

Année Académique : 2019-2020

N° d'ordre :

MEMOIRE DE FIN D'ETUDES EN MASTER

Stage effectué à la Société Générale Burkina Faso du 05/02/2021 au 06/07/2021

Thème :

Gestion des garanties bancaires : cas de SGBF

Soutenu publiquement le 08/03/2022 par :

SORE Assané

Licence Professionnelle en Système d'Informations et Réseaux

Pour l'obtention du diplôme de
MASTER EN SYSTEMES D'INFORMATIONS ET RESEAUX

Directeur de Mémoire

TINDANO Yonouyaba Olivier, Ingénieur en
Informatique, Enseignant à l'ESTA, Ouagadougou

Maître de stage

SEDGO Thierry Paulin, Ingénieur en Informatique,
Responsable Service Innovation à la SGBF,
Ouagadougou

Composition du Jury

Président

Pr. SIE Oumarou, Professeur Titulaire en Informatique, Université Joseph KI-ZERBO

Membres

TINDANO Yonouyaba Olivier, Ingénieur en Informatique, Enseignant à l'ESTA, Ouagadougou
OUEDRAOGO Dieudonné, Ingénieur en Informatique, ESTA, Chef de Département Informatique

Sommaire.....	i
Biographie	iv
Dédicace.....	v
Remerciements	vi
Sigles et Abréviations	vii
Glossaire	ix
Résumé.....	xi
Liste des Figures.....	xii
Liste des Tableaux.....	xiv
Préambule.....	xv
I. Présentation de l'ESTA.....	xv
II. Formations de l'ESTA.....	xv
Introduction Générale.....	1
I. Introduction du sujet	1
II. Contexte et justification (ou problématique)	1
III. Objectifs	1
IV. Hypothèses (?)	2
V. Questions de recherche	2
VI. Intérêt de l'étude	2
VII. Méthodologie de collecte des données.....	3
VIII. Structure d'accueil	4
Chapitre I : Revue de littérature	7
Introduction.....	8
I. Les garanties bancaires.....	8
II. Le risque de crédit.....	10

Conclusion	10
Chapitre II : Etude fonctionnelle du système	11
Introduction.....	12
I. Critique de l'existant.....	12
II. Les fonctionnalités identifiées	15
III. Démarche de développement	20
IV. Planification du projet	28
Conclusion	30
Chapitre III : Etude technique du système.....	31
Introduction.....	32
I. Architecture logicielle.....	32
II. Architecture Réseau	34
III. Méthodologie de développement : Démarche MVC	34
IV. La plateforme PHP	36
V. Frameworks	38
VI. Environnement de travail	43
Conclusion	47
Chapitre IV : Conception détaillée du futur système	48
Introduction.....	49
I. Diagramme de cas d'utilisation	49
II. Diagramme d'activité	64
III. Diagramme de séquence	67
IV. Diagramme de classes	71
V. Diagramme déploiement	77
Conclusion	78

Chapitre V : Implémentation ou mise en œuvre du futur système..... 79

Introduction.....	80
I. Déploiement de Lafia Garanties	80
II. Les tests.....	81
III. Politique de sécurité.....	82
IV. Coût de mise en œuvre	84
V. Charte et quelques interfaces graphiques	88
VI. Bilan et perspectives	95
Conclusion	97
Conclusion Générale.....	98
Bibliographie	99
Webographie.....	100
Annexe.....	101
Table des matières	102

Biographie

SORE Assané est actuellement **développeur web Full-Stack** à la **Société Générale Burkina Faso (SGBF)** et étudiant en **Master** à l'**Ecole Supérieure des Techniques Avancées (ESTA)**. Comment étaient ses débuts ?

Adolescent et au cours du cycle d'études primaires, la passion de l'informatique plus précisément la création de logiciel l'anime. Cela s'est traduit par sa fréquentation de nombreux cybercafés et par son appétence pour les recherches d'informations sur le thème de l'informatique en général, et le génie logiciel en particulier. Après l'obtention du **BAC Série D** au **Lycée Provincial Bafuji de Gaoua (LPBG)**, les multiples recherches sur l'Internet les conseils d'orientation des proches l'ont conduit inéluctablement à opter de poursuivre sa formation dans la filière **Système d'Informations et Réseaux (SIR)** au sein de l'ESTA en **octobre 2015**. Ce choix lui a déjà permis de vivre sa passion par la réalisation de nombreux projets estudiantins, projets présentés notamment lors des **Journées Portes Ouvertes (JPO)** et lors de la validation de la filière par le **Conseil Africain et Malgache pour l'Enseignement Supérieur (CAMES)**. Ces projets ont été très appréciés par les promotionnaires, le personnel de l'établissement et des visiteurs des JPO venus du monde professionnel. Son premier stage débute à sa première année d'études au sein de la structure **ASN Technology** en **juillet 2016** et le second à sa deuxième année au sein de la même structure début **Août 2017**. Ce dernier stage a fait l'objet d'un rapport ayant servi à l'obtention de son **Brevet de Technicien Supérieur d'Etat (BTS)** en **décembre 2017**. Désormais technicien supérieur des travaux sa passion ne s'est pas arrêté là. A la fin de l'année académique 2017-2018, il entame un stage à la structure financière **YUP** filiale de SGBF pour l'obtention du diplôme de **Licence Professionnelle** le portant au titre d'ingénieur des travaux informatique au rang de major de la promotion. Riche de ces trois années d'expériences professionnelles, il signe son premier contrat en tant que **développeur Full-Stack** à la SGBF début **janvier 2019**, son poste actuel. Ce contrat ne lui enlève pas cette envie de continuer parallèlement ses études pour l'obtention du diplôme de **Master Professionnel**, d'où l'intérêt de la rédaction de ce présent mémoire.

Dédicace

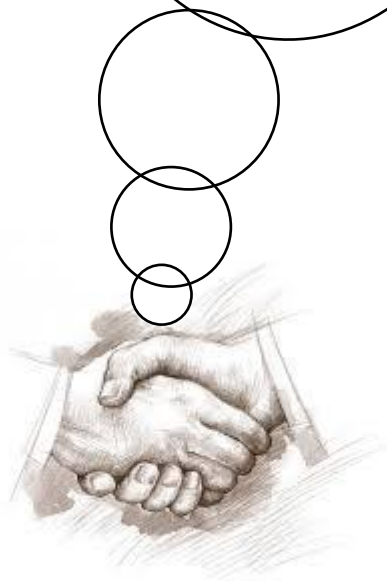
Je dédie ce document

A ma famille pour m'avoir soutenu tout long de ma scolarité et de mes examens.

Remerciements

*Pour l'aboutissement de ce projet, nous avons été emmenés à côtoyer plusieurs personnes qui n'ont pas hésité à nous aider du mieux qu'elles pouvaient et avec une volonté appréciable. A cet effet nous tenons à remercier notre expert métier M. **Koubedion SOME**, notre Maitre de stage M. **Thierry P. SEDGO** et notre directeur de mémoire M. **Olivier Y. TINDANO** qui n'ont ménagé aucun effort pour la réussite de ce projet ;*

Nous ne manquerons pas de remercier nos collègues pour les éclaircissements apportés au moment opportun Et également à toutes celles et ceux qui m'ont soutenu d'une manière ou d'une autre et qui ont contribué à mon épanouissement intellectuel, qu'ils retrouvent ici l'expression de ma profonde gratitude.



Sigles et Abréviations

Sigle ou Abréviation	Signification
SOFIGIB	Société Financière de Garantie Interbancaire du Burkina
ARIZ	Accompagnement du Risque de financement de l'Investissement privé en Zone d'intervention
LP	Licence Professionnelle
BTS	Brevet Technicien Supérieur
DTS	Diplôme de Technicien Supérieur
DUT	Diplôme Universitaire de Technologie
Bac	Baccalauréat
IBAM	Institut Burkinabè des Arts et Métiers
TIC	Technologie de l'Information et de Communication
CU	Cas d'Utilisation
DC	Diagramme de Classe
DS	Diagramme de Séquence
PHP	Personal Home Page
SGBD	Système de Gestion de Base de Données
UML	Unified Modeling Language
UP	Unified Process
2TUP	2Track Unified Process
RUP	Rational Unified Process
XP	eXtreme Programming
IHM	Interface Homme Machine
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
DAO	Data Access Object
ORM	Object Relation Mapping
SQL	Structured Query Language

MVC	Model View Controller
UI	User Interface
BLL	Business Logic Layer
DAL	Data Access Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
COCOMO	Constructive Cost Model
KLSL	Kilo-Lignes-Sources Livrés
TDEV	Temps de Développement
CTD	Cout Total de Développement
HM	Homme-Mois

Glossaire

Garantie : est une couverture en risque qui couvre une partie de la perte finale de la banque (ou d'un organisme de financement) en cas de défaillance de l'emprunteur, moyennant une commission payée directement par la banque ou par l'emprunteur.

Garant : est le fait qu'une personne se porte garant (particulier, banque, organisme), auprès d'un créancier (banque, propriétaire) en cas de défaillance du débiteur.

Crédit : est une avance de somme d'argent.

Prêt : est le fait pour un établissement de crédit, de proposer une solution de financement, en mettant à disposition des fonds à un bénéficiaire, sans en exiger le remboursement immédiat. Plusieurs caractéristiques sont à prendre en compte dans un prêt bancaire : la somme empruntée, la durée du prêt, le taux d'emprunt et les éventuels frais.

Engagement : désigne toute créance résultant des prêts et avances accordés par l'établissement bancaire, ainsi que toute obligation de l'établissement bancaire en vertu d'un contrat ou tout autre mécanisme, de fournir des fonds à une autre partie (engagement de financement) ou de garantir à un tiers l'issue d'une opération en se substituant à son client s'il n'honore pas ses obligations (engagement de garantie).

Echéance : une échéance ou un échéancier de paiement correspond à la date ou au montant de la mensualité de remboursement d'une somme d'argent. Cette échéance peut être journalière, hebdomadaire, mensuelle ou trimestrielle. Pour un crédit, la somme due correspond au capital et aux intérêts.

Créancier chirographaire : est un créancier simple, c'est-à-dire ne disposant d'aucune garantie particulière (privilège, nantissement, hypothèque) lui permettant d'être payé avant les autres créanciers sur le prix de vente des biens de son débiteur.¹

Sûreté : absence de risque ou de danger. C'est un moyen juridique permettant de garantir les créanciers contre le risque d'insolvabilité du débiteur.²

Compte courant : le compte courant est le dispositif bancaire dont se sert le client d'une banque pour ses achats, ses retraits et ses dépôts quotidiens.³

¹ <https://www.saint-lo.fr/justice?audience=pro&xml=F22354>

² <https://www.observatoiredelafranchise.fr/dossier-franchise/suretes-cautions-garanties-nantissements-gages-545.htm>,

³ <https://www.journaldunet.fr/patrimoine/guide-des-finances-personnelles/1201671-le-compte-courant/>

Titre de propriété : constate un droit de propriété. En immobilier, il s'agit d'une copie de l'acte de vente d'un bien (appartement, maison ou terrain), signée par le notaire, sur lequel l'administration fiscale appose ses cachets.⁴

⁴<https://www.journaledunet.fr/patrimoine/guide-de-l-immobilier/1202139-titre-de-propriete-comment-l-obtenir/>

Résumé

Ce travail couvre l'analyse du futur système sur la base de l'existant, la modélisation du nouveau système pour en avoir une vue représentative et compréhensible, et enfin son implémentation. A l'issue de ce travail, nous avons réalisé et mis en place un système de gestion efficace des garanties bancaires dans le contexte de l'institution financière Société Générale Burkina Faso. Cette gestion inclut également le suivi efficace des échéances de ces garanties et engagements du client garant.

Mots clés : analyse, conception, implémentation, finance, client, garant, garanties, prêts, engagements, échéances.

Summary

This work covers the analysis of the future system on the basis of the existing one, the modelling of the new system to have a representative and comprehensible view of it, and finally its implementation. As a result of this work, we have developed and implemented an efficient bank guarantee management system in the context of the financial institution Société Générale Burkina Faso. This management also includes the effective monitoring of the expiry dates of these guarantees and the commitments of the guaranteeing client.

Keywords : analysis, design, implementation, finance, client, guarantor, guarantees, loans, commitments, deadlines.

Liste des Figures

Figure 1 : Sopra Banking Amplitude	13
Figure 2: Microsoft Excel	14
Figure 3 : La méthode Scrum	26
Figure 4 : Exemple de « user stories »	27
Figure 5 : Exemple de « Back Log du sprint »	27
Figure 6 : Exemple de « burn down chart »	27
Figure 7 : Diagramme Gantt de notre projet	29
Figure 8 : Architecture réseau	34
Figure 9 : « Architecture 3 tiers »	35
Figure 10 : Comparaison entre Laravel et Symfony	37
Figure 11 : Comparaison entre Laravel et Symfony, répartition par région	37
Figure 12 : Logo de Laravel	38
Figure 13 : Logo d'Angular	40
Figure 14 : Logo HTML5	41
Figure 15 : Logo CSS3	42
Figure 16 : Logo de Bootstrap	43
Figure 17 : Logo Visual Studio Code	43
Figure 18 : Logo de Visio	45
Figure 19 : Diagramme de cas d'utilisation principal ou de base	59
Figure 20 : Diagramme de cas d'utilisation : pour le Chargé Garanties	60
Figure 21 : Diagramme de cas d'utilisation : pour le Responsable Garanties	61
Figure 22 : Diagramme de cas d'utilisation : pour le Conseiller Clientèle	62
Figure 23 : Diagramme de cas d'utilisation : pour l'Administrateur	63
Figure 24 : Diagramme d'activité : Gestion des garanties	66
Figure 25 : Diagramme de séquence du scénario « Authentification »	68
Figure 26 : Diagramme de séquence du scénario : « Gestion garanties »	69
Figure 27 : Le formalisme de la classe	71
Figure 28 : Le formalisme de la navigabilité de l'association	71

Figure 29 : Le formalise de l'agrégation	72
Figure 30 : Le formalisme de la composition.....	73
Figure 31 : Le formalisme de la « classe-association »	73
Figure 32 : Diagramme de classes	74
Figure 33 : Diagramme de déploiement.....	77
Figure 34 : Page d'authentification	90
Figure 35 : Page d'accueil	91
Figure 36 : Liste des garanties saisies	92
Figure 37 : Formulaire de saisie d'une garantie	94
Figure 38 : Diagramme de Gantt du planning réel du projet.....	97

Liste des Tableaux

Tableau 1 : Tableau comparatif des différents cycles de vie.....	23
Tableau 2 : Planification du projet.....	28
Tableau 3 : Fiche technique de Laravel.....	38
Tableau 4 : Etude comparative des environnements de développement.....	45
Tableau 5 : Etude comparative des SGBD	46
Tableau 6 : Identification des acteurs	50
Tableau 7 : Cas d'utilisation « S'authentifier ».....	51
Tableau 8 : Cas d'utilisation « Saisir une garantie ».....	52
Tableau 9 : Cas d'utilisation « Mettre à jour une garantie »	53
Tableau 10 : Cas d'utilisation « Consulter une garantie ».....	54
Tableau 11 : Cas d'utilisation « Valider une garantie »	55
Tableau 12 : Cas d'utilisation « Lier une garantie à un engagement ».....	56
Tableau 13 : Cas d'utilisation « Echoir une garantie ».....	57
Tableau 14 : Dictionnaire de données	76
Tableau 15 : Etude comparative des méthodes d'estimation	85
Tableau 16 : Méthode de calcul COCOMO II	86
Tableau 17 : Evaluation du coût du matériels.....	87
Tableau 18 : Coût de développement de l'application	87
Tableau 19 : Coût de formation des utilisateurs.....	88
Tableau 20 : Coût de total de mise en œuvre	88

Préambule

I. Présentation de l'ESTA

L'Ecole Supérieure des Techniques Avancées (ESTA) a été créée en 2003. Elle est agréée par l'Etat du Burkina Faso par l'arrêté ministériel portant autorisation d'ouverture 2004-125/MESSRS/SG/CNESSP/SP du 5 juillet 2004. Située à l'ouest de la mairie de Nongr-Massom à 400 m de l'hôtel Sopatel Silmandé et depuis 2011, elle dispose également d'une annexe située à la zone 1 sur la circulaire en face de la station Shell. Son adresse est la suivante :

- Adresse postale : 09 BP 384 OUAGADOUGOU 09 BURKINA FASO
- Téléphone : Siège : 25 35 56 87 / 25 50 80 48 / 70 21 61 48 / 63 16 88 90
Annexe : 25 50 80 49 / 72 08 61 86 / 25 36 54 77
- Site Web : www.estabf.com
- E-mail : info@esta.bf

II. Formations de l'ESTA

Parmi les valeurs de l'ESTA on peut citer :

- la qualité et la rigueur de l'encadrement pédagogique ;
- un partenariat diversifié et actif au niveau académique, avec des entreprises et des institutions ;
- la qualité et le volume du matériel didactique.

Dans le souci de mettre à la disposition des entreprises nationales et sous-régionales des techniciens supérieurs et ingénieurs qualifiés, immédiatement opérationnels et répondant aux profils recherchés dans les divers domaines d'activités, l'ESTA offre depuis sa création, une formation professionnelle de haut niveau en cours du jour et soir dans les filières ci-dessous.

1. Formations diplômantes

🇵🇷 BTS d'Etat et DTS (Bac +2) :

- Réseaux Informatiques et Télécommunications (RIT) ;

- Génie Electrique : Option Electronique et Informatique Industrielle (GEII) ;
- Systèmes d'information et Réseaux (SIR) ;
- Finance – Comptabilité (FC) ;
- Marketing et Gestion Commerciale (MGC) ;

✚ Les DTS délivrés par l'ESTA sont reconnus par le CAMES dans les filières suivantes :

- Réseaux Informatiques et Télécommunications ;
- Génie Electronique et Informatique Industrielle ;
- Finances – Comptabilité ;
- Informatique de Gestion ;
- Systèmes d'information et Réseaux (SIR).

✚ DUT Français, en partenariat avec l'Université du Littoral (Bac + 2) :

- Génie Electrique et Informatique Industrielle.

✚ L'ESTA assure également des formations post BTS en Licence Professionnelle dans les filières suivantes :

- Réseaux Informatiques et Télécommunications (RIT) ;
- Electronique et Automatismes Industriels (EAI) ;
- Génie Electrique : Option Electromécanique (ELM) ;
- Génie Biomédical (GBM) ;
- Systèmes d'Information et Réseaux (SIR) ;
- Finances & Audit comptables (FAC), en partenariat avec l'IBAM.

2. Masters

L'ESTA assure également des formations en master dans les filières suivantes :

- Comptabilité Contrôle Audit (CCA) ;
- Réseaux Informatiques et Télécommunications (RIT) ;
- Electronique et Automatismes Industriels (EAI) ;
- Electromécanique (ELM) ;
- Génie Electrique : Option Electromécanique (ELM) ;
- Génie Biomédical (GBM) ;

- Systèmes d'Information et Réseaux (SIR) ;
- Sécurité des Systèmes d'Informations (SSI).

3. Formation Continue Professionnelle

L'ESTA organise également des séminaires de formation à l'intention des entreprises pour le renforcement des capacités de leur personnel. Ces formations contribuent énormément à l'actualisation des méthodes et des contenus des formations diplômantes.

4. Certifications en TIC de l'ESTA

En sa qualité d'académie, l'ESTA assure des formations préparant aux tests de certification TIC sur les produits des fournisseurs suivants :

- Cisco ;
- Oracle ;
- Microsoft.

L'ESTA est par ailleurs un centre de test agréé Pearson VUE.

Pour toutes les formations diplômantes, l'étudiant doit effectuer un stage en entreprise dont la durée est fonction du niveau de formation. C'est dans ce cadre que nous avons effectué ce stage à la Société Générale Burkina Faso (**SGBF**).

Introduction Générale

I. Introduction du sujet

La collecte de l'épargne, la fourniture de moyens de paiements et l'octroi de prêts sont les missions principales d'une banque. Ainsi, dans l'exercice de ces activités, la banque est amenée à octroyer des prêts à ses clients. Mais, elle est contrainte par le risque de crédit qui se pose entre elle et les clients emprunteurs.

Ce risque de crédit se définit par le fait que n'existe pas de parfaite certitude que l'autre partie arrive à honorer ses engagements. Que faire alors pour encadrer ce risque et limiter les pertes pour la banque au cas où elles surviennent ?

II. Contexte et justification (ou problématique)

Ces dernières décennies ont vu une remise en question des habitudes bancaires conventionnelles, tout particulièrement lorsqu'il est apparu évident qu'il y avait des échecs dans la fourniture des services financiers. Ces services financiers sont en grandes parties des prêts bancaires qui conditionnent lors de leur mise en place des garanties dont les clients doivent disposer obligatoirement. Cette obligation occasionne la création de plusieurs types de garanties afin d'offrir plusieurs possibilités de garanties aux clients désirant un crédit. Ainsi, les nombreux engagements (prêts) contractés par ces clients génèrent en retour un nombre important de garanties. La gestion et le suivi de ces dernières s'avère difficile voire impossible au regard de la volumétrie de données manipulées. Existe-t-il alors un système efficace pour gérer un tel processus afin de répondre au cadre juridique ? Notre étude vise à proposer une solution idoine à cette problématique posée.

III. Objectifs

1. Objectif général

L'objectif général se résume à la mise en place d'une solution de gestion efficace des garanties bancaires.

2. Objectifs spécifiques

Ils se déclinent comme suit :

- analyser le système afin de mieux le comprendre ;
- concevoir le futur système à mettre en place ;
- implémenter, pour finir, une solution de gestion efficace des garanties bancaires.

IV. Hypothèses (?)

1. Hypothèse Principale

La garantie du client ne serait pas échue.

2. Hypothèses Spécifiques

Nous pouvons en ressortir deux (02) hypothèses spécifiques à savoir :

- La garantie du client recouvrerait ses engagements ;
- L'engagement du client serait arrivé à terme ;

V. Questions de recherche

Pour mener à bien notre étude, nous devons être à mesure de donner des réponses aux questions suivantes :

- Est-elle efficace la gestion des garanties ?
- Est-elle échue une garantie donnée ?
- Recouvre-t-elle, la garantie du client à ses engagements ?
- Est-il arrivé à terme l'engagement du client ?

VI. Intérêt de l'étude

Le résultat de notre travail sera :

- une source d'informations et d'inspiration pour les futures générations de notre école ;
- une nouvelle solution pour une gestion plus efficace et moderne des garanties et des engagements des clients dans un respect du cadre juridique ;
- un approfondissement de notre culture générale du domaine financier que nous découvrons ;

- une maîtrise technique supplémentaire des nouvelles technologies de dernières générations que nous aurons à utiliser dans la mise en place de la solution.

VII. Méthodologie de collecte des données

Afin de valider nos hypothèses, nous avons fait recours à des techniques d'étude pour collecter les données :

1. Interview

Les interviews sont nos premières sources de collecte de données. Nous les avons menés au préalable, ce qui nous a permis de collecter des informations sur l'état de gestion actuelle des garanties. Ils ont été réalisés auprès d'experts métiers du périmètre défini sur la base de fiches d'interview spécialisées selon le service et la fonction de l'expert métier. Ces fiches sont après l'interview amendés, signées et scannées par l'utilisateur interviewé.

2. Documents

En plus des interviews réalisés, nous avons eu recours aux procédures actuelles régulant la gestion des garanties.

3. Recherche bibliographique

Nous avons fait recours à des références bibliographiques et webographies traitant du même domaine pour enrichir notre compréhension du sujet.

4. Traitements des résultats

Les données collectées ont été analysées et traitées selon la méthode d'analyse et de conception que nous vous présenterons à la suite de ce document.

VIII. Structure d'accueil

1. Présentation

Société Générale Burkina Faso est une filiale d'un acteur de premier plan des services financiers en France, en Europe et dans le monde : le groupe Société Générale. S'appuyant sur un modèle diversifié de banque universelle, Société Générale Burkina Faso allie solidité financière et stratégie de croissance durable afin de mettre sa performance au service du financement de l'économie et des projets de ses clients. Elle a été créée en mai 1998 avec la participation de l'Etat Burkinabé et de plusieurs partenaires financiers nationaux et internationaux.

Société Générale Burkina Faso a pour ambitions de :

- bâtir avec ses clients une relation équilibrée, et équitable où elle est avec eux, à leurs côtés, pour les aider à progresser ;
- mettre sa performance au service de ses clients ;
- être la banque relationnelle, référence sur ses marchés, choisie pour la qualité et l'engagement de ses équipes.

2. Historique

De la Caisse Nationale des Dépôts et des Investissements (CNDI) à Société Générale Burkina Faso, découvrez l'histoire de votre partenaire au quotidien.

Septembre 1973 : Création de la Caisse Nationale des Dépôts et des Investissements (CNDI).

Août 1984 : Création de l'Union Révolutionnaire de Banques (UREBA).

Juin 1986 : Création de la Caisse Autonome d'Investissements (CAI).

Août 1986 : Transformation de la CNDI en banque commerciale sous la forme d'une société d'économie mixte.

Décembre 1987 : Changement de dénomination de la Caisse Nationale des Dépôts et des Investissements (CNDI) en Banque pour le Financement du Commerce et des Investissements au Burkina (BFCI-B).

