dans le serveur Jenkins.

La *Figure 41* suivant illustre cela.

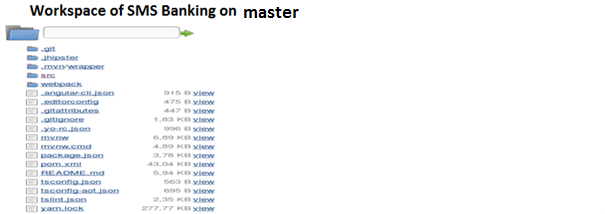


Figure 43 : Importation du projet dans Jenkins

1. e-Banking : est un service qui permet aux clients d’une banque de gérer leur compte depuis une interface web. [↑](#footnote-ref-1)
2. Structure financière : elle représente la banque ou structure bancaire dans notre cas [↑](#footnote-ref-2)
3. Le terme **bancarisation** désigne premièrement un processus mené par les banques, qui consiste à ouvrir des comptes bancaires aux ménages, aux particuliers et aux entreprises dans un pays. [↑](#footnote-ref-3)
4. La digitalisation est le procédé qui vise à **transformer un objet**, un outil, un process ou un métier en un code informatique afin de le remplacer et le **rendre plus performant.** [↑](#footnote-ref-4)
5. Service internet permettant d’accéder à tous ses comptes à tout moment [↑](#footnote-ref-5)
6. X.25 est un protocole de communication normalisé par commutation de paquets en point à point offrant de nombreux services. [↑](#footnote-ref-6)
7. Interface Homme-Machine [↑](#footnote-ref-7)