Février 1991 à Décembre 1996 : Mise sous administration provisoire du Groupe BFCI-B/UREBA /CAI. Fusion-absorption de l'UREBA et de la CAI par la BFCI-B en mai 1995.

Février 1997 : Cession par l'Etat de 34% du capital à des privés nationaux.

Mai 1998 : Cession par l'Etat de 51% du capital à des partenaires étrangers. La BFCI-B devient la Société Générale de Banques au Burkina (SGBB).

14 Janvier 2012 : Déploiement de la nouvelle signature du groupe Société Générale : « Développons ensemble l'esprit d'équipe ».

08 février 2013 : Changement de dénomination sociale : Société Générale de Banques au Burkina devient Société Générale Burkina Faso.

3. Les valeurs du groupe

Rejoignez une banque où la transversalité des carrières, la cohésion managériale, le développement des compétences de ses collaborateurs sont une vision.

L'engagement de ses collaborateurs est en effet crucial pour la satisfaction de ses clients et la construction de la banque relationnelle de référence.

L'esprit d'équipe, sa première valeur, va au-delà de sa façon de travailler « **Ensemble** ». Il s'agit de créer des relations durables, basées sur le respect et la confiance tant en interne entre les collègues qu'en externe avec ses clients. « Développons ensemble l'esprit d'équipe », c'est le plaisir d'être ensemble, de s'entraider et d'apprendre des autres. La relation tissée avec les collaborateurs en est le fondement.

Ses collaborateurs nourrissent ensemble un environnement alliant une expertise approfondie et une volonté de toujours anticiper les besoins de ses clients ainsi que l'évolution des marchés.

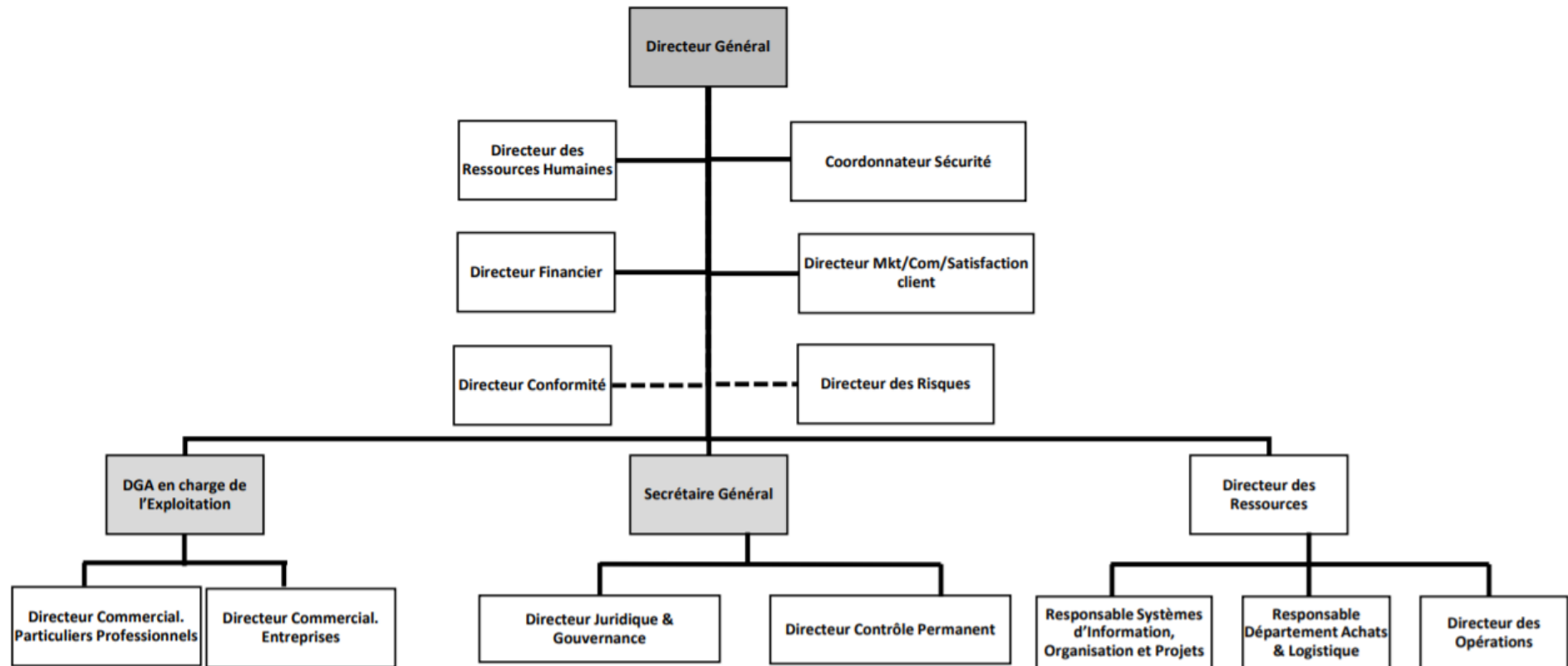
L'esprit d'équipe, l'innovation, la responsabilité et l'engagement sont les valeurs que partagent tous les collaborateurs de Société Générale. Des valeurs au cœur de sa vision de banque responsable et engagée au service de ses clients.

4. Le capital social

Réparti entre le groupe Société Générale, les acteurs financiers nationaux et la FINADEI, le capital social de Société Générale Burkina Faso s'élève à douze milliards huit cents millions de francs CFA (12 800 000 000 FCFA).

5. L'organigramme

L'organigramme de Société Générale Burkina Faso se présente comme suit :



Chapitre I

Revue de littérature

Introduction

Selon **Charles MOUMOUNI (1997)** dans son article de la revue juridique, les garanties bancaires sont nées de la pratique du commerce international à l'aube du XX^e siècle. Le rapport de force sur le marché international s'est progressivement renversé en faveur des importateurs. Ces derniers exigent désormais des vendeurs des sûretés quant à la qualité des prestations qu'ils offrent, en contrepartie de la souscription des crédits documentaires. Dans un premier temps, les exportateurs remettent des titres ou déposent des sommes d'argent pour garantir l'exécution parfaite de leurs obligations. Mais une telle pratique s'est révélée incommode parce qu'elle immobilise des valeurs et assèche la trésorerie des vendeurs, d'où la naissance des garanties bancaires⁵.

I. Les garanties bancaires

Dans la présentation générale des garanties bancaires conventionnelles usuelles du crédit de **Mehdi Kettani (2017)**, les garanties sont définies sous deux formes (la finalité et les aspects juridiques). Selon la finalité, pour le créancier, il s'agit de lui offrir une sécurité accrue, une chance supplémentaire de paiement. Pour paraphraser **Joseph KAMGA (2020)**, le créancier nanti d'une sûreté, titulaire d'une hypothèque, d'un nantissement, d'un gage ou d'un cautionnement est plus confiant⁶. Pour le débiteur, il s'agit d'une prérogative superposée aux prérogatives ordinaires du créancier par un contrat et qui a pour finalité juridique exclusive de le protéger contre l'insolvabilité de son débiteur. Selon les aspects juridiques en complément :

- les garanties permettent de faire échec au principe général de l'égalité des créanciers ;
- elles offrent un régime préférentiel qui sort du droit commun (droit de préférence et droit de suite) ;
- le créancier titulaire d'une sûreté est donc un créancier privilégié : il a plus de chances de se faire payer qu'un créancier chirographaire⁷.

⁵ Charles MOUMOUNI, « Article de la revue juridique Thémis », 1997, page 781

⁶ Joseph KAMGA, « Droit et pratique hypothécaires dans l'espace OHADA », 2020, page 35

⁷ Mehdi Kettani, « Présentation générale des garanties bancaires conventionnelles usuelles du crédit », 2017, page 1.

Les garanties bancaires sont subdivisées en deux grandes catégories : les garanties personnelles et les garanties réelles.⁸ Les garanties personnelles composées principalement :

- du cautionnement, définie d'une part comme : « un contrat par lequel une personne s'oblige envers le créancier à satisfaire à l'obligation du débiteur, si celui-ci n'y satisfait pas » et d'autre part comme : « une sûreté personnelle accessoire créée par un contrat unilatéral qui oblige la caution à exécuter la dette du débiteur principal et lui donne un recours en remboursement contre ce dernier »⁹;
- de la garantie à première demande avec proposition de définition comme étant une convention par laquelle, à la requête ou sur instruction du donneur d'ordre, le garant s'engage à payer une somme déterminée au bénéficiaire, sur première demande de la part de ce dernier. En exemple, dans le cadre d'opérations internationales de crédit, lorsqu'une entreprise originaire d'un pays (A) cherche à emprunter auprès d'une banque d'un pays (B), cette dernière demande souvent à l'entreprise de lui fournir une garantie à première demande émanant d'une banque de premier rang de son pays d'origine (A)¹⁰.

En ce qui concerne les garanties réelles, les plus pratiquées par les banques sont :

- l'hypothèque : elle est définie comme la garantie hypothécaire a toujours le vent en poupe en Afrique subsaharienne francophone. Les auteurs classiques la présentaient comme la « reine des sûretés », celle qui garantissait au créancier la chance d'être payé en cas de débâcle du débiteur¹¹;
- nantissement : selon le droit OHADA en vigueur, on parle de nantissement, lorsque la garantie porte sur les valeurs incorporelles. C'est ainsi qu'on parle de nantissement de fonds de commerce, de compte, de titres de propriété. Par définition, le nantissement est « un contrat par lequel le débiteur, ou un tiers agissant dans son intérêt, affecte une chose mobilière incorporelle à la garantie d'une obligation, et confère au créancier le droit de se payer sur cette chose, par

⁸ Mehdi Kettani, « Présentation générale des garanties bancaires conventionnelles usuelles du crédit », 2017, page 2.

⁹ Mehdi Kettani, « Présentation générale des garanties bancaires conventionnelles usuelles du crédit », 2017, page 2.

¹⁰ Mehdi Kettani, « Présentation générale des garanties bancaires conventionnelles usuelles du crédit », 2017, page 4.

¹¹ Joseph KAMGA, « Droit et pratique hypothécaires dans l'espace OHADA », 2020, page 21.

préférence à tous autres créanciers, au cas où le débiteur manquerait à le satisfaire »¹².

II. Le risque de crédit

Mehdi Kettani (2017), le crédit est le moteur de l'économie dans un système économique moderne¹³.

Pour **Hervé Diaz (2015/2016)** dans sa mémoire de fin d'études, selon RANSO GP, « le risque de contrepartie représente la perte potentielle réalisée par la banque dans l'hypothèse d'une défaillance future de sa contrepartie. Le risque de crédit peut être défini comme la perte totale enregistrée sur une opération suite à la défaillance de la contrepartie. On l'appelle aussi parfois risque de signature. Il est courant d'employer le terme de risque de contrepartie pour désigner exclusivement le risque de crédit »¹⁴.

Le risque de contrepartie génère des impacts bien précis au sein des banques :

- des impacts financiers directs (non restitution du capital prêté, moins-value, détournement de fonds) ;
- des impacts financiers indirects (provision élevée sur les bénéfices, anticipation de perte probable, charges supplémentaires) ;
- des impacts commerciaux (perte de clientèle, dévalorisation de l'image de la banque)¹⁵.

Conclusion

Il paraît nécessaire de trouver les moyens et les techniques afin de gérer ce risque c'est-à-dire mettre en place des outils de gestion pour identifier et évaluer le risque de crédit bancaire. **Les garanties se présentent comme un puissant instrument d'atténuation du risque de crédit**¹⁶. Tout de même, une gestion efficace des garanties bancaires est l'un des moyens importants des institutions financières afin de diminuer ou de limiter le risque de crédit.

¹² Mehdi Kettani, « Présentation générale des garanties bancaires conventionnelles usuelles du crédit », 2017, page 7.

¹³ Mehdi Kettani, « Présentation générale des garanties bancaires conventionnelles usuelles du crédit », 2017, page 1.

¹⁴ Hervé Diaz, « La gestion du risque de crédit bancaire sur les portefeuilles professionnels et particuliers », 2016, page 5.

¹⁵ Hervé Diaz, « La gestion du risque de crédit bancaire sur les portefeuilles professionnels et particuliers », 2016, page 6.

¹⁶ Joseph KAMGA, « Droit et pratique hypothécaires dans l'espace OHADA », 2020, page 39.

Chapitre II

Etude fonctionnelle du système

Introduction

Tout au long de ce chapitre nous ferons une critique sur le système existant, ensuite nous identifierons les fonctionnalités que le futur système devrait contenir, et nous présenterons par la suite la démarche de développement adoptée pour terminer avec la planification du projet que nous avons dénommé « Lafia Garanties ».

I. Critique de l'existant

L'étude du système existant nous a permis de ressortir de nombreuses difficultés dans la gestion des garanties. Ces difficultés ont été révélées suite aux entretiens avec le service « garanties ». La gestion et le suivi sont essentiellement basés sur le module de gestion de garanties du progiciel Amplitude Delta de Sopra Banking et le tableur de la suite Microsoft Office Excel, outils qui au final avaient de nombreuses contraintes

1. Amplitude Delta



¹⁷ Créée depuis 1968 soutient les institutions financières tout au long de leur changement avec les technologies qu'elle met à leur disposition. En 2012, elle décide de regrouper son expertise et son expérience sous une seule marque : Sopra Banking Software. Une filiale à part entière avec un projet autonome et ambitieux. Depuis cette fondation, elle est passée d'un fournisseur majoritairement français à fournisseur mondial¹⁸. Les principales difficultés rencontrées au sein du métiers à sa version V10 sont :

- le suivi général des garanties ;
- le rattachement des garanties aux engagements ;
- le suivi des échéances ;
- l'absence d'alertes.

Aux vues de ces difficultés, une demande d'évolution du progiciel auprès de l'éditeur était nécessaire. Mais cela nécessite de longs délais de mis en œuvre sans oublier le coût exorbitant de la prestation.

¹⁷ <https://www.soprabanking.com/fr/#firstPage>

¹⁸ <https://www.soprabanking.com/about-us/>

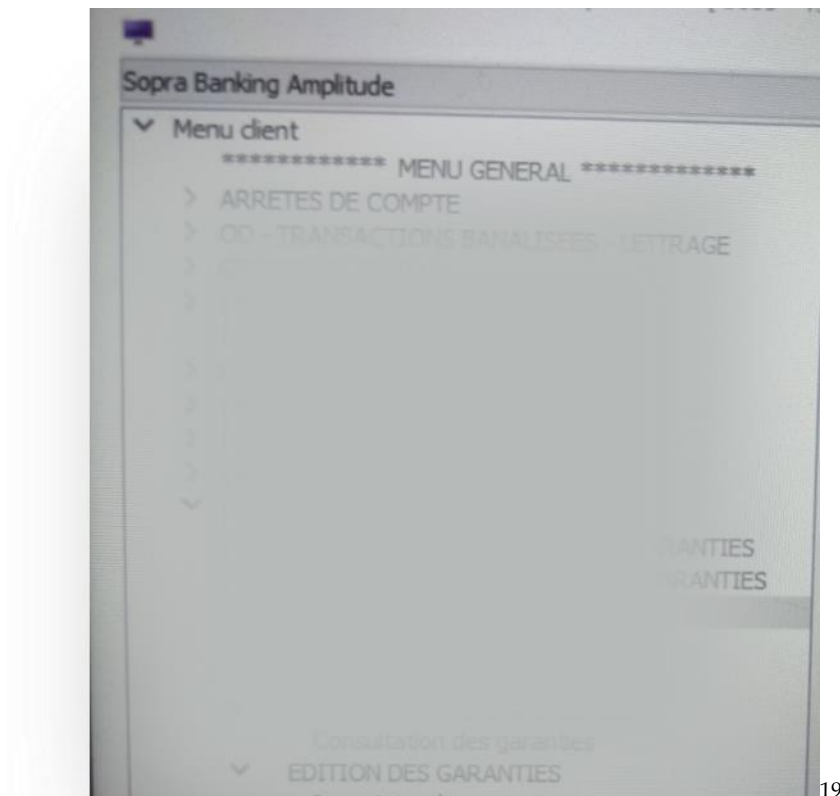


Figure 1 : Sopra Banking Amplitude

2. Microsoft Excel



²⁰A l'origine, Microsoft a commercialisé un tableur appelé « Multiplan » en 1982 qui fut très populaire. La première version d'Excel est sortie en 1985 (Multiplan ne fut pas réutilisé) pour les ordinateurs Apple Macintosh alors que la première version pour Windows, la 2.05 (en ligne avec celle pour Macintosh), est sortie en 1987 ²¹. La dernière version actuellement utilisée est celle de 2016 qui présente presque les mêmes difficultés au sein du service « garanties » que les versions précédentes à savoir :

- difficulté de stockage ou problème d'enregistrement de nouvelles données qui serait lié d'une part à l'accès concurrentiel dû aux limites de Excel et d'autre part à la grande taille du fichier, le tableur contenant un nombre élevé de lignes ;

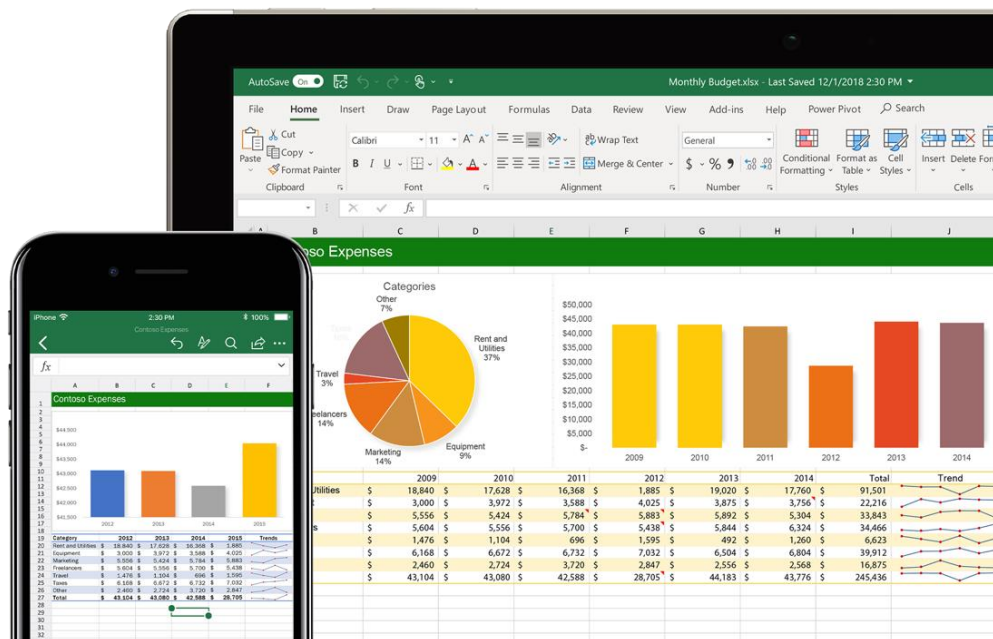
¹⁹ Capture d'écran prise le 13/01/2022

²⁰ <https://www.microsoft.com/fr-ww/microsoft-365/excel>

²¹ <https://www.techno-science.net/glossaire-definition/Microsoft-Excel.html>

- difficulté d'exploitation des données renseignées dans ledit fichier : traitement difficile, expositions à d'énormes risques d'erreurs, production et mise à jour difficile des statistiques conduisant à une lenteur dans le traitement de façon générale.

Ces difficultés rencontrées sur ce tableur font d'Excel un outil non adapté aux traitements de données à grande échelle. La mise en place d'une solution plus efficace devrait rapidement être envisagée.



22

Figure 2: Microsoft Excel

II. Les fonctionnalités identifiées

Tout au long de ce point, nous nous étalerons sur les modules et fonctionnalités identifiés pendant nos entretiens avec le métier qui est au cœur de la gestion de garanties. Ces entretiens nous ont permis de retenir principalement cinq (5) modules :

- le module de gestion des garanties ;
- le module de gestion des assurances ;
- le module de production de statistiques ;
- le module de gestion de la sécurité
- et le module de paramétrage.

1. Le module de gestion des garanties

Les garanties sont réparties en plusieurs types et chaque type en plusieurs natures. On dénombre au sein de notre structure d'accueil principalement huit (8) types de garanties :

- les hypothèques (hypothèque ou privilège sur maison,
- les hypothèques sur immeuble industriel),
- les nantissements,
- les gages,
- les garanties institutionnelles,
- les actes de cautionnement solidaire, l
- es garanties ARIZ,
- les garanties SOFIGIB,
- les garanties fudiciaire).

Cette gestion doit permettre de :

- saisir une garantie ;
- archiver des fiches d'une garantie ;
- éditer une fiche d'une garantie ;
- éditer un certificat de levée de garantie ;
- joindre certificat signé ;
- mettre à jour une garantie ;
- valider une garantie ;
- annuler une garantie ;

- lever une garantie ;
- lier une garantie à un ou des engagements ;
- lever un lien entre une garantie et engagements ;
- voir les détails d'une garantie ;
- générer des alertes à chaque étape et avant chaque échéance d'une garantie ;
- renouveler une garantie : si elle arrive presque à échéance et déclenchée par une alerte ;
- réévaluer une garantie ;
- enregistrer les entrées et sorties des titres de propriété²³ des hypothèques immobilières.

2. Le module de gestion des assurances

Le module des assurances est un module complémentaire à la gestion des garanties. Il permet de gérer surtout les assurances incendies liées aux hypothèques immobilières qui nécessitent obligatoirement d'être assurées. On dénombre également deux types d'assurances gérées au sein du service « garanties » :

- les assurances incendies ;
- les assurances vies.

La gestion des assurances doit permettre de :

- saisir une assurance et de lier également à l'hypothèque concernée si assurance incendie ;
- mettre à jour une assurance ;
- valider une assurance ;
- annuler une assurance ;
- voir les détails d'une assurance ;
- et générer des alertes à chaque étape et avant l'échéance d'une assurance incendie.

²³ Un titre de propriété constate un droit de propriété. En immobilier, il s'agit d'une copie de l'acte de vente d'un bien (appartement, maison ou terrain), signée par le notaire, sur lequel l'administration fiscale appose ses cachets.

3. Le module de production de statistiques

Ce module doit permettre de :

- produire l'état des garanties ;
- produire l'état sur des garanties avec engagements soldés ;
- produire l'état des mouvements des titres ;
- produire l'état sur les renouvellements des hypothèques ;
- produire l'état sur les réévaluations des garanties (hypothèques, nantissement, garanties institutionnelles) ;
- produire l'état des assurances.

4. Le module de gestion de la sécurité

4.1. La gestion des utilisateurs

Elle doit permettre :

- de créer un utilisateur ;
- de l'attribuer un profil ou rôle ;
- d'activer un utilisateur dans le système ;
- de désactiver un utilisateur dans le système ;
- de réinitialiser les accès d'un utilisateur au système ;

4.2. La gestion des profils ou des rôles

Elle doit permettre :

- de créer un profil utilisateur ;
- d'attribuer les droits ou habilitations d'un profil ;
- de retirer les droits ou habilitations d'un profil ;

4.3. La gestion de l'historique ou la traçabilité

Elle doit permettre :

- de tracer chaque action d'un utilisateur dans le système ;
- de filtrer les actions d'un utilisateur sur une période donnée.

4.4. La gestion des sauvegardes

Elle doit permettre :

- de créer une sauvegarde des données enregistrées dans le système ;
- d'exécuter une sauvegarde automatiquement et quotidiennement ;
- de télécharger une sauvegarde.

5. Le module de paramétrage

5.1. La gestion des agences

Les garanties sont définies par agence. Elle doit permettre de :

- créer une agence ;
- et de mettre à jour une agence.

5.2. La gestion des services

Elle doit permettre de :

- créer un service des collaborateurs ;
- et de mettre à jour un service.

5.3. La gestion des fonctions

Elle doit permettre de :

- créer une fonction d'un collaborateur ;
- et de mettre à jour une fonction.

5.4. La gestion des collaborateurs

Elle doit permettre de :

- d'enregistrer un collaborateur intervenant dans le système ;
- et de mettre à jour un collaborateur.

5.5. La gestion des institutions garantes

Elle doit permettre de :

- définir une institution garante des garanties institutionnelles ;
- et de mettre à jour une institution garante.

5.6. La gestion des devises

Elle doit permettre de :

- définir la devise des garanties institutionnelles ;
- et de mettre à jour la devises.

5.7. La gestion des types de concours

Un type de concours est le type d'engagement couvert par garantie, généralement un prêt ou le compte courant ²⁴ du client. Elle doit permettre de :

- saisir un type de concours ;
- et de mettre à jour un type de concours.

5.8. La gestion des types de garanties

Elle doit permettre de :

- définir un type de garantie ;
- mettre à jour un type de garantie ;
- et de lier les natures de garanties à chaque type de garantie donné.

5.9. La gestion des natures de garanties

Elle doit permettre de :

- définir une nature de garantie ;
- et de mettre à jour une nature de garantie.

5.10. La gestion des évaluateurs

Elle doit permettre de :

- créer un évaluateur ;
- et de mettre à jour un évaluateur.

5.11. La gestion des notaires

Elle doit permettre de :

- créer un notaire ;
- et de mettre à jour un notaire.

²⁴ Le compte courant est le dispositif bancaire dont se sert le client d'une banque pour ses achats, ses retraits et ses dépôts quotidiens.

5.12. La gestion des originaux titres

Elle doit permettre de :

- saisir l'original d'un titre ;
- et de mettre à jour l'original d'un titre.

5.13. La gestion des localités immeubles

Elle doit permettre de :

- créer une localité immeuble d'une hypothèque ;
- et de mettre à jour une localité immeuble.

5.14. La gestion des types d'assurances

Elle doit permettre de :

- créer un type d'assurance ;
- et de mettre à jour un type d'assurance.

5.15. La gestion des paramètres généraux

Elle doit permettre de :

- définir les paramètres généraux ;
- et de mettre à jour les paramètres généraux.

III. Démarche de développement

La mise en place d'un tel système prend la forme d'un projet. La conduite du projet en « mode projet », est très important pour la bonne mise en œuvre du futur système.

Le processus de développement constitue un facteur déterminant dans la réussite d'un projet, du fait qu'il cadre ses différentes phases et caractérise les principaux traits de sa conduite. Pour cela, le choix d'une méthode de développement, qui soit adéquate aux particularités et exigences d'un projet, doit être élaboré au préalable afin d'obtenir un produit de qualité qui répond aux besoins et aux attentes du métier. Parmi les processus de développement les plus connus on trouve.

1. Tableau comparatif des différents cycles de vie

Cycles de vie	Description	Points forts	Points faibles
Cascade	<ul style="list-style-type: none"> propose de dérouler les phases projet de manière séquentielle. 	<ul style="list-style-type: none"> distingue clairement les phases projet. 	<ul style="list-style-type: none"> non itératif ; ne propose pas de modèles de documents.
2TUP (Two Track Unified Process)	<ul style="list-style-type: none"> s'articule autour de l'architecture ; propose un cycle de développement en Y détaillé. 	<ul style="list-style-type: none"> itératif ; fait une large place à la technologie et à la gestion du risque ; définit les profils des intervenants, les livrables, les plannings, les prototypes. 	<ul style="list-style-type: none"> plutôt superficiel sur les phases situées en amont et en aval du développement : capture des besoins, support, maintenance, gestion du changement.
En V	<ul style="list-style-type: none"> il est représenté par un V dont la branche descendante contient toutes les étapes du projet, et la branche montante toutes les étapes de tests du projet. 	<ul style="list-style-type: none"> facile à utiliser ; les tests sont effectués à chaque étape ; le contrôle se fait progressivement à chaque étape ; les phases de validation sont prises en main très tôt dans le processus de développement. 	<ul style="list-style-type: none"> le processus n'est pas itératif ; une mauvaise prise en compte des changements de la spécification des besoins ; ne contient pas les activités d'analyses de risques.
En Spirale	<ul style="list-style-type: none"> le cycle en spirale reprend les étapes du cycle en V, mais prévoit l'implémentation de n 	<ul style="list-style-type: none"> sans coût élevé, donne des indications sur les risques majeurs ; les fonctions critiques à haut risque sont 	<ul style="list-style-type: none"> le temps consacré à l'évaluation des risques est trop élevé pour des petits projets. le temps mis à planifier, évaluer les risques, fixer les

	versions successives, ce qui permet de mettre l'accent sur la gestion des risques.	développées en premier lieu ; <ul style="list-style-type: none"> la conception ne doit pas forcément être terminée ; les utilisateurs ont dès le départ une vue globale du système. 	objectifs, le prototype peut être excessif. <ul style="list-style-type: none"> ce modèle est complexe. Une expertise en évaluation des risques est nécessaire.
Scrum	<ul style="list-style-type: none"> un schéma d'organisation de développement de produits complexes ; il est défini par ses créateurs comme un « cadre de travail permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible ». 	<ul style="list-style-type: none"> Scrum est centrée sur le produit. le progrès est incrémental, facilement mesurable et clairement visible pour les acteurs économiques ; les développeurs fixeront le rythme, ne sont pas surchargés de travail, et bénéficient d'un rôle accru ; la charge du travail est réglable, basé sur la capacité de l'équipe et la 	<ul style="list-style-type: none"> projet Scrum sans Product owner actif et sans acteurs économiques engagés sera probablement fiasco ; si le propriétaire du produit n'a pas apprécié Scrum, ou ne comprend pas le rôle du propriétaire de produit, le succès est plus difficile.

		<p>priorité des tâches à réaliser ;</p> <ul style="list-style-type: none"> • les questions sont généralement découvertes avant qu'ils ne deviennent endémiques ; • l'équipe a l'autorité de prendre des décisions ; • l'équipe est encouragée à « consulter » avec les entreprises ; • les tâches ont tendance à être granulaire, et par conséquent, plus facilement testables. 	
--	--	---	--

Tableau 1 : Tableau comparatif des différents cycles de vie²⁵

Après avoir étudié la nature de notre projet, nous avons opté pour une méthode agile et plus précisément la méthode Scrum. En effet les spécifications du projet n'ont pas cessé d'être modifiées tout au long de sa mise en place et les outils et technologies n'ont pas été explicitées depuis le départ. Scrum permet d'entamer le projet avec un

²⁵ Mlle. Joairia Lafhal ; Mlle. Soumaya Moussati, « Réalisation et développement d'une plateforme pour la gestion et le suivi des projets des entreprises », 2016, page 33.

minimum d'informations sur les travaux à faire et de procéder à des ajustements au fil de la mise en œuvre du projet.

2. Méthode de Scrum

Les méthodes agiles sont des méthodologies essentiellement dédiées à la gestion de projets informatiques. Elles reposent sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client. Elles permettent notamment d'impliquer l'ensemble des collaborateurs ainsi que le client dans le développement du projet.

3. Approche Agile

Depuis une quinzaine d'années, la majorité des développements des logiciels s'appuie sur des méthodes dites "agiles". Sous cette bannière se regroupent plusieurs méthodes basées sur un développement itératif et incrémental, dans lequel la recherche de solutions aux problèmes rencontrés s'appuie sur la collaboration de pair à pair. Elle promeut des réponses rapides et flexibles, une planification des tâches adaptatives dans des laps de temps très courts permettant une très grande réactivité.

Les approches plus classiques tels que cycle en V ou le modèle en cascade sont souvent mises en œuvre pour les projets répondant à un imposant cahier des charges. Le client et le prestataire s'entendent alors sur un contrat et si le projet prend du retard ou ne répond pas à tous les besoins à la date butoir, des pénalités sont alors facturées au prestataire. De la même manière si le client s'aperçoit en cours de route que certains besoins ont été omis dans le cahier des charges, il devra alors renégocier avec le prestataire le contrat.

Les premières méthodes agiles apparues sont EVO (Evolutionary Project Management) (1976), RAD (développement rapide d'applications) (1991), puis DSDM, la version anglaise du RAD (1995). Les trois méthodes agiles désormais les plus utilisées sont : la méthode Kanban, issue de la méthode industrielle Lean, la méthode Scrum publiée en 2001 par Ken Schwaber et Jeff Sutherland, et la méthode XP (eXtreme Programming) publiée en 1999 par Kent Beck.

4. Pourquoi la méthode Scrum ?

Scrum est de très loin la méthodologie la plus utilisée parmi les méthodes Agile existantes. Elle est donc la plus éprouvée, documentée et supportée. Livres, blogs, formations, vidéos, associations, conférences traitant de Scrum ne manquent pas et bon nombre de ces ressources sont accessibles gratuitement. On pourrait pratiquement parler d'un standard Agile. Un autre atout important : Scrum est simple à comprendre.

Les experts de Scrum, même ses fondateurs, le décrivent comme un « cadre de travail permettant de répondre à des problèmes complexes et changeants tout en livrant de manière productive et créative des produits de la plus grande valeur possible ». Scrum propose un modèle de contrôle de processus basé sur l'empirisme.

Il s'appuie sur les trois piliers suivants :

- **la transparence** : les aspects importants du processus doivent être visibles à ceux qui sont responsables des retombées. La transparence requiert la définition d'un standard commun pour ces aspects afin que les observateurs partagent une compréhension commune de ce qui est observé ;
- **l'inspection** : les utilisateurs de Scrum doivent fréquemment inspecter les artéfacts Scrum et l'état d'avancement par rapport à un objectif de Sprint (Sprint Goal) afin de détecter les écarts indésirables ;
- **l'adaptation** : si un inspecteur détermine qu'un ou plusieurs aspects du processus dérivent hors des limites acceptables, et que le produit qui en résulte sera inacceptable, le processus ou le matériel utilisé par le processus doit être ajusté. Un ajustement doit être fait dès que possible afin de minimiser le risque d'autres dérives.

5. Fonctionnement de Scrum

La méthode Scrum s'appuie sur le découpage des projets en itérations encore nommées « **sprints** ». Un sprint peut avoir une durée qui varie généralement entre deux semaines et un mois.

Avant chaque sprint, les tâches sont estimées en temps et en complexité à l'aide de certaines pratiques comme le « **planning poker** », une manière ludique de chiffrer la complexité des tâches ou évolutions à l'aide de cartes à l'instar du célèbre jeu dont le nom est repris. Ces estimations permettent à la fois de planifier les livraisons, mais

aussi d'estimer le coût de ces tâches auprès du client. Les fonctionnalités (encore appelées « **user stories** ») (voir figure 4) qui font l'objet d'un sprint constituent ce que l'on appelle un « **sprint back log** » (voir figure 5) du produit éventuellement livrable à la fin du sprint. Il est nécessaire de distinguer le sprint back log du « Product back log » qui lui correspond à l'ensemble des fonctionnalités attendues pour le produit sur l'ensemble des sprints.

La méthode Scrum est aussi caractérisée par une « mêlée » quotidienne, encore appelée « **morning** » ou « **stand-up** » ou « **daily scrum** », dans laquelle les collaborateurs (chefs de projets, développeurs et responsables fonctionnels) indiquent tour à tour les tâches qu'ils ont effectuées la veille, les difficultés rencontrées et enfin ce sur quoi ils vont poursuivre leur travail le jour suivant. Cela permet d'évaluer l'avancement du projet, de mobiliser des ressources là où cela est le plus nécessaire, mais aussi de venir en aide aux collaborateurs rencontrant des difficultés lorsque celles-ci ont déjà été rencontrées auparavant par d'autres membres de l'équipe. Pour visualiser l'état d'avancement du sprint, on peut créer un « **burn down chart** » (voir figure 6), un graphique permettant de suivre le « reste à faire » (RAF).

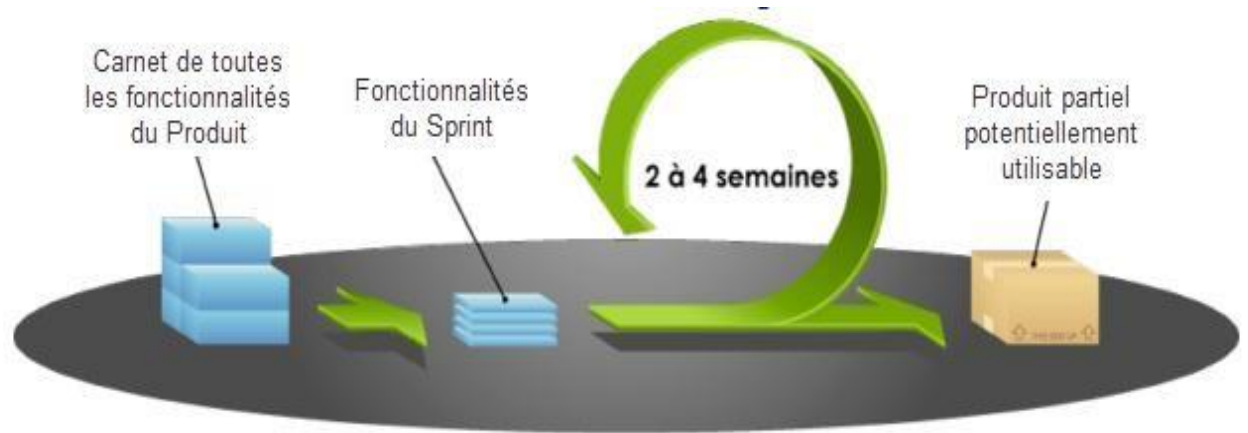
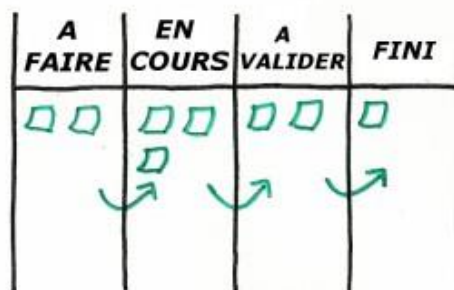
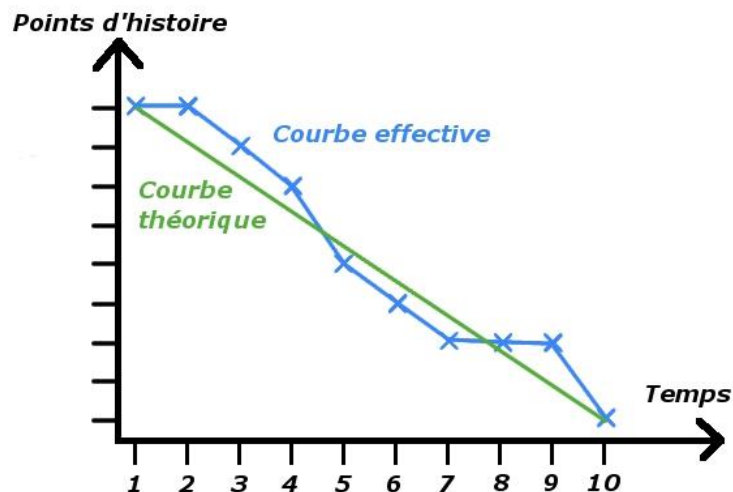


Figure 3 : La méthode Scrum²⁶

²⁶ <https://www.newave.be/blog/2017/02/apprendre-facilement-a-utiliser-la-methodologie-scrum/>

ToDo List			
ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		30	

Figure 4 : Exemple de « user stories » ²⁷Figure 5 : Exemple de « Back Log du sprint » ²⁸Figure 6 : Exemple de « burn down chart » ²⁹

²⁷ <https://www.newave.be/blog/2017/02/apprendre-facilement-a-utiliser-la-methodologie-scrum/>

²⁸ <https://www.newave.be/blog/2017/02/apprendre-facilement-a-utiliser-la-methodologie-scrum/>

²⁹ <https://www.newave.be/blog/2017/02/apprendre-facilement-a-utiliser-la-methodologie-scrum/>

IV. Planification du projet

Cette partie a pour objectif de présenter la manière avec laquelle le projet a été géré. Nous allons présenter dans un premier temps comment le projet a été découpé en tâches afin d'assurer son bon déroulement.

1. Planification

La planification du projet est une étape importante d'avant-projet. Elle consiste à prévoir le déroulement du projet tout au long des phases constituant le cycle de développement. Le tableau ci-dessous montre le planning de notre projet. Réalisé au stade de l'élaboration du cahier des charges, nous avons essayé de bien le respecter, même si parfois des contraintes nous ont obligées à retarder la réalisation de certaines tâches :

Activité	Début	Fin	Jours
Rédaction du mémoire	05/02/21	06/07/21	108
Etude préalable	05/02/21	06/02/21	2
Cadre général du projet	05/02/21	05/02/21	1
Etude de l'existant	06/02/21	06/02/21	1
Etude de projet	07/02/21	19/02/21	12
Etude fonctionnelle	07/02/21	11/02/21	5
Rédaction du cahier de charges	12/02/21	13/02/21	2
Etude technique	14/02/21	18/02/21	5
Conception	19/02/21	06/03/21	15
Diagramme de cas d'utilisation	19/02/21	23/02/21	5
Diagramme de classes	24/02/21	27/02/21	3
Diagramme de séquences	28/02/21	04/03/21	5
Diagramme d'activité	05/03/21	06/03/21	2
Réalisation	07/03/21	06/07/21	87
Sprint 1 : Module de paramétrage	07/03/21	18/03/21	10
Sprint 2 : Module de gestion des garanties	19/03/21	07/04/21	20
Récupération du sprint 1	19/03/21	08/04/21	1
Développement : Garanties + tests unitaires	08/04/21	27/04/21	19
Sprint 3 : Module de gestion des assurances	28/04/21	12/05/21	15
Récupération du sprint 2	03/05/21	13/05/21	1
Développement : Assurances + tests unitaires	13/05/21	22/05/21	9
Sprint 4 : Module de gestion de la sécurité	23/05/21	01/06/21	10
Récupération du sprint 3	01/06/21	02/06/21	1
Développement : Sécurité + tests unitaires	02/06/21	11/06/21	9
Sprint 5 : Module de production des statistiques	12/06/21	21/06/21	10
Récupération du sprint 4	12/06/21	22/06/21	1
Statistiques	22/06/21	01/07/21	9
Tests fonctionnels	01/07/21	06/07/21	5

Tableau 2 : Planification du projet

2. Diagramme de Gantt

Diagramme de Gantt est un outil permettant de modéliser la planification de tâches nécessaires à la réalisation du projet, il visualise dans le temps les diverses tâches liées composant le projet en fournissant une représentation graphique de son avancement. La figure ci-dessous montre le diagramme de Gantt de notre projet.

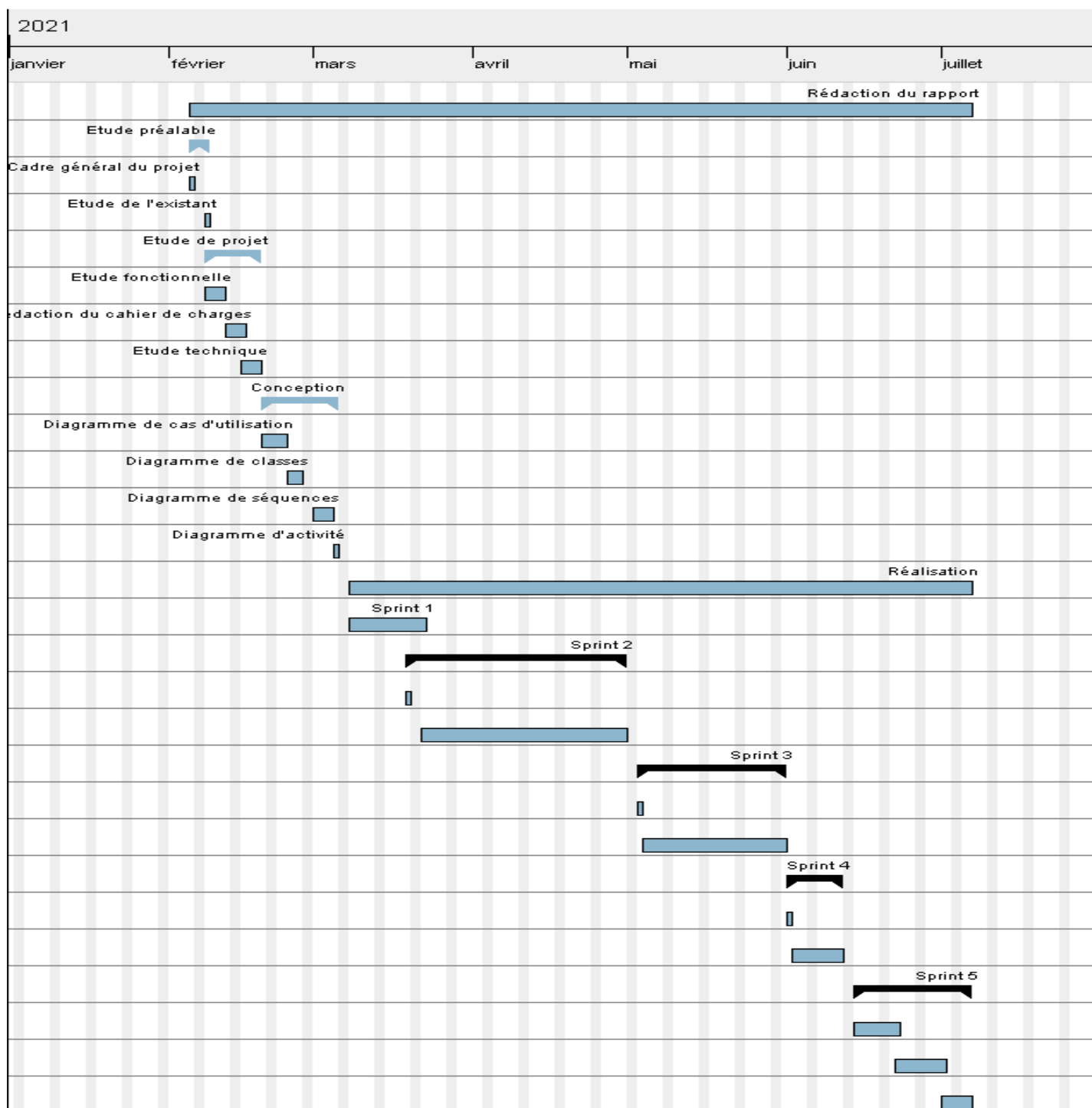


Figure 7 : Diagramme Gantt de notre projet

Conclusion

Dans ce chapitre nous avons porté une critique sur l'existant. Ce qui nous a permis de décrire le cahier de charges fonctionnel du futur système. Ensuite, nous avons abordé notre démarche de développement avec Scrum. Enfin, nous avons présenté un aperçu de la planification du projet avec l'appui d'un diagramme de Gantt.

Chapitre III

Etude technique du futur système

Introduction

Tout au long de ce chapitre, nous présenterons l'étude technique détaillée de notre projet. Nous allons évoquer l'architecture logicielle de notre futur système ainsi que la plateforme de développement adoptée, et les outils et techniques utilisés.

I. Architecture logicielle

Parmi les différentes façons de structurer une architecture, la mieux adaptée et maîtrisée en informatique est l'approche par couches. Une couche (Layer en anglais) est une division logique et horizontale d'un système qui fournit une abstraction particulière du système à couches supérieures. Chaque couche possède des responsabilités spécifiques. Dans une structuration par couches, les couches inférieures prennent en charges des responsabilités qui offrent un socle de base pour les couches supérieures, permettant d'abstraire celles-ci de problématiques qui ne les concernent pas. Ainsi, nous avons adopté un découpage en quatre couches. Une telle architecture permet également d'obtenir un bon niveau de réutilisabilité, à travers la réplication des solutions aux problèmes rencontrés pour chaque couche, dans d'autres applications utilisant la même architecture. Ce volet technique (découpage en couches) a été testé par un projet prototype qui nous a permis de valider notre architecture logicielle et que les messages voyagent bien depuis la couche de présentation jusqu'à la couche de stockage.

1. Couche Présentation

Cette couche correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'Interface Homme-Machine (**IHM**). Dans la plupart des cas, il s'agit d'une interface client riche ou d'une interface web. Dans notre cas, les langages Javascript, HTML/CSS que nous aborderons par la suite, avec l'appui des frameworks Angular, JQuery, Bootstrap sont utilisés pour construire cette couche. Elle intercepte les événements et fait appel aux services de la couche ViewModel.

2. Couche ViewModel

Cette couche regroupe les classes nécessaires à la logique métier, elle comporte donc les services offerts par le système à l'utilisateur. Ces classes de cette couche sont créées à l'aide de notre langage PHP utilisé avec l'appui du framework Laravel que

nous donnerons également un aperçu à la suite de ce document. Une classe de cette couche est appelée « **Model** » selon la nomenclature de Laravel. Ces services ne sont autres que l'implémentation des scénarios identifiés lors de la phase de conception. Lors de l'appel d'un service, ce dernier fait appel aux objets métier nécessaires pour répondre à la requête de l'utilisateur.

3. Couche Service

C'est une couche intermédiaire entre les couches ViewModel et la couche DAO (Data Access Object), son rôle est de gérer les transactions qui seront ouverts à chaque appel de la base de données. Une classe de cette couche est nommée « **Controller** » vue son rôle d'aiguillage et de répartition des ressources entre les utilisateurs de l'application afin de garantir une meilleure performance du serveur (utilisation des ressources seulement à la demande).

4. Couche d'accès aux données

C'est la couche qui regroupe les classes PHP capables de fournir les données nécessaires à la couche métier (ViewModel), et par la suite aux services demandés par l'utilisateur, cette couche est modélisée dans notre système par des composants DAO. Un DAO est un design pattern (patron de conception) utilisé dans les architectures logicielles objet visant à isoler la logique de persistance dans des classes d'accès aux données. En fait, le framework Laravel intègre un ORM(Objet-Relation-Mapping) appelé « **Eloquent** » qui gère le mappage des données entre les tables de la base de données de la couche de stockage et les modèles de la couche ViewModel. Cette couche est abstraite du point de vue que les requêtes SQL (Strutured Query Language)³⁰ sont automatiquement en arrière-plan après l'action d'un utilisateur.

5. Couche stockage

Cette couche est responsable du stockage des données dans une base de données relationnelles dont le modèle physique est déduit à partir du diagramme de classes du système. Elle est gérée dans notre cas par le SGBD MySQL.

³⁰ Le SQL (Structured Query Language) est un langage permettant de communiquer avec une base de données.

6. Couche sécurité

Il s'agit d'une couche dédiée à la sécurité, chargée de vérifier l'authentification et les autorisations d'accès aux pages et aussi le hachage des mots de passes des utilisateurs.

II. Architecture Réseau

Le scenario consiste à mettre en place une application de type client/serveur à trois niveaux comme décrit ci-dessus. Chaque poste de travail communiquera avec une base de données commune via le réseau local. Les utilisateurs qui auront accès à l'application sont les agents de la SGBF. Ils y accéderont selon leurs droits à travers un navigateur web (Microsoft Edge, Mozilla Firefox ou Google chrome) sur ordinateur bureau ou portable connecté au réseau.

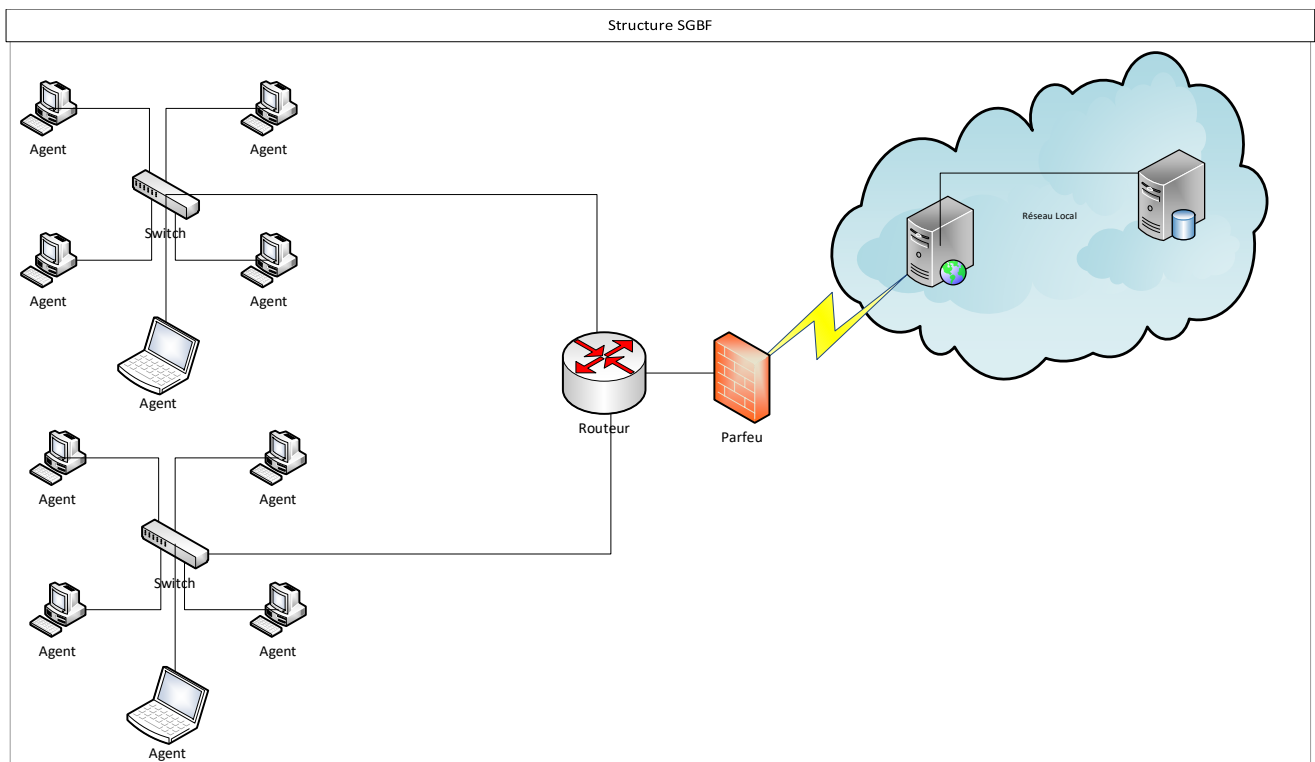


Figure 8 : Architecture réseau

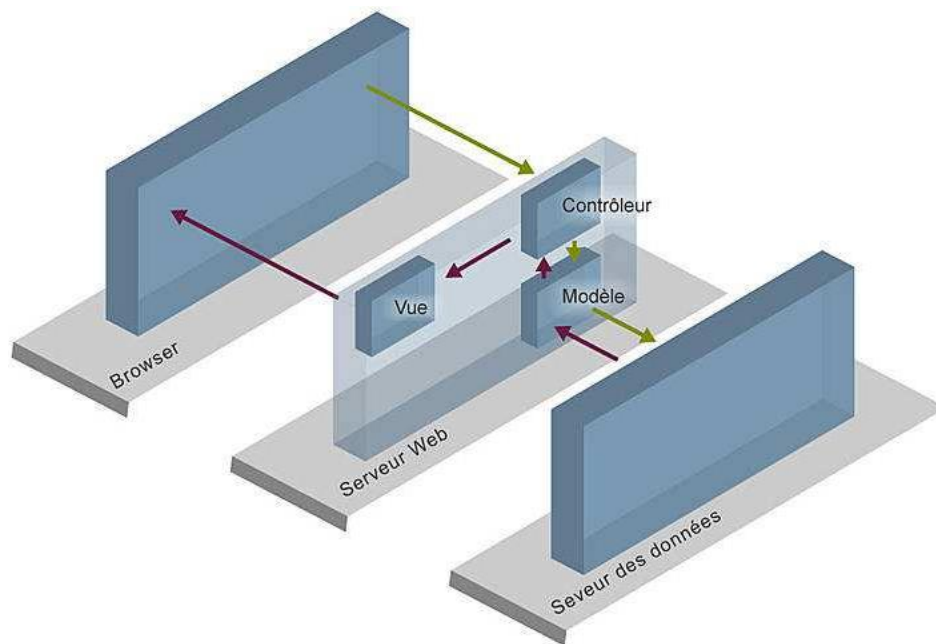
III. Méthodologie de développement : Démarche MVC

Le modèle d'architecture qui cherche à séparer nettement les couches de présentation (UI : User Interface), métier (BLL : Business Logic Layer) et d'accès aux données (DAL : Data Access Layer).

Le but étant d'avoir une dépendance minimale entre les différentes couches de l'application, ainsi les modifications effectuées sur n'importe quelle couche de l'application n'affectent pas les autres couches :

- **Modèle** : encapsule le cœur fonctionnel de l'application, le domaine logique.
- **Vue** : les données sont envoyées, par le modèle, à la vue qui les représente à l'utilisateur.
- **Contrôleur** : reçoit les données et les transmet au modèle ou à la vue.

Une telle architecture est communément appelée « **architecture 3 tiers** » ou « **architecture à 3 niveaux** » :



31

Figure 9 : « Architecture 3 tiers »

³¹ <https://www.cours-gratuit.com/cours-php/cours-avance-du-langage-php5-et-mvc>

Pourquoi MVC ?

Comme nous l'avons déjà cité dans l'étude fonctionnelle, notre application doit être à la disposition de tous ses utilisateurs (administrateur, utilisateur).

C'est pourquoi nous avons décidé qu'elle soit une application WEB, comme ça elle va être accessible à tous les utilisateurs. Donc nous avons été amenés à faire une étude sur les Frameworks permettant de réaliser ce type d'application.

L'objectif de MVC est de faire collaborer deux équipes : une équipe à consonance infographie et une équipe à consonance informatique. Le paradigme MVC est un schéma de programmation qui propose de séparer une application en 3 parties :

- Le modèle qui contient la logique et l'état de l'application.
- La vue qui représente l'interface utilisateur.
- Le contrôleur qui gère la synchronisation entre la vue et le modèle. Le contrôleur réagit aux actions de l'utilisateur en effectuant les actions nécessaires sur le modèle et surveille les modifications du modèle et informe la vue des mises à jour nécessaires.

IV. La plateforme PHP

Le langage PHP a été inventé par Rasmus LERDORF en 1995 pour son usage personnel. Autrefois abréviation de « Personal HomePage » devenue aujourd'hui « Hypertext Preprocessor », PHP s'impose comme un standard dans le monde de la programmation web par ses performances, sa fiabilité, sa souplesse et sa rapidité.³² Il faut dire que notre structure a adopté PHP comme langage principal de développement et Laravel comme framework PHP. La version utilisée pour notre projet est **PHP 7.1.3**. Mais nous tenons tout de même à vous présenter une étude comparative pour vous montrer le choix de ce framework devenu populaire. Tout d'abord selon le blog « mobiskill », Laravel et Symfony sont principalement les deux frameworks PHP les plus utilisés en 2021.³³ Nous avons aussi réalisé avec « Google Trends » qui va nous permettre de savoir la fréquence de recherche de ces deux termes sur le plus grand moteur de recherche Google sur les cinq (5) dernières années dans tous les pays. Ainsi, les résultats qui se présentent dans les figures suivantes nous ont

³² <https://apprendre-php.com/tutoriels/tutoriel-19-introduction-au-langage-php.html>

³³ <https://mobiskill.fr/blog/conseils-emploi-tech/les-meilleurs-frameworks-php-a-utiliser-en-2021/>

donné une idée globale sur le framework PHP le plus utilisé ou autrement dit le framework qui a la plus grande quantité des ressources sur Internet :

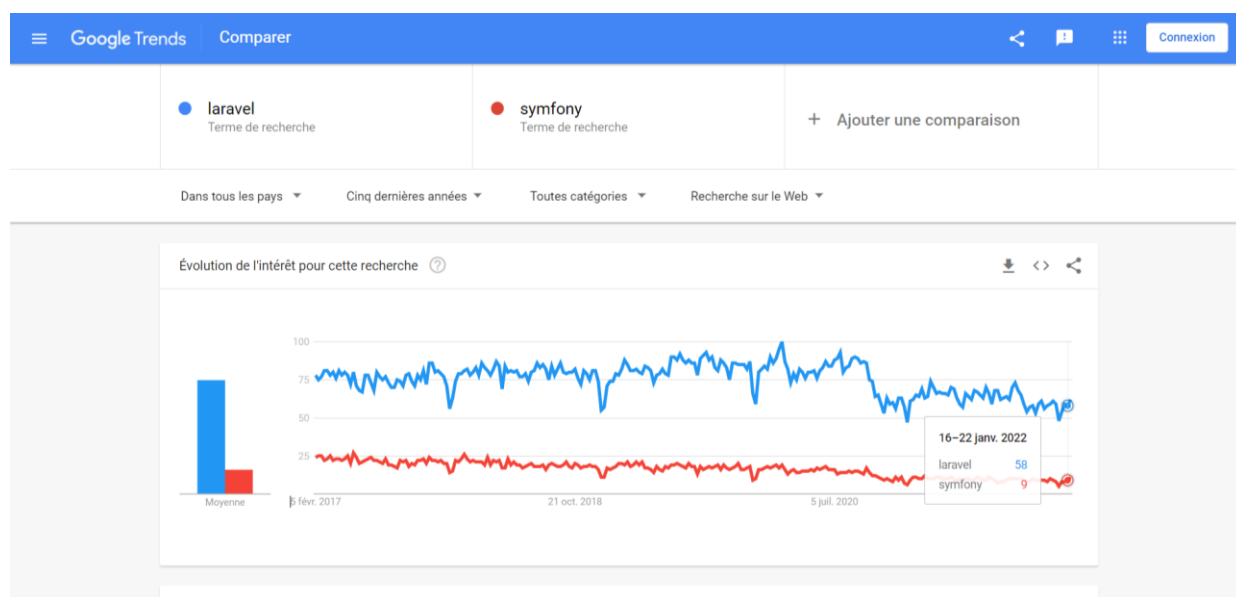


Figure 10 : Comparaison entre Laravel et Symfony

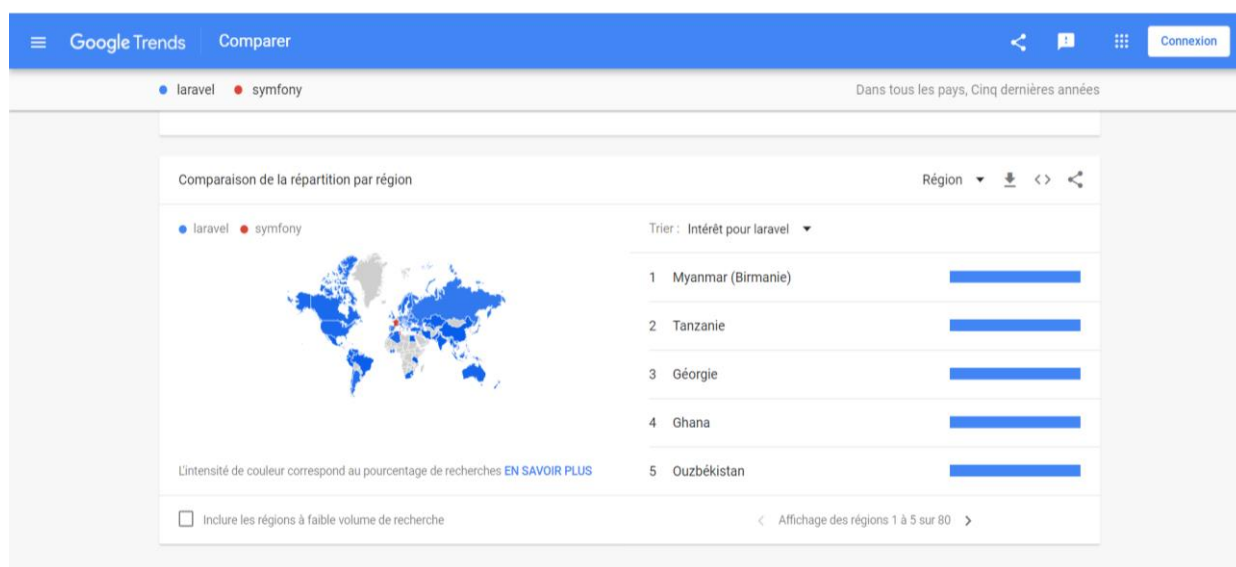


Figure 11 : Comparaison entre Laravel et Symfony, répartition par région

On peut conclure que le terme « Laravel » est le plus utilisé sur Google en cette dernière année 2021 et jusqu'à aujourd'hui.

³⁴ <https://trends.google.fr/trends/explore?date=today%205-y&q=laravel,symfony>

³⁵ <https://trends.google.fr/trends/explore?date=today%205-y&q=laravel,symfony>

V. Frameworks

Un framework est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

1. Laravel



36

Figure 12 : Logo de Laravel

Laravel est un framework libre pour construire et définir l'infrastructure d'une application PHP, dont il facilite le développement et les tests. Il a été créé par Taylor Otwell en juin 2011³⁶.

Première version	15 juin 2011
Dernière version	8.27 (10 février 2021)
Environnement	Plateforme Web
Ecrit-en	PHP
Type	Framework
Licence	Licence MIT ³⁸
Site web	www.laravel.com

Tableau 3 : Fiche technique de Laravel

Pourquoi notre structure s'oriente véritablement vers le framework Laravel ?

« Laravel est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub. »³⁹

³⁶ <https://laravel.com/>

³⁷ <https://fr.wikipedia.org/wiki/Laravel>

³⁸ https://fr.wikipedia.org/wiki/Licence_MIT

³⁹ <https://fr.wikipedia.org/wiki/Laravel>

Laravel est aujourd'hui une solution largement reconnue dans le monde informatique et très populaire pour le développement de logiciels personnalisés. Selon Taylor Otwell, le créateur de Laravel : « Laravel est le plus puissant rival de l'écosystème PHP, simplement parce qu'il inclut les fonctionnalités nécessaires à la création d'applications Web modernes ».⁴⁰

Principe et avantages :

✓ Sécurité et performance

L'un des avantages les plus importants du choix de Laravel pour le développement de vos applications Web réside dans ses capacités à fournir une sécurité de haut niveau⁴¹. Si vous choisissez Laravel, votre application Web ne présente aucun risque d'injections SQL involontaires et cachées.

De plus, Laravel est capable de fournir une excellente performance des applications Web. Il arrive que certaines caractéristiques et fonctionnalités affectent les performances du site. Mais Laravel propose divers outils qui aident les développeurs à améliorer leurs performances.

✓ Bibliothèques orientées objet

Laravel est l'un des meilleurs frameworks PHP car il possède des bibliothèques orientées objet⁴² et d'autres préinstallées, qui ne se trouvent dans aucun autre framework PHP. L'une des bibliothèques pré installées est la bibliothèque d'authentification. Ces bibliothèques regorgent d'excellentes fonctionnalités faciles à utiliser et à implémenter pour chaque développeur.

✓ Documentation et communauté

Laravel possède une puissante communauté de développeurs qui fournit en permanence une assistance pour la rendre plus flexible et évolutive. Ainsi, si vous souhaitez apporter des fonctionnalités complexes, de nombreuses documentations sont à votre disposition.

⁴⁰ <https://easypartner.fr/blog/pourquoi-choisir-le-framework-laravel/>

⁴¹ <https://www.cloudways.com/blog/best-laravel-security-practices/>

⁴² <https://www.apprendre-laravel.fr/laraguide/2017-11-25-utilisation-de-bibliotheques-externes>

✓ Tests unitaires

Avec les tests unitaires de Laravel, chaque module de votre application Web est testé avant la mise en ligne du site. Ces tests garantissent une application performante, sans bug et finalement sans tracas pour vos utilisateurs finaux. C'est une autre exception du framework Laravel.

✓ Artisan

Laravel fournit un outil de ligne de commande intégré appelé Artisan. Cet outil aide à créer une architecture de code « squelette » et de base de données, ainsi que leurs migrations. La gestion de la base de données en devient plus facile. L'outil Artisan permet d'effectuer presque toutes les tâches de programmation répétitives et fastidieuses.

2. Angular



Figure 13 : Logo d'Angular

Angular est une plateforme de développement, construite sur **TypeScript**⁴⁴ et TypeScript lui-même basé sur **JavaScript** qui est un langage de script côté client ou utilisateur facilitant l'Interaction Homme Machine (IHM). En tant que plate-forme, Angular comprend :

- Un cadre basé sur des composants pour créer des applications Web évolutives ;
- Une collection de bibliothèques bien intégrées qui couvrent une grande variété de fonctionnalités, y compris le routage, la gestion des formulaires, la communication client-serveur ;
- Une suite d'outils de développement pour vous aider à développer, construire, tester et mettre à jour votre code.

⁴³ <https://angular.io/>

⁴⁴ <https://www.typescriptlang.org/>

Avec Angular, vous profitez d'une plate-forme qui peut évoluer de projets à développeur unique à des applications au niveau de l'entreprise. Angular est conçu pour rendre la mise à jour aussi simple que possible, alors profitez des derniers développements avec un minimum d'effort. Mieux encore, l'écosystème Angular se compose d'un groupe diversifié de plus de 1,7 million de développeurs, d'auteurs de bibliothèques et de créateurs de contenu.⁴⁵ **Angular 7.3** est la version utilisée pour réaliser notre application.

3. HTML5/CSS3



46

Figure 14 : Logo HTML5

Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour représenter les pages web. Cette version a été finalisée le 28 octobre 2014.⁴⁷

Ce langage permet :

- d'écrire de l'hypertexte, d'où son nom ;
- de structurer sémantiquement la page ;
- de mettre en forme le contenu ;
- de créer des formulaires de saisie ;
- d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques ;
- de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

⁴⁵ <https://angular.io/guide/what-is-angular>

⁴⁶ https://fr.wikipedia.org/wiki/HTML5#/media/Fichier:HTML5_logo_and_wordmark.svg

⁴⁷ <https://fr.wikipedia.org/wiki/HTML5>

Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). HTML est inspiré du Standard Generalized Markup Language (SGML)⁴⁸. Il s'agit d'un format ouvert.

HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM⁴⁹ : HTML5 et XHTML5⁵⁰.



Figure 15 : Logo CSS3

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML⁵². Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C)⁵³. Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.⁵⁴

⁴⁸ https://fr.wikipedia.org/wiki/Standard_Generalized_Markup_Language

⁴⁹ Le Document Object Model (DOM) est une interface de programmation normalisée par le W3C, qui permet à des scripts d'examiner et de modifier le contenu du navigateur web.

⁵⁰ XHTML est un langage de balisage servant à écrire des pages pour le World Wide Web. Conçu à l'origine comme le successeur de HTML.

⁵¹ https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade#/media/Fichier:CSS3_logo_and_wordmark.svg

⁵² https://fr.wikipedia.org/wiki/Extensible_markup_language

⁵³ https://fr.wikipedia.org/wiki/World_Wide_Web_Consortium

⁵⁴ https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

4. Bootstrap



Figure 16 : Logo de Bootstrap

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.⁵⁵

VI. Environnement de travail

1. Environnement de développement



Figure 17 : Logo Visual Studio Code

Un environnement de développement intégré (EDI ou IDE en anglais pour Integrated Development Environment) est un programme regroupant un ensemble d'outils pour le développement de logiciels. En règle générale, un EDI regroupe un éditeur de texte, un compilateur, des outils automatiques de fabrication, et souvent un débogueur. L'environnement de développement choisi pour ce projet est **Visual Studio Code 1.63** après étude comparative du tableau ci-dessous :

⁵⁵ [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)#/media/Fichier:Bootstrap_logo.svg](https://fr.wikipedia.org/wiki/Bootstrap_(framework)#/media/Fichier:Bootstrap_logo.svg)

⁵⁶ [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))

⁵⁷ <https://code.visualstudio.com/>

Désignation	Avantages	Inconvénients	Acquisition
Dreamweaver⁵⁸	<ul style="list-style-type: none"> • outil visuel de développement d'application web et de site web dynamique ; • intégration parfaite des scripts PHP et JavaScript ; • support complet avec assistance du langage HTML ; • logiciel difficile à appréhender par un novice sans une formation spécifique ; 	<ul style="list-style-type: none"> • logiciel difficile à appréhender par un novice sans une formation spécifique ; 	Payant
NetBeans⁵⁹	<ul style="list-style-type: none"> • facilité de faire des interfaces ; • interfaces conviviales ; • évènementiel ; • Assez robuste ; 	<ul style="list-style-type: none"> • demande une rigueur dans le codage ; • demande beaucoup de mémoire ; 	Gratuit
Visual Studio Code⁶⁰	<ul style="list-style-type: none"> • facilité de faire des interfaces ; • interfaces conviviales ; • évènementiel ; • assez robuste ; • multi terminal intégré : possibilité d'exécuter à la fois les commandes du système Windows et les commandes 	<ul style="list-style-type: none"> • demande une rigueur dans le codage ; 	Gratuit

⁵⁸ <https://www.adobe.com/fr/products/dreamweaver.html>

⁵⁹ <https://netbeans.apache.org/>

⁶⁰ <https://code.visualstudio.com/>

	basiques du système linux. <ul style="list-style-type: none"> • Espace de travail personnalisable selon la convenance • Très léger (moins de ressources mémoire) 		
--	---	--	--

Tableau 4 : Etude comparative des environnements de développement

2. Outil de modélisation



61

Figure 18 : Logo de Visio

Microsoft Visio (officiellement Microsoft Office Visio) est un logiciel de diagrammes et de synoptique pour Windows qui fait partie de la suite bureautique Microsoft Office mais se vend séparément. On peut ainsi créer des diagrammes de Gantt, des réseaux de PERT ou encore des diagrammes IDEFO.

Dans Visio, les graphiques utilisés pour créer des diagrammes sont vectoriels. Les versions Standard et Professionnel de l'édition 2019 partagent la même interface, mais cette dernière permet de faire des diagrammes plus avancés, grâce à des modèles supplémentaires. Cette version offre également une fonctionnalité unique qui permet aux utilisateurs de relier facilement leurs diagrammes à un grand nombre de sources de données et d'afficher les informations recueillies graphiquement. Dans notre projet nous avons utilisé Microsoft Office Visio pour la création des diagrammes de cas d'utilisation, diagramme de classe, diagramme d'activité et les diagrammes de séquence.

⁶¹ [https://fr.wikipedia.org/wiki/Microsoft_Visio#/media/Fichier:Microsoft_Office_Visio_\(2019\).svg](https://fr.wikipedia.org/wiki/Microsoft_Visio#/media/Fichier:Microsoft_Office_Visio_(2019).svg)

3. Système de Gestion de Base de Données

Un système de gestion de base de données (SGBD) est un logiciel destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations. Pour la gestion de base de données, **MySQL 8.0.20** a été retenu comme SGBD (Système de Gestion de Base de Données). Ce choix a été opéré suite à une étude comparative consignée dans le tableau ci-dessous. Ce tableau est le résultat de recherches GOOGLE séparées concernant les avantages et inconvénients de chaque SGBD cité :

Désignation	Avantages	Inconvénients	Acquisition
MySQL ⁶²	<ul style="list-style-type: none"> Fonctionne sous plusieurs plateformes ; Système de droits et de mots de passe souple et sécurisé. 	<ul style="list-style-type: none"> Ne supporte pas les standards SQL92 ; Support incomplet des triggers et des procédures stockées. 	Gratuit
Oracle ⁶³	<ul style="list-style-type: none"> Très robuste avec de grosses volumétries de donnée ; Supporte un grand nombre d'utilisateurs. 	<ul style="list-style-type: none"> Forte demande de ressources ; Administration complexe ; Distribution fortement liée au système d'exploitation. 	Propriétaire
SQL server ⁶⁴	<ul style="list-style-type: none"> Administration aisée ; Compressions des données et des sauvegardes. 	<ul style="list-style-type: none"> Mono-plateforme (Windows seulement). 	Propriétaire

Tableau 5 : Etude comparative des SGBD

⁶² <https://www.mysql.com/fr/>

⁶³ <https://www.oracle.com/fr/index.html>

⁶⁴ <https://www.microsoft.com/fr-fr/sql-server/sql-server-2019>

Conclusion

Au terme de ce chapitre, nous avons capturé les besoins techniques du projet, ainsi présenté l'architecture logicielle ainsi que l'architecture de notre système, et ensuite nous avons cité les différents outils et l'environnement de travail et le rapport détaillé des frameworks adoptés.

Chapitre IV

Conception détaillée du futur système

Introduction

Ce présent chapitre a pour objectif de détailler la conception du futur système, partant par l'architecture logique de l'application puis la conception détaillée(diagrammes).

I. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML (Unified Modeling Language) utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. UML est un langage de modélisation graphique à base de pictogrammes conçue comme une méthode normalisée de visualisation dans les domaines du développement logiciel en conception orientée objet.⁶⁵ Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet. UML s'adapte beaucoup plus aux méthodes agiles comme la nôtre qui est Scrum. Il a été donc un choix rapide pour notre projet.

⁶⁵ [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)),

1. Identification des acteurs

Un acteur représente un élément externe qui interagit avec le système. Cet élément peut être un utilisateur ou un système tiers (autre ordinateur, autre programme, base de données). En observant les utilisateurs directs du système ainsi que les autres acteurs qui interagissent avec le système que nous essayons de définir, nous trouvons ces acteurs qui opèrent avec le système : le Chargé Garanties, le Responsable Garanties, le Conseiller Clientèle, l'Administrateur.

Acteur	Description
Chargé Garanties	Cet acteur occupe beaucoup plus un rôle de saisie. Le système devra lui permettre de saisir ou annuler une garantie, une assurance et les originaux des titres.
Responsable Garanties	Il doit vérifier et annuler ou valider, lier une garantie à un engagement, échoir une garantie, annuler ou valider une assurance et quelques paramètres (nouveau notaire, nouveau évaluateur, nouvelle localité immeuble).
Conseiller Clientèle	Il pourra juste consulter les dossiers de garanties saisis et validés.
Administrateur	Il doit habilitier les utilisateurs et le paramétrer le système.

Tableau 6 : Identification des acteurs

2. Identification des cas d'utilisations

Un cas d'utilisation est une entité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système sans imposer le mode réalisation de ce service.

Cas d'utilisation « S'authentifier » :

Titre	S'authentifier
Acteurs	Chargé Garanties, Responsable Garanties, Conseiller Clientèle, Administrateur
Pré-conditions	L'utilisateur (acteur) saisit le login (nom d'utilisateur) et le mot de passe.
Scénarios	Identification : <ul style="list-style-type: none"> • l'utilisateur saisit ses droits d'accès (login et mot de passe) ; • le système vérifie si les champs ne sont pas vides, si erreur alors un message d'erreur s'affiche ; • il vérifie ensuite si les informations sont valides, si erreur alors Exception ; • le système redirige l'acteur vers la page d'accueil selon son rôle.
Post-conditions	L'accès au système.

Tableau 7 : Cas d'utilisation « S'authentifier »

Cas d'utilisation « Saisir une garantie » :

Titre	Saisir une garantie
Acteurs	Chargé Garanties
Pré-conditions	<p>L'utilisateur doit avoir le droit d'accès ;</p> <p>L'utilisateur doit s'authentifier ;</p> <p>L'utilisateur doit avoir l'habilitation nécessaire ;</p>
Scénarios	<p>Etapes :</p> <ul style="list-style-type: none"> • l'utilisateur soumet une demande de saisir une nouvelle garantie ; • le système l'envoie le formulaire de saisie de nouvelle garantie ; • l'usager remplit le formulaire et clique sur le bouton « Enregistrer » ; • le système vérifie si les champs obligatoires sont complets si erreur alors un message d'erreur s'affiche ; • le système vérifie si les champs obligatoires et optionnels sont valides si erreur alors un message d'erreur s'affiche ; • le système enregistre la nouvelle garantie.
Post-conditions	La garantie est enregistrée avec le statut « en attente » et la situation « en cours », et l'utilisateur peut y accéder à nouveau et la modifier.

Tableau 8 : Cas d'utilisation « Saisir une garantie »

Cas d'utilisation « Mettre à jour une garantie » :

Titre	Mettre à jour une garantie
Acteurs	Chargé Garanties
Pré-conditions	<p>L'utilisateur doit avoir le droit d'accès ;</p> <p>L'utilisateur doit s'authentifier ;</p> <p>L'utilisateur doit avoir l'habilitation nécessaire ;</p> <p>La garantie doit avoir le statut « en attente ».</p>
Scénarios	<p>Etapes :</p> <ul style="list-style-type: none"> • l'utilisateur consulte la liste des garanties existantes ; • l'utilisateur choisit une garantie à mettre à jour parmi celles de la liste ; • le système l'envoi le formulaire de mise à jour ; • l'utilisateur effectue les mises à jour nécessaires des champs concernés ; • le système vérifie la validité des champs et si les champs obligatoires sont complets, si erreur un message d'erreur s'affiche ; • le système sauvegarde les nouvelles informations fournies par l'utilisateur.
Post-conditions	Les nouvelles mises à jour sont appliquées à la garantie.

Tableau 9 : Cas d'utilisation « Mettre à jour une garantie »

Cas d'utilisation « Consulter une garantie » :

Titre	Consulter une garantie
Acteurs	Chargé Garanties, Responsable Garanties, Conseiller Clientèle
Pré-conditions	L'utilisateur doit avoir le droit d'accès ; L'utilisateur doit s'authentifier ; L'utilisateur doit avoir l'habilitation nécessaire ;
Scénarios	<p>Etapes :</p> <ul style="list-style-type: none"> • l'utilisateur consulte la liste des garanties existantes ; • l'utilisateur choisit une garantie à consulter parmi celles de la liste ; • le système l'envoie la fenêtre des détails à consulter ; • l'utilisateur consulte les détails nécessaires de la garantie ; • l'utilisateur clique sur le bouton « Fermer » pour fermer la fenêtre des détails ;
Post-conditions	La garantie est consultée par l'utilisateur.

Tableau 10 : Cas d'utilisation « Consulter une garantie »

Cas d'utilisation « Valider une garantie » :

Titre	Valider une garantie
Acteurs	Responsable Garanties
Pré-conditions	<p>L'utilisateur doit avoir le droit d'accès ;</p> <p>L'utilisateur doit s'authentifier ;</p> <p>L'utilisateur doit avoir l'habilitation nécessaire ;</p> <p>La garantie doit avoir le statut « en attente ».</p>
Scénarios	<p>Etapes :</p> <ul style="list-style-type: none"> • l'utilisateur consulte la liste des garanties existantes ; • l'utilisateur coche une garantie à valider parmi celles de la liste ; • l'utilisateur clique sur le bouton « Valider » pour valider la garantie ; • le système l'affiche une fenêtre de confirmation ; • l'utilisateur confirme la validation ; • le système sauvegarde le nouveau statut de la garantie.
Post-conditions	La garantie a un nouveau statut « validé », et l'utilisateur ne peut plus mettre à jour certains informations de la garantie.

Tableau 11 : Cas d'utilisation « Valider une garantie »

Cas d'utilisation « Lier une garantie à un engagement » :

Titre	Lier une garantie à un engagement
Acteurs	Responsable Garanties
Pré-conditions	<p>L'utilisateur doit avoir le droit d'accès ;</p> <p>L'utilisateur doit s'authentifier ;</p> <p>L'utilisateur doit avoir l'habilitation nécessaire ;</p> <p>La garantie doit avoir un statut « validé » ;</p> <p>La garantie doit avoir une situation « en cours » ;</p> <p>La garantie doit être liée un prêt.</p>
Scénarios	<p>Etapes :</p> <ul style="list-style-type: none"> • l'utilisateur consulte la liste des garanties existantes ; • l'utilisateur choisit une garantie à lier parmi celles de la liste ; • le système l'affiche la liste des engagements du client ; • l'utilisateur coche l'engagement(prêt) concerné ; • l'utilisateur clique sur le bouton « Lier » pour lier l'engagement à la garantie ; • le système affiche la fenêtre de confirmation ; • l'utilisateur confirme la liaison ; • le système sauvegarde la liaison de la garantie à l'engagement.
Post-conditions	La garantie est liée à l'engagement, et l'utilisateur peut y accéder à nouveau et dissocier la liaison.

Tableau 12 : Cas d'utilisation « Lier une garantie à un engagement »

Cas d'utilisation « Lever une garantie » :

Titre	Lever une garantie
Acteurs	Responsable Garanties
Pré-conditions	<p>L'utilisateur doit avoir le droit d'accès ;</p> <p>L'utilisateur doit s'authentifier ;</p> <p>L'utilisateur doit avoir l'habilitation nécessaire ;</p> <p>La garantie doit avoir un statut « validé » ;</p> <p>La garantie doit avoir une situation « en cours ».</p>
Scénarios	<p>Etapes :</p> <ul style="list-style-type: none"> • l'utilisateur consulte la liste des garanties existantes ; • l'utilisateur coche une garantie à lever parmi celles de la liste ; • l'utilisateur clique sur le bouton « Lever » pour lever la garantie ; • le système l'affiche une fenêtre de confirmation ; • l'utilisateur confirme la levée ; • le système sauvegarde la levée de la garantie.
Post-conditions	La garantie est levée et a une nouvelle situation « échue ».

Tableau 13 : Cas d'utilisation « Lever une garantie »

3. Diagramme de cas d'utilisation principal ou de base

Le diagramme suivant présente les cas d'utilisations des quatre modules de base : gestion des garanties, gestion des assurances, gestion de la sécurité, gestion paramétrage. Les acteurs principaux sont le Chargé Garanties, le Responsable Garanties, le Conseiller Clientèle et l'Administrateur. Le Conseiller Clientèle n'a qu'un rôle de consultation dans la gestion des garanties. Le Chargé Garanties participe à la gestion des garanties ainsi qu'à la gestion des assurances. Tandis que, le Responsable Garanties en plus de contribuer à la gestion des garanties et la gestion des assurances, il intervient également à la gestion du paramétrage. L'administrateur a un rôle central dans la gestion de la sécurité et la gestion du paramétrage du système. L'accès de chaque utilisateur au système nécessite une authentification préalable.

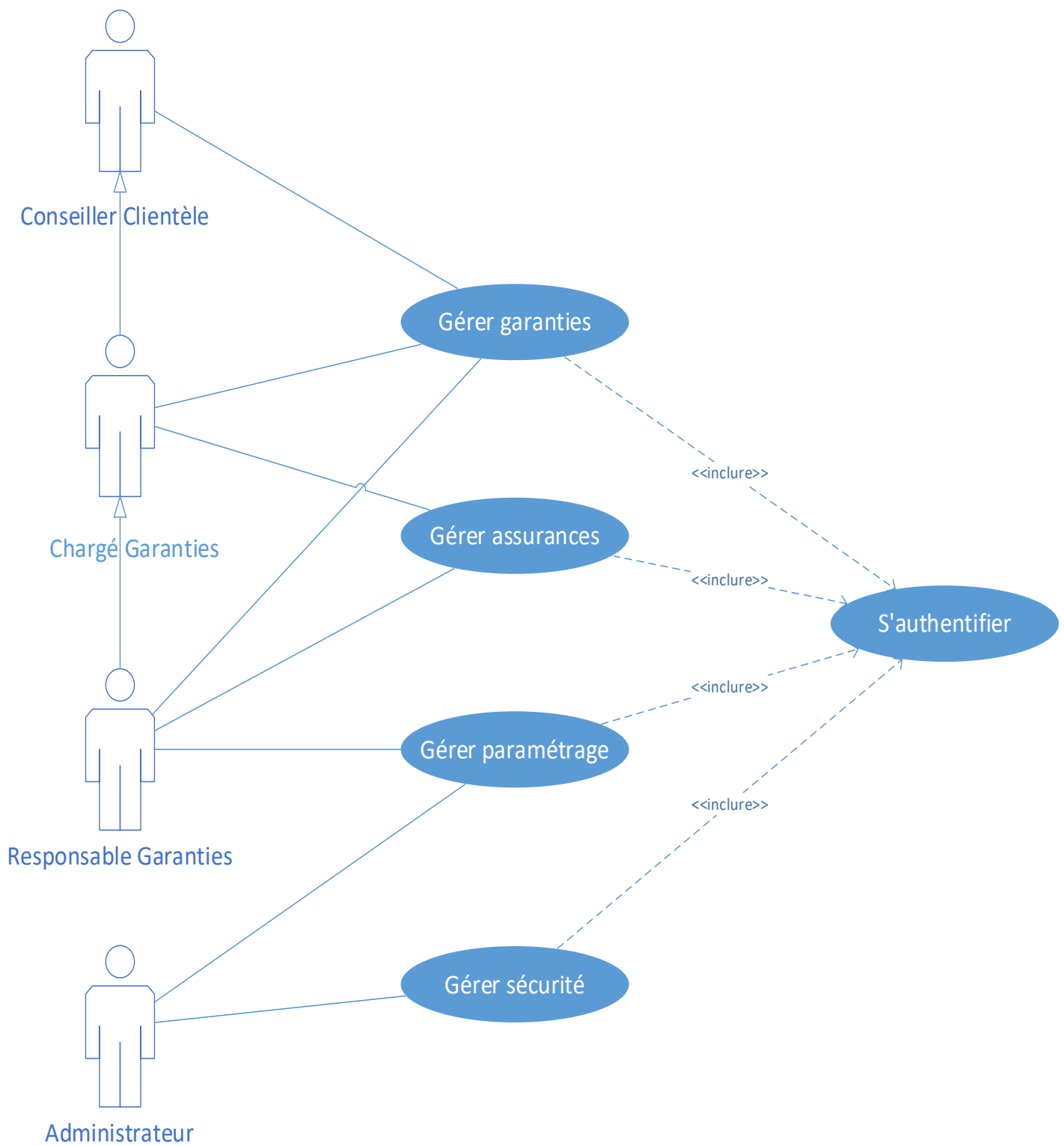


Figure 19 : Diagramme de cas d'utilisation principal ou de base

4. Diagramme de cas d'utilisation : pour le Chargé Garanties

Ce diagramme suivant montre les différentes actions menées par le Chargé Garanties après authentification par le système sur les deux modules de gestion (gestion des garanties et des assurances). Il a comme rôle principal la saisie des garanties et des assurances dans le système qui constitue la porte d'entrée des données.

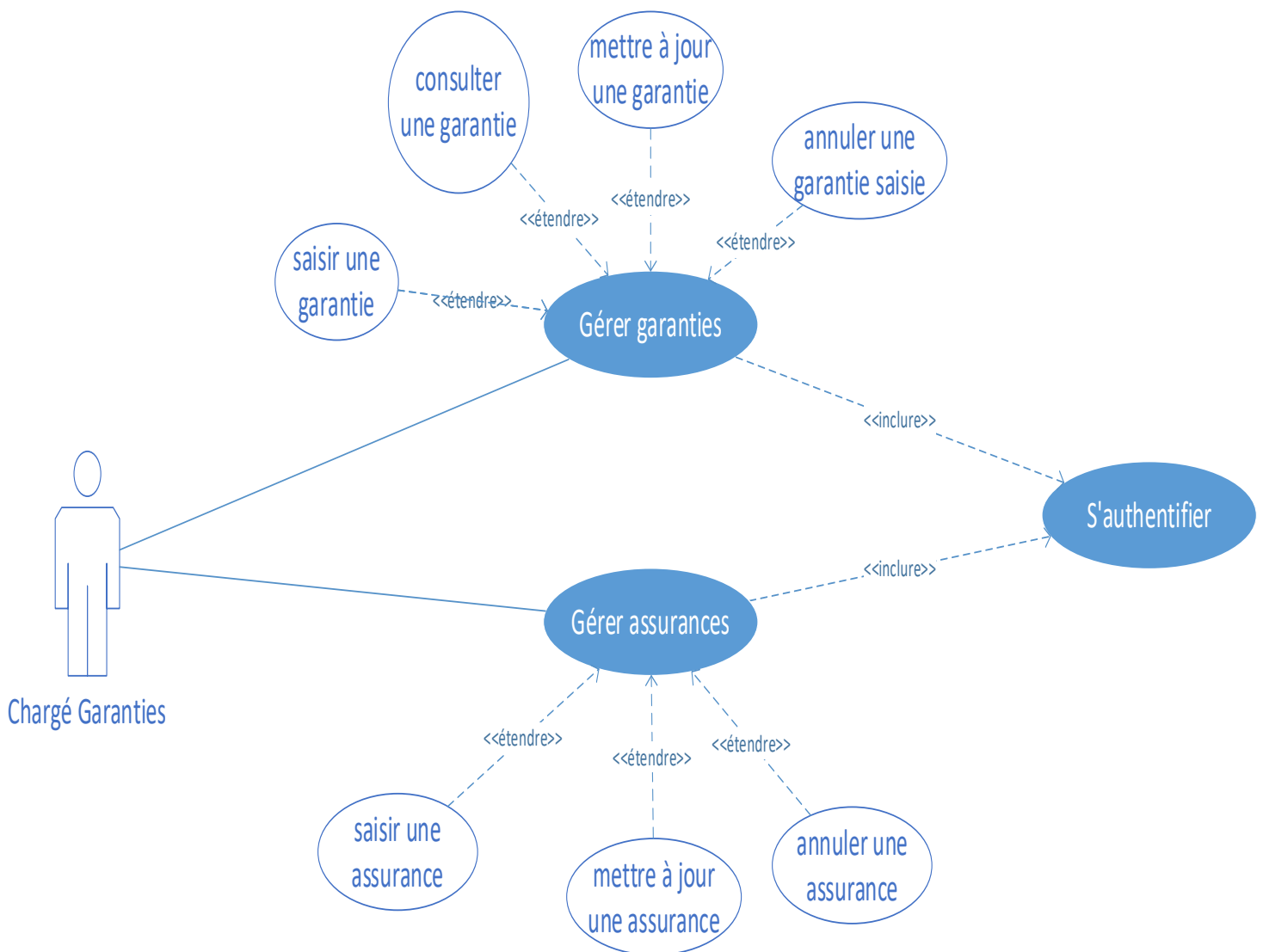


Figure 20 : Diagramme de cas d'utilisation : pour le Chargé Garanties

5. Diagramme de cas d'utilisation : pour le Responsable Garanties

Le Responsable Garanties intervient beaucoup plus dans la validation et le paramétrage des données de garanties et des assurances comme présente la figure suivante :

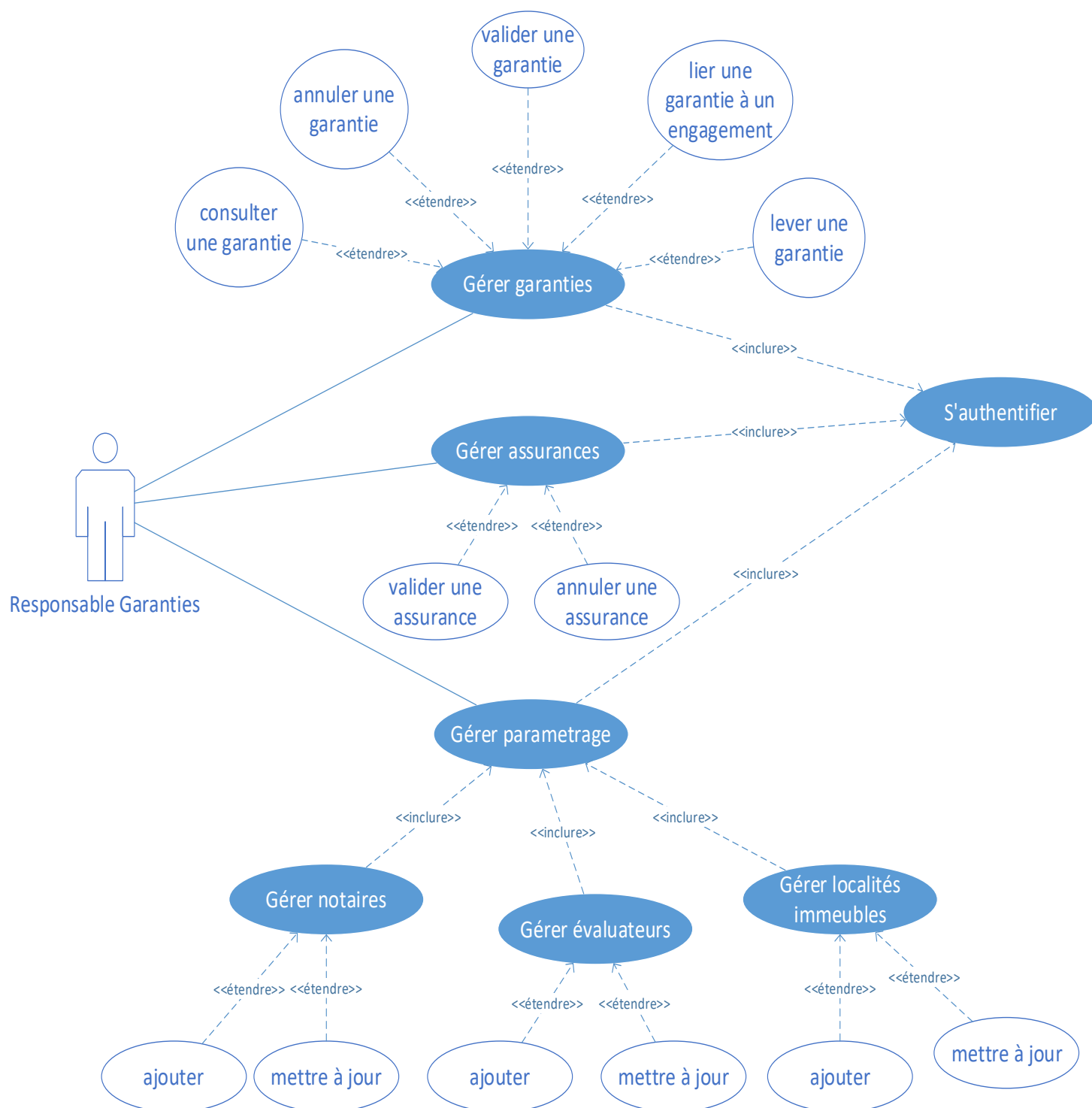


Figure 21 : Diagramme de cas d'utilisation : pour le Responsable Garanties

6. Diagramme de cas d'utilisation : pour le Conseiller Clientèle

Le Conseiller Clientèle n'a qu'un rôle de consultation des garanties saisies et validées.

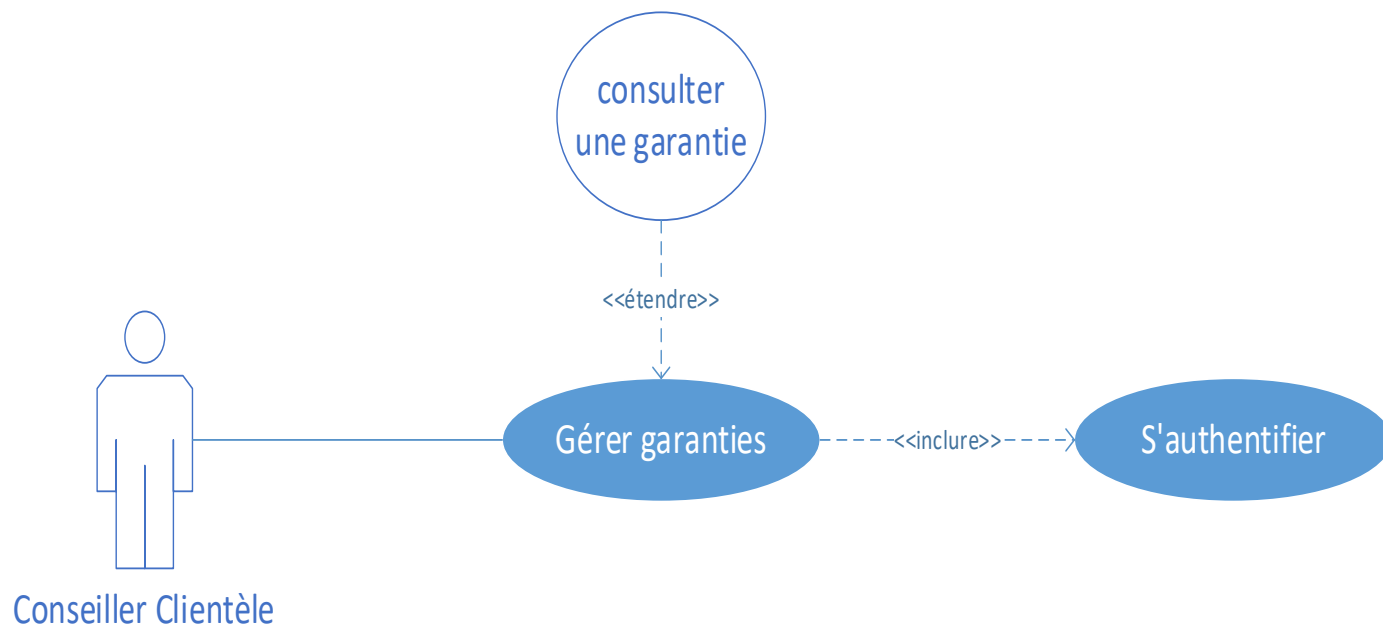


Figure 22 : Diagramme de cas d'utilisation : pour le Conseiller Clientèle

7. Diagramme de cas d'utilisation : pour l'administrateur

L'administrateur a comme rôle principal l'habilitation des utilisateurs et le paramétrage du système.

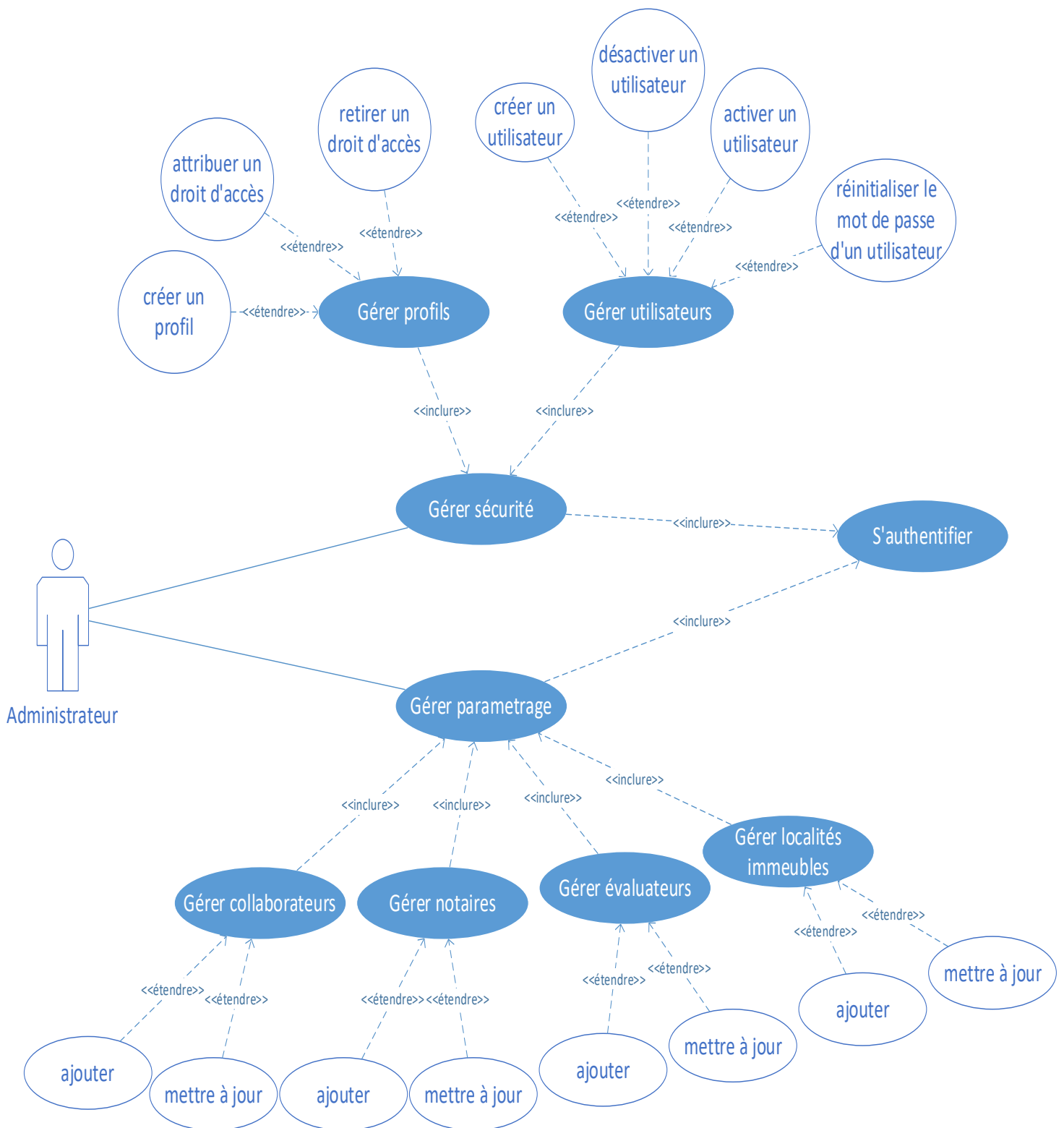


Figure 23 : Diagramme de cas d'utilisation : pour l'Administrateur

II. Diagramme d'activité

Dans le langage UML, un diagramme d'activité fournit une vue du comportement d'un système en décrivant la séquence d'actions d'un processus. Les diagrammes d'activité sont similaires aux organigrammes de traitement de l'information, car ils montrent les flux entre les actions dans une activité. Les diagrammes d'activité peuvent, cependant, aussi montrer les flux parallèles simultanés et les flux de remplacement.

Diagramme d'activité du module de gestion des garanties :

Le diagramme suivant est le représentatif de ce qui se passe au sein du module de gestion des garanties.

Tout d'abord, le Chargé Garanties s'authentifie, si les informations de l'authentification sont correctes alors il a le droit d'accéder à la gestion des garanties. Ainsi, il se trouve face à deux opérations :

On commence par la saisie d'une nouvelle garantie, il remplit le formulaire par les informations nécessaires, le système valide le formulaire, si les données entrées sont correctes alors la garantie est enregistrée. Ensuite, le Chargé Garanties consulte la liste des garanties, il a le choix entre deux opérations :

- annuler une garantie : le Chargé Garanties choisit une garantie parmi celles que le système retourne et il confirme l'annulation. Ainsi, le système procède à une mise à jour de la base de données avec le statut « annulé » ;
- le Chargé Garanties peut mettre à jour les informations d'une garantie, il choisit la garantie concernée et effectue les mises à jour nécessaires puis valide l'opération.

A l'issue de la saisie de garantie par le Chargé Garanties, le Responsable Garanties s'authentifie à son tour, si les informations de l'authentification sont correctes alors il a le droit d'accéder à la gestion des garanties. Ainsi, il se trouve face quatre (04) opérations :

- annuler une garantie : le Responsable Garanties choisit une garantie parmi celles que le système retourne et il confirme l'annulation. Ainsi, le système procède à une mise à jour de la base de données avec le nouveau statut de la garantie : « annulé » ;

- valider une garantie : le Responsable Garanties choisit une garantie parmi celles que le système retourne et il confirme la validation. Ainsi, le système procède à une mise à jour de la base de données avec le nouveau statut de la garantie : « validé » ;
- lier une garantie à un engagement : le Responsable Garanties choisit une garantie parmi celles de la liste que le système retourne. Ensuite, il choisit l'engagement (prêt) parmi ceux du client que le système affiche. Le Responsable Garantie confirme la liaison.
- lever une garantie : le Responsable Garanties choisit une garantie parmi celles que le système retourne et il confirme la levée. Ainsi, le système procède à une mise à jour de la base de données avec une nouvelle situation de la garantie : « échue ».

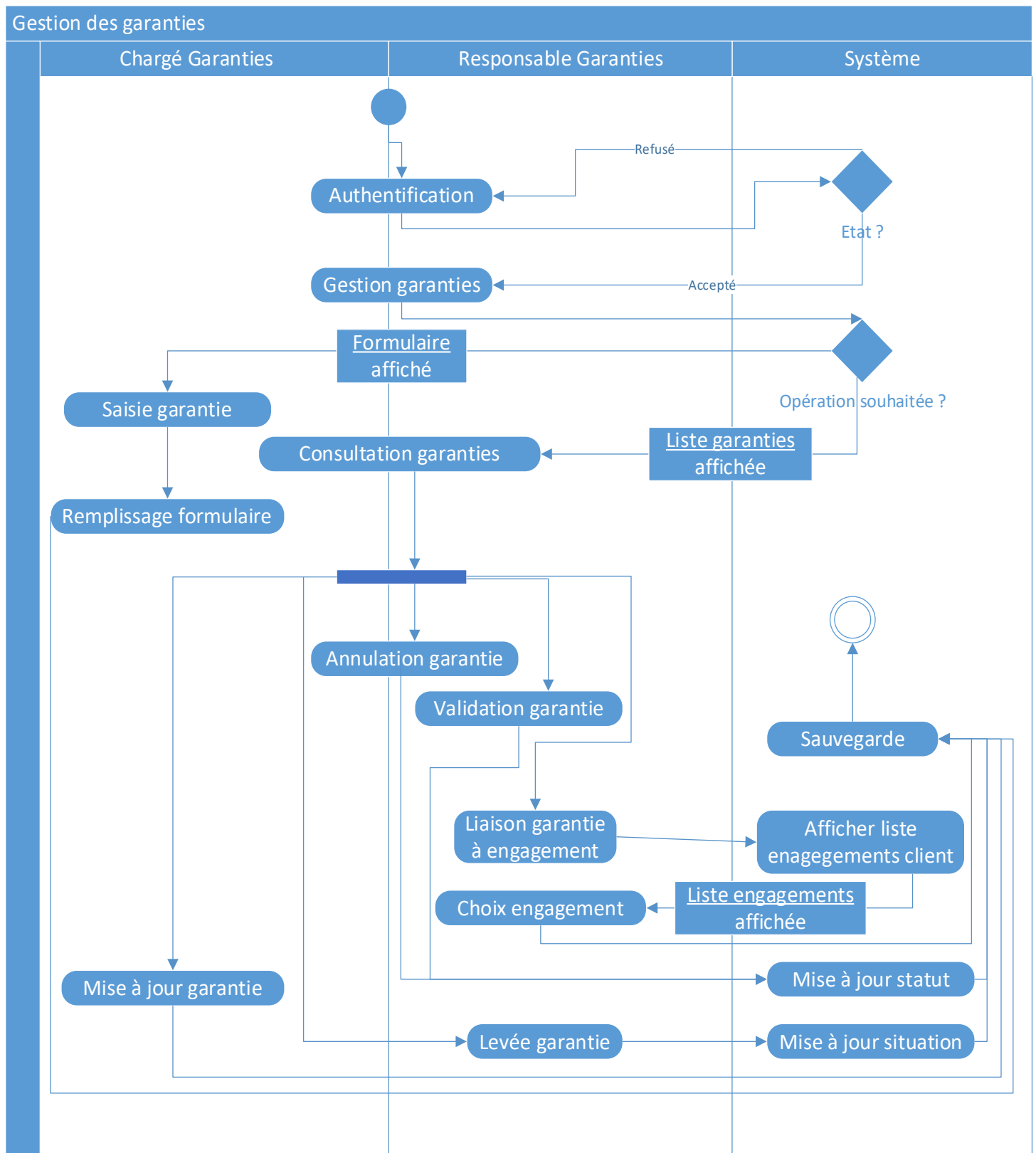


Figure 24 : Diagramme d'activité : Gestion des garanties

III. Diagramme de séquence

Les diagrammes de séquence présentent la coopération entre différents objets. Les objets sont définis et leur coopération est représentée par une séquence de messages entre eux.

Le diagramme de séquence permet de cacher les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification. Dont le but est de décrire comment se déroulent les actions entre les acteurs ou objets.

Après la description des cas d'utilisation, nous allons élaborer le modèle dynamique dans lequel nous allons décrire les scénarios de quelques cas d'utilisation, sous forme de diagrammes de séquence.

1. Diagramme de séquence du scénario : « Authentification »

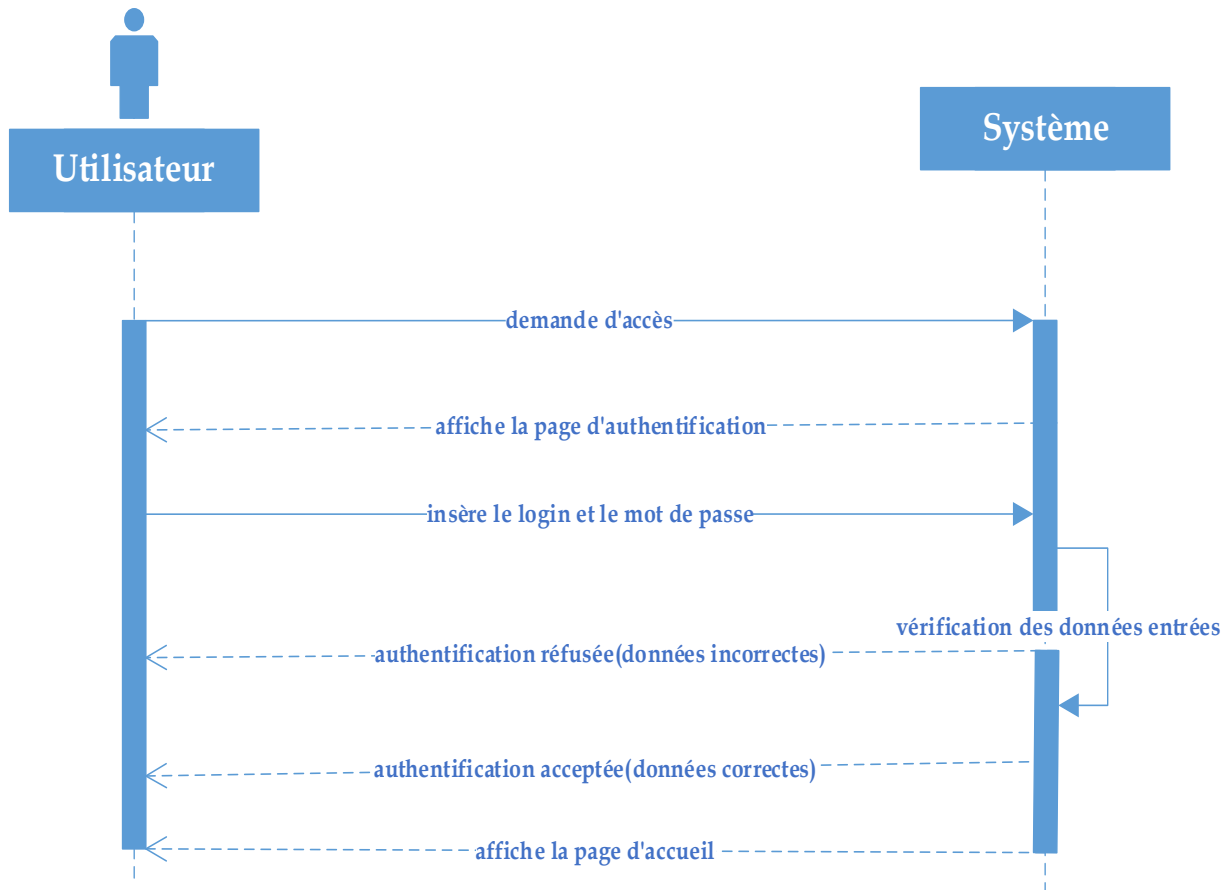


Figure 25 : Diagramme de séquence du scénario « Authentification »

Détails du scénario :

Tout d'abord l'utilisateur insère son login (nom d'utilisateur) et mot de passe, puis le système procède à la vérification si les données saisies avec celles qui existent dans la base de données sont les mêmes. Si les données entrées sont correctes alors le système l'utilisateur redirige vers la page d'accueil sinon cet utilisateur reçoit un message d'erreur.

2. Diagramme de séquence du scénario : « Gestion des garanties »

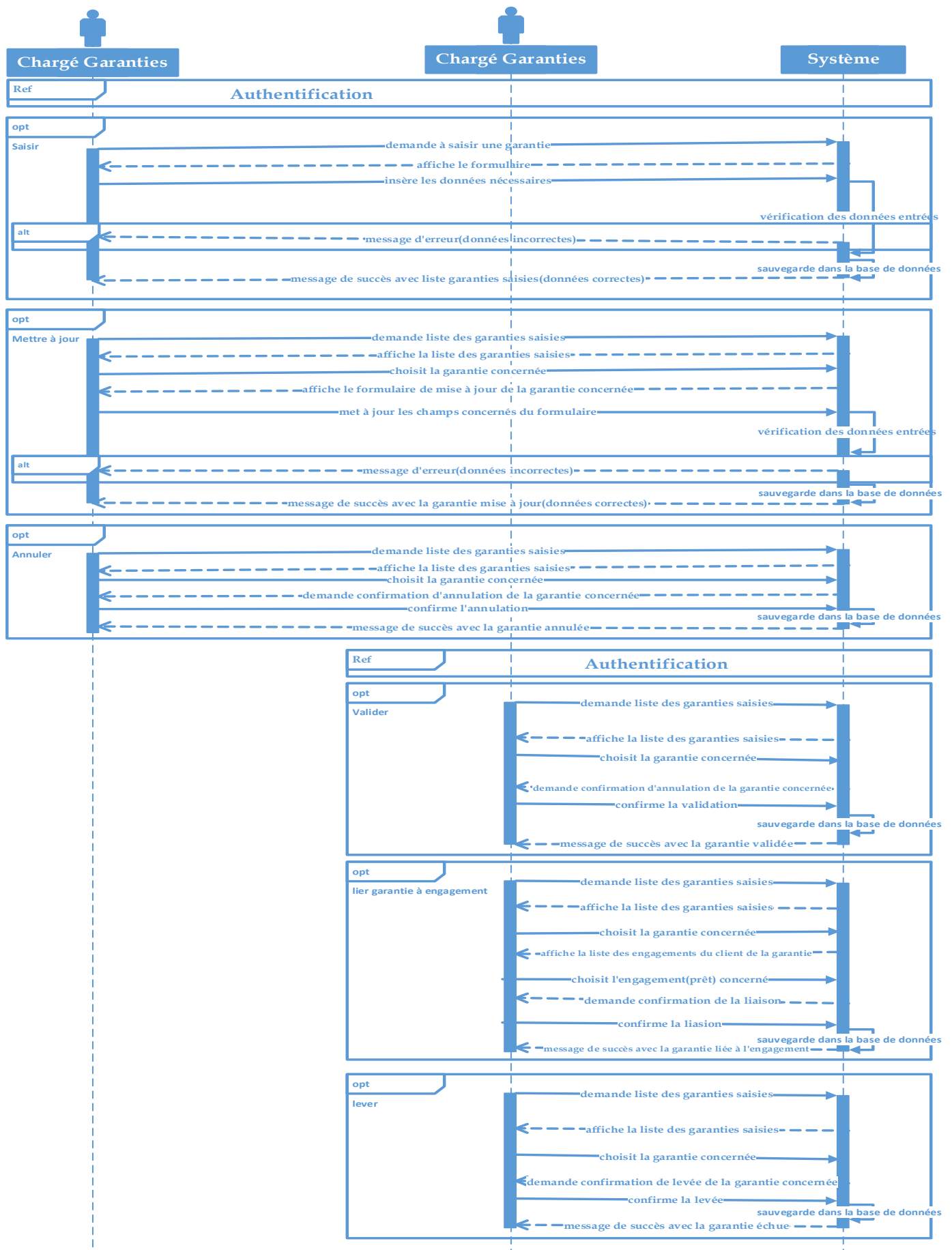


Figure 26 : Diagramme de séquence du scénario : « Gestion garanties »

Détails du scénario :

Le Chargé Garanties ou Responsable Garanties doit premièrement s'authentifier, puis il effectue une demande de consultation de la liste des garanties. Le système affiche la liste des garanties déjà saisies ainsi que la liste des opérations (saisir, annuler, valider, lier garantie à engagement, lever).

Pour le Chargé Garanties :

- **le premier choix** : saisir une garantie, il demande de saisir une garantie, le système affiche le formulaire. L'utilisateur remplit et soumet le formulaire des données. Ainsi, le système procède à la sauvegarde des données dans la base de données si les informations nécessaires sont correctes sinon un message d'erreur s'affiche ;
- **le deuxième choix** : annuler une garantie, il choisit la garantie à annuler et il effectue l'opération d'annulation. Ensuite, il reçoit un message de confirmation par le système. Il confirme et le système annule dans la base de données.

Pour le Responsable Garanties :

- **le premier choix** : annuler une garantie, il choisit la garantie à annuler et il effectue l'opération d'annulation. Ensuite, il reçoit un message de confirmation par le système. Il confirme et le système annule dans la base de données ;
- **le deuxième choix** : valider une garantie, il choisit la garantie à annuler et il effectue l'opération de validation. Ensuite, il reçoit un message de confirmation par le système. Il confirme et le système valide dans la base de données ;
- **le troisième choix** : lier une garantie à un engagement, il choisit la garantie à lier. Le système affiche la liste des engagements du client de la garantie concernée. Le Responsable Garanties choisit l'engagement (prêt) concerné et effectue l'opération de l'opération. Ensuite, il reçoit un message de confirmation par le système. Il confirme et le système lie dans la base de données ;

- **le quatrième choix** : lever une garantie, il choisit la garantie à lever et il effectue l'opération de levée. Ensuite, il reçoit un message de confirmation par le système. Il confirme et le système lève dans la base de données ;

IV. Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

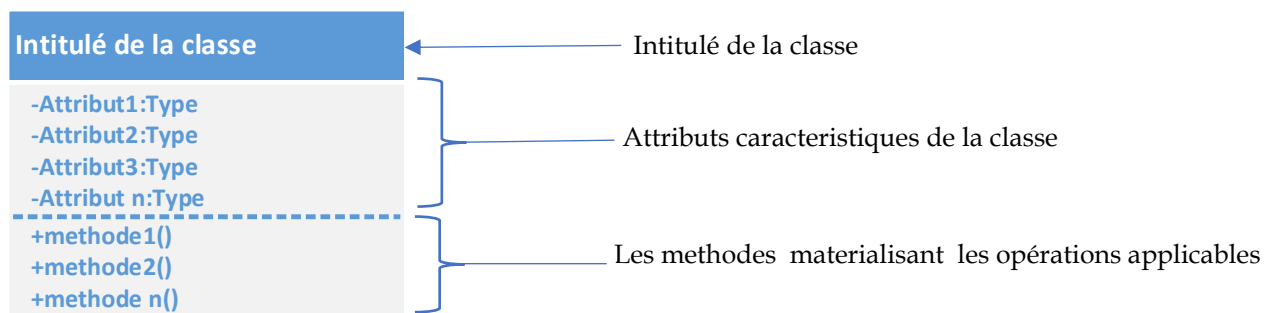


Figure 27 : Le formalisme de la classe

Nous avons également les associations qui sont des relations existantes entre deux ou plusieurs classes et qui décrivent les connexions structurelles entre ces classes. Ces associations possèdent des propriétés telles que :

- **La navigabilité** :

La navigabilité indique s'il est possible de traverser une association. On représente graphiquement la navigabilité par une flèche du côté de la terminaison navigable. Par défaut, une association est navigable dans les deux sens. En exemple⁶⁶ :



Figure 28 : Le formalisme de la navigabilité de l'association

⁶⁶https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme_classes#fig_ex_navigabilite_equivalence_attribut

La terminaison du côté de la « **Classe 1** » n'est pas navigable : cela signifie que les instances (objets) de la « **Classe 2** » ne peuvent pas avoir de liste d'objets du type « **Classe 1** ». Inversement, la terminaison du côté de la classe « **Classe 2** » est navigable : chaque objet « **Classe 1** » contient une liste de produits.

- **L'agrégation :**

Une association simple entre deux classes représente une relation structurelle entre pairs, c'est-à-dire entre deux classes de même niveau conceptuel : aucune des deux n'est plus importante que l'autre. Tandis qu'une agrégation est une association qui représente une relation d'inclusion structurelle ou comportementale d'un élément dans un ensemble. Graphiquement, on ajoute un losange vide (◊) du côté de l'agregat. Par illustration, nous avons ⁶⁷:

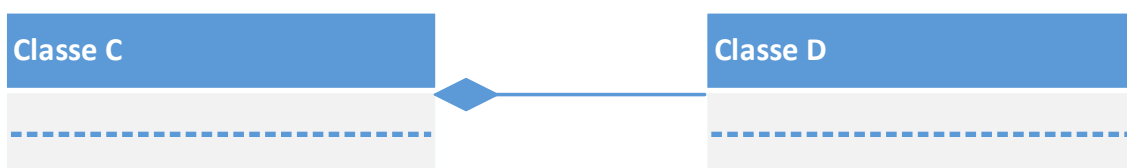


Figure 29 : Le formalise de l'agrégation

Un objet de type « **Classe A** » inclut un ensemble d'objets de type de « **Classe B** ». Mais l'existence d'un objet de la « **Classe B** » ne dépend pas forcément d'un objet de la « **Classe A** ».

- **La composition :**

La composition, également appelée agrégation composite ou agrégation forte, décrit une contenance structurelle entre instances. Ainsi, la destruction de l'objet composite implique la destruction de ses composants. Une instance de la partie appartient toujours à au plus une instance de l'élément composite : la multiplicité du côté composite ne doit pas être supérieure à 1 (1 ou 0..1). Graphiquement, on ajoute un losange plein (◆) du côté de l'agregat. En exemple ⁶⁸:



⁶⁷ https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes#fig_ex_agregation

⁶⁸ https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes#fig_ex_agregation

Figure 30 : Le formalisme de la composition

La disparition ou suppression d'un objet de la « **Classe C** » implique la disparition automatique des objets lui appartenant (ou liés à elle) existant dans la « **Classe D** ».

- **La classe-association :**

Il peut résulter également entre deux (binaire) ou plusieurs (n-aire) classes une association de type classe appelée « classe-association ». Cette classe-association peut posséder des propriétés qui n'appartiennent pas aux classes auxquelles est issue la classe-association. Elle possède les mêmes caractéristiques des associations et des classes. Ce cas d'association se présente lorsque chacune des classes peut avoir plusieurs ou le multiple de l'autre. Une classe-association est caractérisée par un trait discontinu entre la classe et l'association qu'elle représente. Elle se figure de la façon suivante :

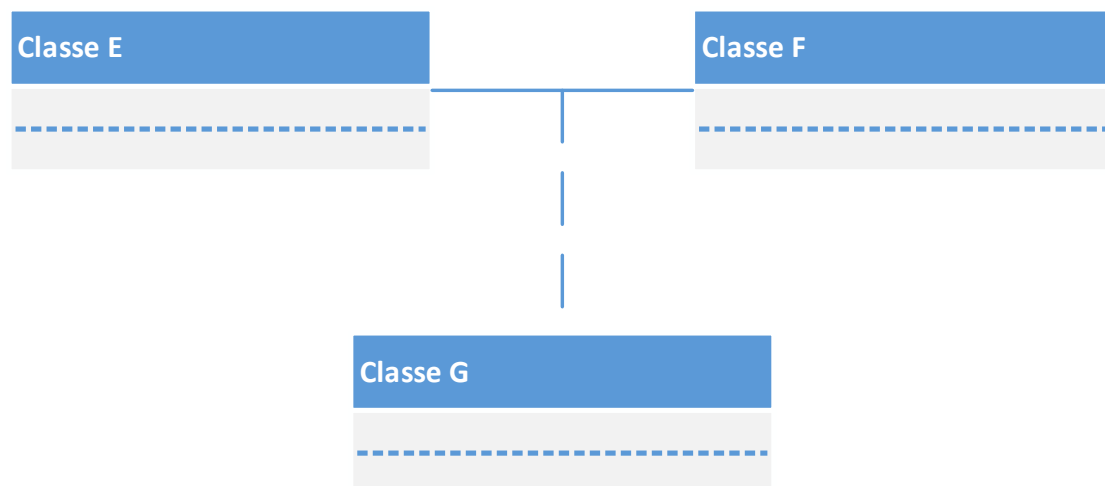


Figure 31 : Le formalisme de la « classe-association »

A partir de l'analyse lors de la partie analyse du projet et des diagrammes de séquence présentée, nous avons extrait un ensemble d'entités et de dépendances, cela a été traduit par UML en un diagramme de classes (figure suivante) qui sera ensuite décrit par un tableau détaillant le rôle de chaque entité.

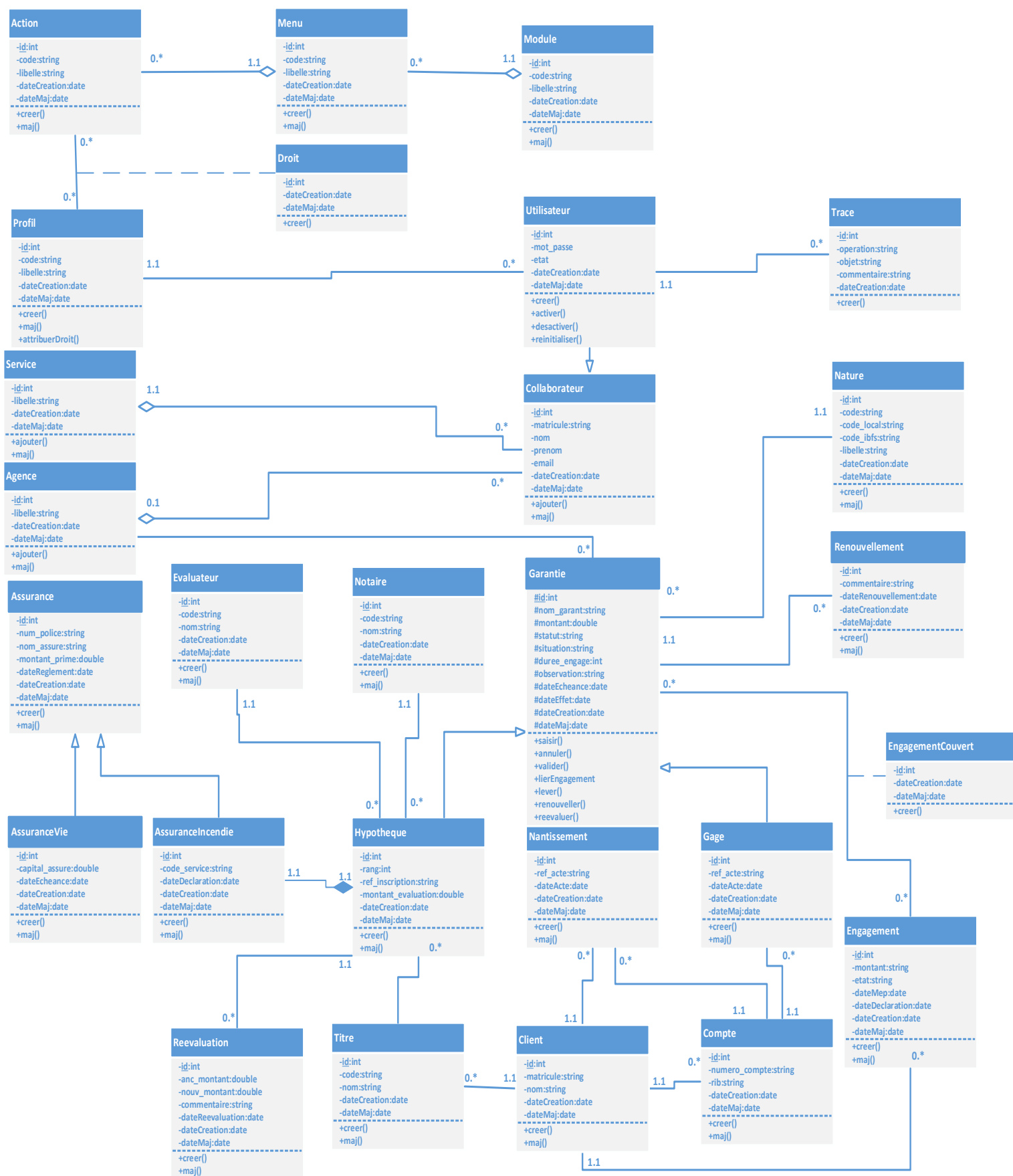


Figure 32 : Diagramme de classes

Classe	Description
Service	Classe contenant les informations sur le service d'un collaborateur.
Agence	Une classe qui représente l'agence du collaborateur et également l'agence de la garantie du client.
Collaborateur	Le collaborateur interne de la structure d'accueil.
Garantie	La classe principale des garanties dont hérite les autres types de garanties.
Nature	La nature d'un type de garantie donné
Renouvellement	Cette classe contient les renouvellements d'une garantie pour une autre échéance.
Hypothèque	Elle contient les informations des garanties de type hypothécaire.
Nantissement	Cette classe regroupe les garanties de type nantissement.
Gage	Cette classe regroupe les garanties de type gage.
Engagement	Les engagements du client qui sont les prêts octroyés par la banque.
EngagementCouvert	Cette classe contient les informations sur la liaison de la garantie à un engagement.
Notaire	Elle contient les informations sur le notaire qui rédige l'acte hypothécaire.
Evaluteur	Elle contient les informations sur l'évaluateur de l'hypothèque.
Reevaluation	Elle contient les informations sur la réévaluation d'une hypothèque après une période donnée.
Titre	Elle contient les informations sur le titre de l'hypothèque du client.
Assurance	Elle est la classe principale des assurances dont hérite les autres types d'assurance.
AssuranceVie	Elle contient les informations sur l'assurance vie d'un client.

AssuranceIncendie	Elle contient les données sur l'assurance incendie d'une hypothèque d'un client.
Client	Elle contient les données sur le client.
Compte	Elle contient les informations sur le compte d'un client.
Utilisateur	Elle contient les données sur un utilisateur du système.
Profil	Elle contient les informations sur le profil d'un utilisateur.
Action	Elle contient les informations sur une action qu'un utilisateur peut exécuter.
Droit	Elle contient les informations sur les actions d'un profil donné.
Menu	Elle contient les informations sur un menu qui regroupe les actions d'un profil.
Module	Elle contient les informations sur un module qui regroupe les menus d'un profil.
Trace	Cette classe historise les informations d'une action menée par un utilisateur dans le système.

Tableau 14 : Dictionnaire de données

Quelques règles de gestion :

- une garantie a un type donné comme un type peut avoir plusieurs garanties ;
- selon le type de garanties on peut avoir plusieurs natures ;
- une garantie est dans une agence donnée ;
- une garantie peut faire l'objet de plusieurs renouvellements ;
- une garantie est liée à un et un seul engagement (si elle couvre un prêt) sinon à tous les engagements du client (si elle couvre un compte courant) ;
- une hypothèque peut être réévaluer plusieurs fois ;
- une hypothèque est évaluée par évaluateur ;
- une hypothèque dont l'acte est rédigé par un notaire ;
- une hypothèque dispose d'une assurance incendie ;
- une assurance a un type donné (assurance vie ou assurance incendie) ;

V. Diagramme déploiement

Dans UML, les diagrammes de déploiement modélisent l'architecture physique d'un système. Les diagrammes de déploiement affichent les relations entre les composants logiciels et matériels du système, d'une part, et la distribution physique du traitement, d'autre part.

Les diagrammes de déploiement, qu'on prépare généralement pendant la phase d'implémentation du développement, présentent la disposition physique du nœud et les composants et autres éléments que les artefacts⁶⁹ implémentent. Les nœuds représentent des périphériques matériels tels que des ordinateurs, des détecteurs et des imprimantes, ainsi que d'autres périphériques qui prennent en charge l'environnement d'exécution d'un système. Les chemins de communication et les relations de déploiement modélisent les connexions dans le système. Le diagramme de déploiement suivant met en évidence les différents niveaux de l'application.

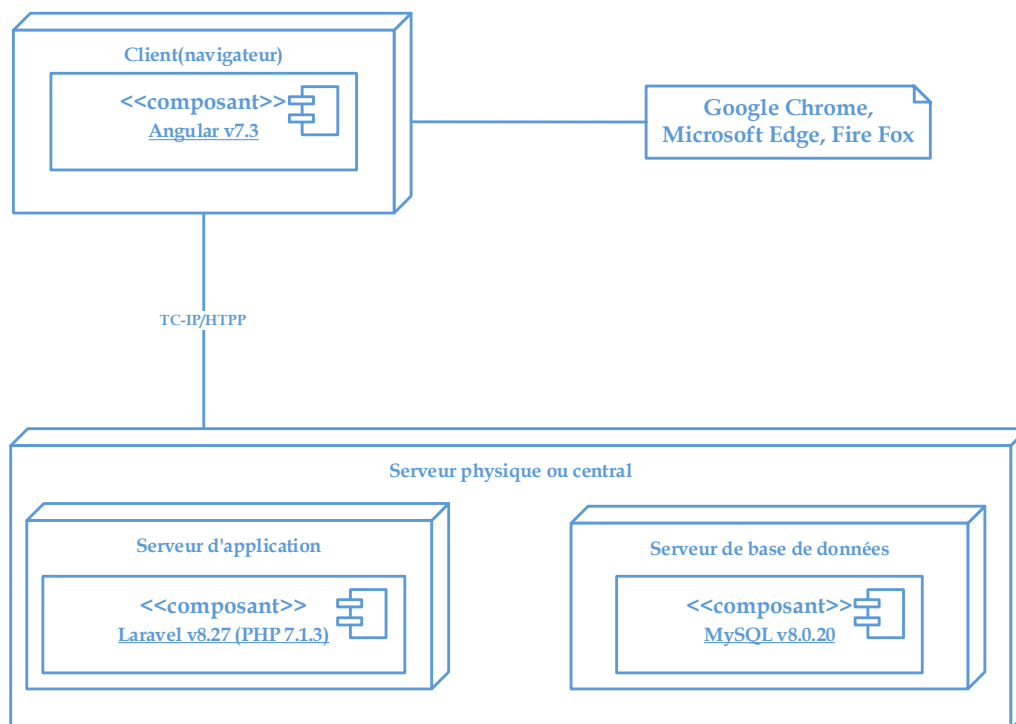


Figure 33 : Diagramme de déploiement

⁶⁹ Un artefact désigne toute sorte d'information créée, produite, modifiée ou utilisée par un homme dans la mise au point d'un « système » informatique.

Conclusion

Ce chapitre a permis de donner une image claire et nette du futur système. Nous avons spécifié les cas d'utilisations de base du système, ses quelques diagrammes d'activité et de séquence, un diagramme de classe et un diagramme déploiement spécifiant l'infrastructure physique du système.

Chapitre V

Implémentation ou mise en œuvre du futur système

Introduction

Au cours de ce chapitre nous allons présenter la procédure déploiement de notre solution, les tests effectués, le coût de mise œuvre et la charte graphique adoptée ainsi que quelques interfaces graphiques de Lafia Garanties que nous présenterons.

I. Déploiement de Lafia Garanties

Le déploiement appelé aussi « livraison » ou « mise en exploitation ». Il regroupe toutes les activités qui mènent à l'installation et mise en marche de l'application développée.

Pour déployer l'application dans le serveur local de notre structure d'accueil, nous avons suivi l'ensembles des étapes suivantes :

- configuration réseau du serveur dans lequel Lafia Garanties sera déployé ;
- installation du serveur d'application ;
- installation du client ;
- configuration de la communication entre le client et le serveur d'application ;
- création des services de démarrage du serveur d'application, du client et du SGBD MySQL ;
- traitement des données existantes afin de générer des scripts SQL ;
- importation de ces données en scripts SQL dans la base de données ;
- tests d'intégrations qui seront détaillés dans la section suivante.

II. Les tests

En informatique, un test désigne une procédure de vérification partielle d'un système. Son objectif principal est d'identifier un nombre maximum de comportements problématiques du logiciel afin d'en augmenter la qualité (si les problèmes identifiés lors des tests sont corrigés). Néanmoins, le test peut aussi avoir pour objectif d'apporter des informations quant à cette qualité afin de permettre la prise de décisions. Le but de cette activité est de tendre vers un produit « zéro défaut ». Au cours du projet, nous avons réalisés trois principaux types de tests⁷⁰ décrits ci-après pour une meilleure qualité de notre application :

1. Les tests unitaires (TU)

But : validation de la conformité de chaque composant logiciel pris unitairement par rapport à sa spécification détaillée.

Quand ? : dès qu'une pièce de code a été codée et compilée correctement.

Type de tests : structurels

Comment ? : sur machine hôte, généralement sans banc de tests

Qui ? : pour les logiciels de faible criticité, elle peut être réalisée par l'équipe de développement (mais pas par le développeur ayant codé la pièce de code).

2. Les tests d'intégrations (TI)

But : validation des sous-systèmes logiciels entre eux :

- tests d'intégration logiciel/logiciel (interface entre composants logiciels) ;
- tests d'intégration logiciel/matériel (interface entre le logiciel et le matériel).

Quand ? : dès qu'un sous-système fonctionnel (module, objet) est entièrement testé unitairement.

Type de tests : des interfaces.

Comment ? : logiciel/logiciel généralement sur machine hôte logiciel/matériel sur machine cible avec un banc de test minimal (simulation des entrées, acquisition des sorties).

⁷⁰ F.X. Fornari, P. Manoury, « Introduction aux tests du logiciel », 2011, page 41-42

Qui ? : toujours par une équipe de tests indépendante de l'équipe de développement.

3. Les tests de validations (TV)

But : vérifier la conformité du logiciel à la spécification du logiciel

Quand ? : dès que l'ensemble des sous-systèmes fonctionnels ont été testé et intégrés.

Type de tests : fonctionnels et de robustesse

Comment ? : toujours réalisés sur machine cible et nécessitent généralement la fabrication d'un banc de tests élaboré.

Qui ? : ces tests sont toujours réalisés par une équipe de tests indépendante de l'équipe de développement.

4. Les tests de recette

Le test de recette (ou d'acceptation ou recette utilisateur) est effectué par des représentants des utilisateurs et des experts métier. Ce test doit confirmer la réponse du logiciel aux exigences du cahier des charges et donner confiance dans le logiciel avant sa mise en production. Le représentant des utilisateurs (expert métier dans notre cas) teste le logiciel en reproduisant les tâches qu'un utilisateur accomplirait lors de l'utilisation normale du logiciel (par exemple dérouler un processus de bout en bout). Les principales techniques de tests de recette suivantes ont été réalisés par notre expert métier :

- tests de type boîte noire⁷¹ et d'exécution basés sur les exigences du cahier des charges ;
- tests de convivialité pour vérifier l'interface graphique du logiciel (est-elle intuitive, cohérente, ergonomique).

III. Politique de sécurité

Une politique de sécurité informatique est un plan d'actions définies pour maintenir un certain niveau de sécurité. La politique de sécurité a pour but de minimiser les risques de dysfonctionnement, d'éviter les incohérences des données, de

⁷¹ Le test de la boîte noire, ou test de la boîte opaque, est utilisé en programmation informatique et en génie logiciel pour tester un programme en vérifiant que les sorties obtenues sont bien celles prévues pour des entrées données.

préserver la confidentialité de la base de données et d'éviter la présence de programmes indésirables sur le réseau.

1. Sécurisation contre les virus

Un virus informatique est un programme malveillant conçu pour se propager à d'autres ordinateurs en s'insérant dans des programmes légitimes appelés « hôtes ». Il se propage principalement à travers des supports de stockage (clés USB, disques durs externes, CD-ROM, disquettes) et peut provoquer une déstabilisation du système.

Pour parer aux désagréments pouvant résulter des virus, nous préconisons l'installation sur chaque poste d'un logiciel antivirus régulièrement mis à jour. Par ailleurs, une restriction de l'utilisation des supports de stockage externes sera effectuée.

2. Confidentialité des données

La confidentialité des données passe par la définition des droits d'accès au système. La définition d'un profil utilisateur au moyen de l'utilisation d'un nom de connexion et d'un mot de passe permettra d'offrir à chaque utilisateur les données et le traitement auxquels il a droit.

Les mots de passe feront l'objet d'un cryptage. Des procédures s'assurant de leur renouvellement régulier et de la limitation du nombre de tentatives de connexion seront également mises en place.

Enfin, l'utilisation d'un journal d'historisation (la traçabilité) permettra de retracer les différentes connexions et actions réalisées afin de responsabiliser les utilisateurs pour toutes les opérations.

3. Sécurisation contre les catastrophes

Afin de préserver le système contre d'éventuelles catastrophes (en exemple l'attaque de la base de données), nous préconisons la mise en place d'une stratégie de sauvegarde de la base.

Dans le choix de cette stratégie nous avons privilégié les techniques de sauvegarde suivantes :

- La sauvegarde complète : cette méthode consiste à sauvegarder l'intégralité des données ;

- La sauvegarde différentielle : elle consiste à sauvegarder toutes les données nouvelles ou modifiées depuis la dernière sauvegarde complète. La restauration passe par une restauration de la sauvegarde complète puis de la dernière sauvegarde différentielle.

La stratégie de sauvegarde mise en place consistera à effectuer une sauvegarde différentielle du système du lundi au vendredi puis une sauvegarde complète le samedi. Ces sauvegardes seront planifiées et exécutées automatiquement. Il est possible également d'exécuter une sauvegarde manuelle à travers une fonctionnalité dédiée de l'application vers un répertoire prédéfini dans les paramètres.

IV. Coût de mise en œuvre

Un des problèmes de la conception logicielle est de disposer d'une bonne estimation des charges afin de mieux prendre en compte la complexité logicielle et de disposer d'une meilleure appréhension de l'estimation. Cela a pour objectif d'aboutir à une limitation des erreurs de budget et des retards de livraison.

1. Méthode de calcul de coûts

Il existe encore très peu de modèles bien structurés pour l'estimation des projets Web. À l'exception de COCOMO II et WebMo. Nous vous présentons dans le tableau ci-après une étude comparative des méthodes d'estimation réalisée par OUALID KTATA⁷² :

	Approche traditionnelle	Approche Web
Processus d'estimation	Utilisation courante de la méthode par analogie (projets similaires) en y ajoutant des enseignements tirés des antérieurs.	Estimations informelles et souvent trop optimistes de l'effort par des développeurs projets seulement.
Estimation de la taille	Les lignes de code et les points de fonction sont utilisés dans les modèles algorithmiques incluant les COTS (package) et les logiciels réutilisés (générant l'équivalent d'autres lignes de codes qui seront ajoutées aux estimations). A défaut on procède par analogie.	La taille est estimée selon une variété d'objets Web (HTML, applets, composantes, etc.) mais il n'y a pas encore de consensus sur une

⁷² OUALID KTATA, « Estimation de projets web : application et analyse de fiabilité des modèles COCOMO II et WebMo », 2007, page 25.

		métrique standard de la taille à adopter pour le Web.
Estimation de l'effort	Utilisation de modèles algorithmiques de type COCOMO II ou modèle des points de fonction. A défaut de modèle, on utilise la méthode par analogie.	Les modèles algorithmiques de type WebMo sont rares. Peu d'historique de projets Web pour l'approche par analogie. Alors l'effort est simplement déterminé en subdivisant le projet en tâches ou livrables.
Estimation de la durée	On utilise soit la relation cubique entre la durée et l'effort ou bien les équations de la durée dans les modèles algorithmiques.	Approche par analogie car la relation cubique durée versus effort n'est pas appropriée selon [Reifer2000].
Estimation de la qualité	Utilisation de métriques simples comme le taux d'erreurs ou recours à des modèles algorithmiques comme COQUALMO.	Utilisation de métriques simples comme le taux d'erreurs mais de nouvelles métriques s'avèrent nécessaires pour estimer la qualité du multimédia.
Calibration des modèles	Au moyen des données réelles contenues dans les projets antérieurs.	Au moyen des données réelles contenues dans les projets antérieurs.
Analyse décisionnelle ('what if') et analyse de risque	Les modèles algorithmiques permettent des analyses d'impact dans les équations de régression ainsi que des analyses de risque en donnant une valeur minimum, une valeur maximum et une valeur vraisemblable (probable).	Peu courante pour les petits projets Web mais indispensable pour les projets d'envergure et les projets transactionnels.

Tableau 15 : Etude comparative des méthodes d'estimation⁷³⁷³ <https://www.geeksforgeeks.org/software-engineering-cocomo-model/>

La méthode COCOMO⁷⁴ est la méthode de calcul de coûts qui a été adoptée à partir de l'étude réalisée ci-dessus, de la fiabilité de ses estimations et de sa meilleure adaptation au domaine informatique. En effet elle prend en compte beaucoup de paramètres en se basant sur la complexité du logiciel à développer. On peut distinguer trois (03) types d'application à développer selon la complexité :

- **Organique (Organic en anglais)** : ce sont des applications simples, de routine, réalisées par une équipe expérimentée ayant l'habitude de travailler ensemble, maîtrisant le langage et l'environnement de développement ;
- **Semi-Détaché (Semi-Detached en anglais)** : ce sont des applications de niveau intermédiaire, ni trop simples, ni trop complexes. L'équipe de développement a déjà réalisé quelques projets ensemble mais n'est pas totalement rodée. Les technologies et le domaine d'application sont un peu flous, mais pas de grosses difficultés.
- **Imbriqué (Embedded en anglais)** : ce sont des applications aux techniques innovantes, à l'organisation complexe et au couplage fort avec beaucoup d'interactions. Les technologies et le domaine sont nouveaux.

Mode	Effort (HM)	TDev(mois)
Organique	$HM=2.4(KLS)^{1.05}$	$TDev=2.5*HM^{0.38}$
Semi-Détachée	$HM=3(KLS)^{1.12}$	$TDev=2.5*HM^{0.35}$
Imbriquée	$HM=3.6(KLS)^{1.20}$	$2.5*HM^{0.32}$

Tableau 16 : Méthode de calcul COCOMO II

KLS : Nombre de lignes estimé

TDev : Temps de développement

HM : Homme-Mois

2. Evaluation du coût de mise en œuvre

A base de la méthode de calcul de coût COCOMO et de l'architecture réseau a adopté, que nous évaluerons le cout total de mise en œuvre de ce projet.

2.1. Evaluation du coût de matériels

Les matériels utilisés pour la mise œuvre du projet figurent dans le tableau suivant :

Désignation	Caractéristiques	Disponibilité	Quantité	Prix Unitaire (FCFA)
Ordinateur	RAM : 4G Disque : 500G Processeur : Dual Core	Existant	4	400 000
Imprimante	Multifonction (Scanne, imprime, photocopie)	Existant	1	200 000
Antivirus	Symantec	Existant	4	50 000
SGBD	MySQL 8.0.20	Non existant	1	Gratuit
			TOTAL	2 000 000

Tableau 17 : Evaluation du coût du matériels

2.2. Coût de développement de l'application

Le mode **imbriqué** est la catégorie de la méthode COCOMO la mieux adaptée à ce scénario du projet. Le nombre de lignes de codes est estimé à **neuf mille cinq cent (9.500) lignes** avec **500 000 CFA** comme salaire moyen d'un professionnel pour la mise en œuvre de ce projet. Ce nombre de lignes de code obtenu tient compte du fait que nous avons utilisés des frameworks qui ont considérablement diminué ce nombre.

Intitulé	Formule	Valeur
Effort à consentir (HM)	$3.6 (9.5)^{1.20}$	53.65 Homme-mois
Temps de développement (TDev)	$2.5 (53.65)^{0.32}$	8.94 mois
Nombre de développeurs	$53.65/8.94$	~ 6 personnes
Coût de développement	Effort x Salaire moyen : $53.65 \times 500\,000$	26 825 000 FCFA

Tableau 18 : Coût de développement de l'application

2.3. Coût de formation des utilisateurs

Nous comptons former environ 4 principaux utilisateurs selon le nombre d'acteurs intervenants dans le système (3 chargés de garanties, 1 administrateur).

Utilisateurs	Nombre	Prix de l'heure	Nombre d'heures	Montant total (FCFA)
Chargés garanties	3	75 000	2	450 000
Administrateur	1	50 000	3	150 000
			Total	600 000

Tableau 19 : Coût de formation des utilisateurs

2.4. Coût de mise en œuvre

Désignation	Coût (CFA)
Evaluation du coût de matériels	2 000 000
Développement	26 825 000
Formation des utilisateurs	600 000
Coût total de mise en œuvre (CTD)	29 425 000

Tableau 20 : Coût de total de mise en œuvre

V. Charte et quelques interfaces graphiques

L'aspect graphique est un élément fondamental de l'ergonomie et de l'expérience utilisateur de toute application possédant des interfaces graphiques.

1. Charte graphique

1.1. Définition

La charte graphique représente l'ensemble des documents qui pose les règles de mise en page et définit l'aspect graphique des documents (internes et externes) d'une société. De préférence établi par le graphiste, ce document dote l'organisation cliente d'un système de contrôle permettant de vérifier chaque élément visuel (style de caractères, logotypes, couleurs) est employé à bon escient afin que tout nouveau document émis par la société participe à la construction de son image. L'essentiel est de rester dans l'univers graphique de l'entreprise.

1.2. Objectifs

La charte Web est un outil opérationnel destiné à :

- faciliter la mise en conformité de la plateforme web avec les lois et standards en matière d'accessibilité ;
- faire connaître les exigences, textes, liens et informations concernant l'accessibilité et la qualité de la plateforme web.

1.3. Contenu de la charte

Ce contenu indique que :

- Il est possible de revenir à la page d'accueil depuis toutes les pages, ce retour se fera via le logo « **Lafia Garanties** » ;
- Il est possible de revenir à l'étape précédente ou suivante via les icones de redirections ;
- Les hyperliens sont visuellement différents du reste du contenu ;
- La mise en place d'un moteur de recherche sur chaque tableau de liste de données. Sa nécessité s'imposera dès que les données de la plateforme prendront du volume ;
- Utilisation des feuilles de styles ;
- Le respect des couleurs (rouge, blanche) de l'entreprise ;
- Les messages de contrôles lors des saisies des formulaires ;
- Notification de l'utilisateur après chaque opération.

2. Quelques interfaces graphiques

2.1. Page d'authentification

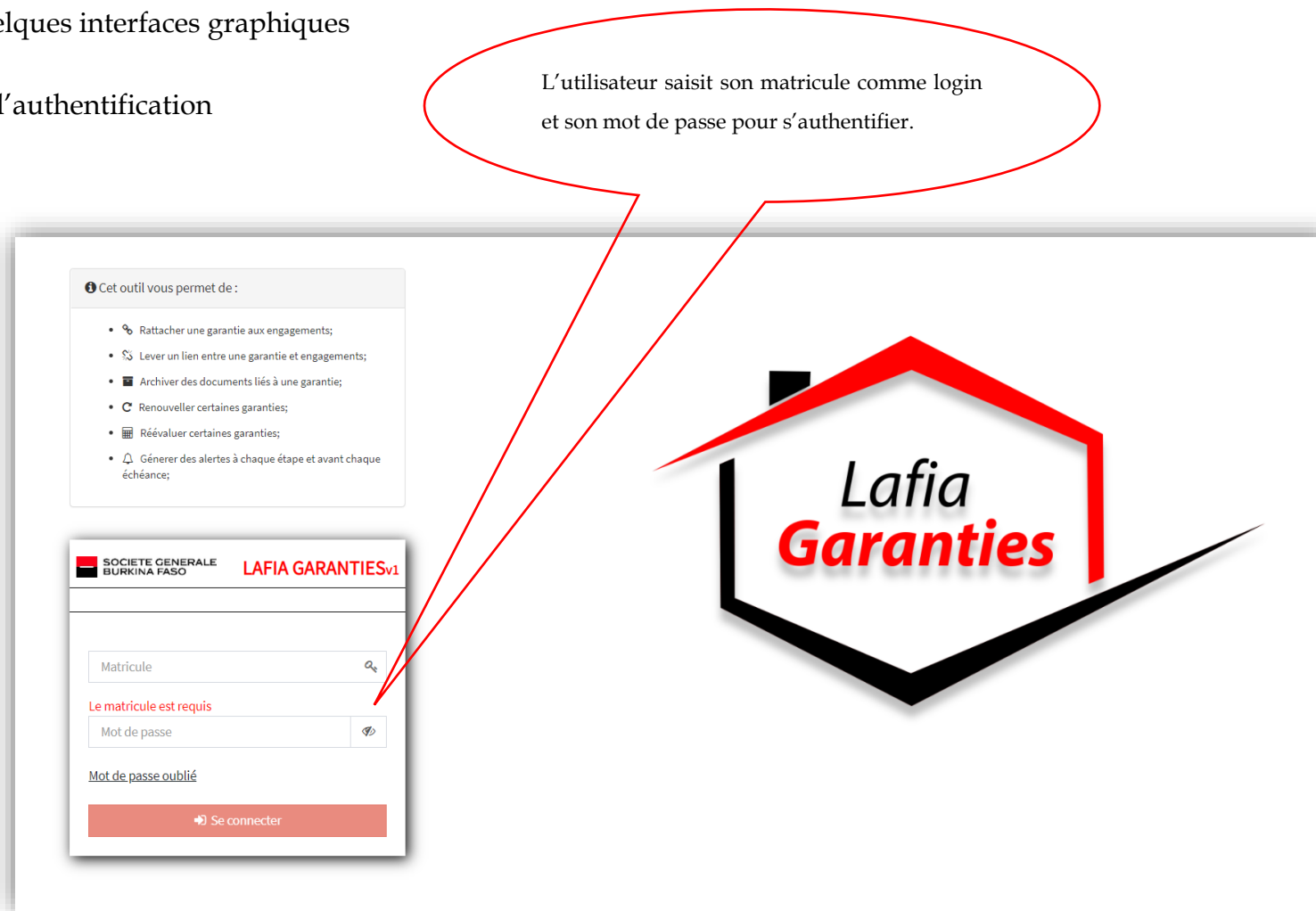
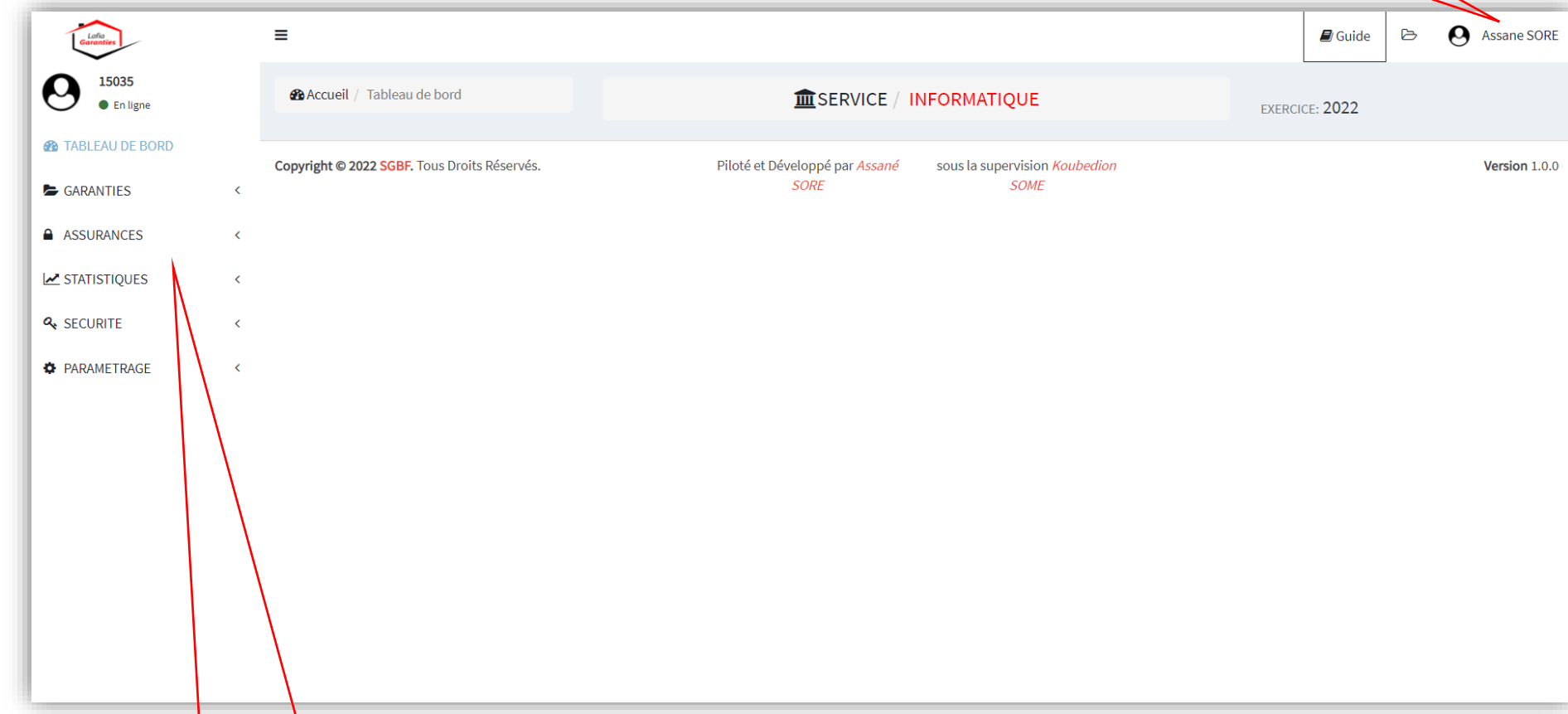


Figure 34 : Page d'authentification

2.2. Page d'accueil

L'utilisateur connecté au système.



Les modules accessibles par l'utilisateur en fonction de son profil.

Figure 35 : Page d'accueil

2.3. Liste des garanties saisies

Le cliquer sur ce bouton affiche le formulaire de saisie de garantie

La liste des garanties saisies sur une période

Le statut et la situation de chaque garantie

Garantie Garanties

Recherche

+ Saisir Extraire

Garanties saisies du: 01 avril 2021 - 30 mars 2022

Rechercher Annuler

26 au total 18 en attente 7 validées 1 annulée 24 non levées 2 levées 23 en cours 3 échues

Liste des garanties saisies

TOTAL INSCRIPTION(en cours): 245 774 515 - TOTAL EVAL/PATRIM/PRET(en cours): 145 622 000

Afficher 10 lignes par page

Rechercher

N° LAFIA	N° DELTA	TYPE	NATURE	ENG COUVERT	CLIENT	ACTION	COMPTE	GARANT/CAUTIONNE	MONT INSCR	MONT EVAL/PATRIM/PRET	DUREE ENG	RANG	SAISIE PAR	AGENCE	STATUT	SITUATION	LEVEE ?	DTE INSC
<input type="checkbox"/> 26-HT-2021	3458	HYPOTHEQUE	Nature 1	PRET	- SOIME KOUBEDION			SOIME Honorine	40 000 000	45 000 000	240	1	(15035)SORE Assane	AGENCE PRINCIPALE	EN ATTENTE	EN COURS	NON	30/05/2021
<input type="checkbox"/> 25-TF-2021		TRANSFERT FUDICIAIRE		PRET	- SAGUIN ADAMA			1-34 KABORE	5 000	0	3		(15035)SORE Assane		VALIDEE	EN COURS	NON	03/04/2021
<input type="checkbox"/> 24-NT-2021		NANTISSEMENT	Nature 3	PRET	- NIKIEMA BERTHE			1-75 PARKOUDA	5 000	0	4		(15035)SORE Assane		VALIDEE	EN COURS	NON	04/02/2021
<input type="checkbox"/> 23-TF-2021		TRANSFERT FUDICIAIRE		PRET	- ZOURE OUMAROU/CAFE KORA			1-08 SANDRINE	5 000	0	3		(15035)SORE Assane		VALIDEE	EN COURS	NON	02/04/2021
<input type="checkbox"/> 22-TF-2021		TRANSFERT FUDICIAIRE		PRET	- CAUET FRANCIS			1-75 SANDRINE	5 000 000	0	3		(15035)SORE Assane		VALIDEE	EN COURS	NON	02/03/2021
<input type="checkbox"/> 21-TF-2021		TRANSFERT FUDICIAIRE		COMPTE COURANT	- ZOURE OUMAROU/CAFE KORA			1-08 SANDRINE	500 000	0			(15035)SORE Assane		VALIDEE	EN COURS	OUI	01/02/2021
<input type="checkbox"/> 20-AC-2021		ACTE CAUTIONNEMENT SOLIDAIRE	Nature 4	PRET	- CAUET FRANCIS			FA-OCHEIL KANAZOE	100 000 000	0	2		(15035)SORE Assane	AGENCE OUAHIGOUYA	EN ATTENTE	EN COURS	NON	02/03/2021
<input type="checkbox"/> 19-AC-2021		ACTE CAUTIONNEMENT SOLIDAIRE	Nature 4	PRET	- SANA ISSA			FA-OCHEIL KANAZOE	3 000	0	3		(15035)SORE Assane		EN ATTENTE	EN COURS	NON	02/03/2021
<input type="checkbox"/> 18-GA-2021		GARANTIE ARIZ		PRET	- SANA ISSA				12 750	50 000	5		(15035)SORE Assane		EN ATTENTE	EN COURS	NON	02/03/2021
<input type="checkbox"/> 17-GS-2021		GARANTIE SOFIB		PRET	107904 - SANA ISSA				765	3 000	3		(15035)SORE Assane		EN ATTENTE	EN COURS	NON	03/02/2021
N° LAFIA	N° DELTA	TYPE	NATURE	ENG COUVERT	CLIENT	ACTION	COMPTE	GARANT/CAUTIONNE	MONT INSCR	MONT EVAL/PATRIM/PRET	DUREE ENG	RANG	AGENCE	SAISIE PAR	STATUT	SITUATION	LEVEE ?	DTE INSC

Prémier Précédent 1 2 3 Suivant Dernier

Figure 36 : Liste des garanties saisies

2.4. Formulaire de saisie d'une garantie

Ce formulaire permet de saisir plusieurs types de garanties. Les champs sont affichés dynamiquement en fonction du type de garantie choisi et certains critères donnés.

The screenshot displays the 'Saisie d'une garantie' (Guarantee Entry) form. The 'Type Garantie' dropdown is open, showing various options. The 'HYPOTHEQUE' option is highlighted. The form includes fields for 'Agence', 'Eng Couvert', 'Montant', 'N° Delta', 'Date Echéance', 'Date Saisie Delta', 'Date Levée Delta', 'Date Mainlevée', and 'Fiche Autre'. A red circle highlights the 'Type Garantie' dropdown, and a red arrow points from it to the text below. The background shows a sidebar with navigation links and a table of existing guarantees.

N° LAFIA	N° DELTA	TYPE	NATURE	ENG COUVERT	CLIENT	ACTION	COMPTE	GARANT/CAUTIONNE	MONT INSCR	MONT EVAL/PATRIM/PRET	DUREE ENG	RANG	AGENCE	SAISIE PAR	STATUT	SITUATION	LEVEE ?	DTE INSC
20-AC-2021		ACTE CAUTIONNEMENT SOLIDAIRE	Nature 4	PRET	107512 - CAJET FRANCIS			FA-OUCHEIL KANAZOE	100 000 000	0	2		(15035)SORE Assane	AGENCE OUAHIOUYA	EN ATTENTE	EN COURS	NON	02/05/2021
19-AC-2021		ACTE CAUTIONNEMENT SOLIDAIRE	Nature 4	PRET	107504 - SANA ISSA			FA-OUCHEIL KANAZOE	3 000	0	3		(15035)SORE Assane		EN ATTENTE	EN COURS	NON	02/03/2021
18-GA-2021		GARANTIE ARIZ		PRET	107504 - SANA ISSA				12 750	50 000	5		(15035)SORE Assane		EN ATTENTE	EN COURS	NON	02/03/2021
17-GS-2021		GARANTIE SOFIGIB		PRET	107504 - SANA ISSA				765	3 000	3		(15035)SORE Assane		EN ATTENTE	EN COURS	NON	03/02/2021

Le choix du type de garantie affichera dynamiquement les autres champs nécessaires.

Le choix du type de garantie affichera dynamiquement les autres champs

Saisie d'une garantie

Type Garantie (*) **HYPOTHEQUE** Nature Garantie (*) **Nature 1** Agence (*) **AGENCE OUAHIGOUYA**

Client (*)

Qualité Garant (*) **Tiers** ☐ **Client** ☐ Personnel ? (*) **Oui** ☐ **Non** ☐

Nom Garant (*) Eng Couvert (*) Durée Eng (*) **(Mois)** Rang (*)

Ref Inscription Montant Inscription (*) Date Inscription

Montant Inscription (*) 0

Montant Evaluation (*) Date Evaluation (*)

Montant Evaluation (*) 0

Notaire (*) Assurance **Taper N° Enregistré(Lafia Assurance)...** Titre (*) **Taper N° Enregistré(Lafia Titre)...**

Ref Acte (*) Date Acte (*) Fiche Acte (*) **Choisir un fichier** Aucun fichier choisi

N° Delta (*) Date Saisie Delta (*) Date Mainlevée

Date Echéance Date Levée Delta Fiche Autre **Choisir un fichier** Aucun fichier choisi

Existence EDR ? (*) **Oui** ☐ **Non** ☐ Existence Exécutoire ? (*) **Oui** ☐ **Non** ☐ Existence Rapport expertise ? (*) **Oui** ☐ **Non** ☐

Observation

Annuler **Enregistrer**

STATUT	SITUATION	LEVÉE ?	DTE INSC
EN ATTENTE	EN COURS	NON	30/05/2021
VALIDÉE	EN COURS	NON	03/04/2021
VALIDÉE	EN COURS	NON	04/02/2021
VALIDÉE	EN COURS	NON	02/04/2021
VALIDÉE	EN COURS	NON	02/03/2021
VALIDÉE	EN COURS	OUI	01/02/2021
EN ATTENTE	EN COURS	NON	02/03/2021
EN ATTENTE	EN COURS	NON	02/03/2021
EN ATTENTE	EN COURS	NON	02/03/2021
EN ATTENTE	EN COURS	NON	03/02/2021

Figure 37 : Formulaire de saisie d'une garantie

VI. Bilan et perspectives

1. Bilan (Post mortem du projet)

1.1. Contraintes et difficultés

Au cours de ce projet, nous avons été confrontés à de nombreuses difficultés :

❖ Pendant la phase de réalisation du projet :

- la principale était la disponibilité des acteurs du métiers lors des entretiens et les différents tests ;
- la complexité des technologies de dernières générations utilisées ;
- nous avons aussi été appelés pour d'autres tâches hors du projet nécessitant des déplacements hors du pays. Ainsi, le projet était en mis en pause pendant un temps donné.

❖ Pendant la phase de déploiement :

- un grand traitement a été fait sur les données des systèmes existants (Excel, Amplitude) pour les restructurer afin d'importer vers le futur système (Lafia Garanties) ;
- un processus de fiabilisation des données remontées a été mis en place avec ses différentes difficultés rencontrées.

1.2. Synthèse

Ce projet a été un succès car nous avons pu réaliser les grandes actions suivantes :

- déployer une solution qui répond aux besoins du métier ;
- rédiger les documents nécessaires (guide d'exploitation, guide utilisateur) ;
- former les utilisateurs à l'utilisation du nouvel outil.

Ce nouvel outil est devenu le quotidien du service « Garanties » et booste sa productivité.

2. Perspectives

Comme perspectives, nous pouvons avoir :

- une nouvelle version de Lafia Garanties qui est à sa première version pour prendre en compte le module de génération des courriers de cautions ;

- un partage d'expérience avec les autres filiales qui sont dans le même besoins que notre structure d'accueil.

3. Planning réel

La figure ci-dessous représente le diagramme de Gantt du planning réel de notre projet après les différents écarts constatés. Comme écarts en comparaison du planning prévisionnel, nous pouvons faire ressortir :

- une augmentation du temps de codage ou de programmation car le développeur été mis sur d'autres tâches hors du projet nécessitant des déplacements hors du pays. Ainsi, cette phase était en mise en pause pendant un temps donné ;
- et le temps des tests de recettes avec l'expert métier a considérablement augmenté dû au fait de son indisponibilité à certains moments.

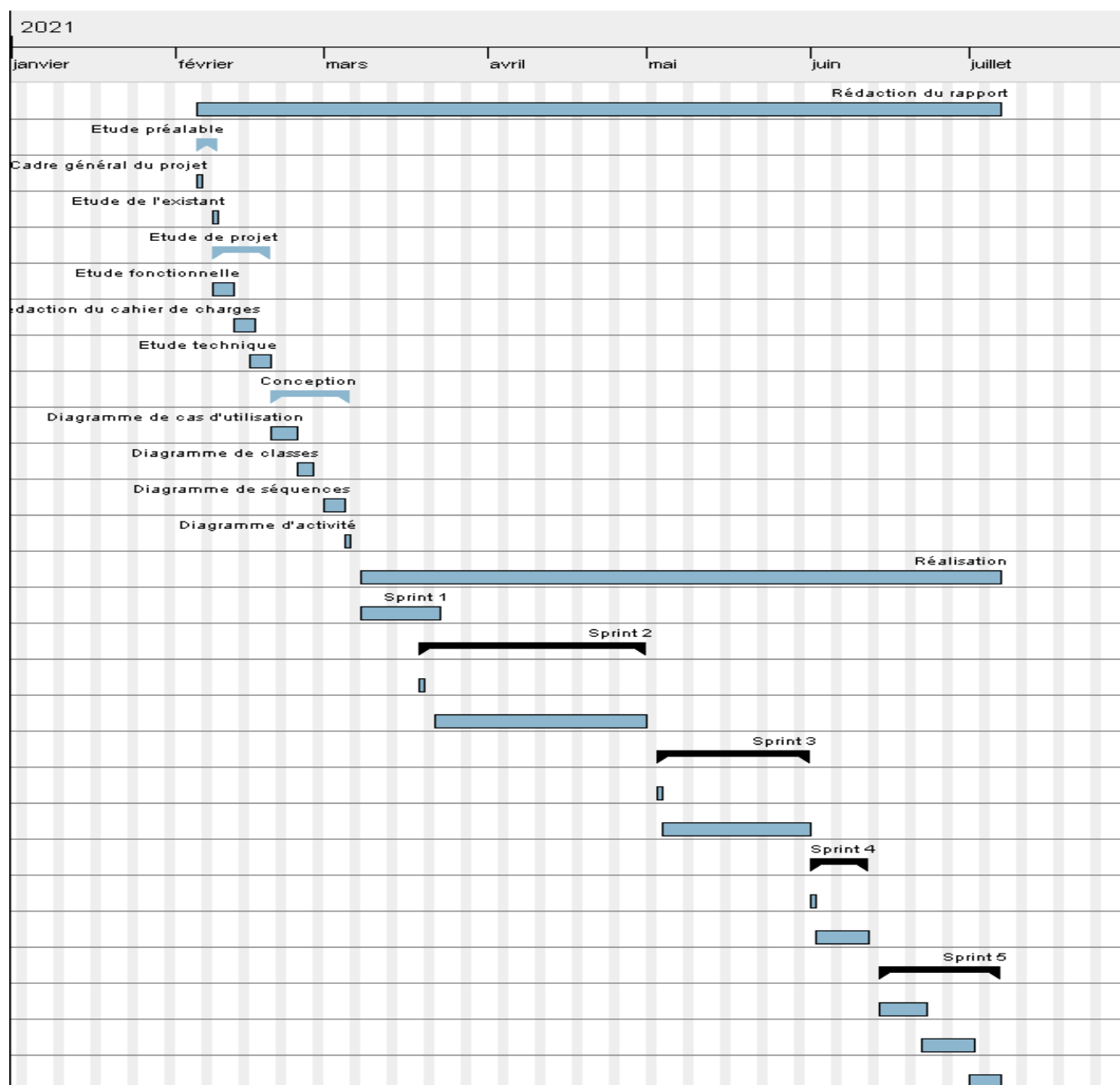


Figure 38 : Diagramme de Gantt du planning réel du projet

Conclusion

Ce chapitre nous a permis de présenter les étapes de déploiement de notre solution, les tests réalisés, une estimation du coût de mise œuvre obtenu à l'issue du projet, la charte graphique adoptée ainsi que quelques interfaces graphiques de Lafia Garanties que nous présenterons. Enfin faire bref bilan des difficultés rencontrées et perspectives à venir.

Conclusion Générale

Notre projet de fin d'étude a été réalisé au profit de notre structure d'accueil qui est la Société Générale Burkina Faso. Durant ce projet la mission suivante nous a été confiée : l'analyse, la conception et la réalisation d'une application de gestion de garanties afin de pallier aux difficultés de gestion ces garanties. Au terme de notre stage, nous avons pu développer une solution (Lafia Garanties) couvrant presque tous les besoins métiers fonctionnels et techniques répondant aux attentes des utilisateurs finaux.

Le choix de la plateforme a été très important car il est le garant de la performance de notre solution. Le découpage en plusieurs a permis de bénéficier d'une accessibilité et d'extensibilité propice à un développement facilement maintenable avec une prise en main facile pour de tiers développeurs.

Pour ce faire nous avons commencé dans un premier temps par comprendre le contexte général du projet et à identifier les différentes exigences de notre futur système dans un second temps. A suivi, une étude détaillée du cahier de charges visant une meilleure compréhension des besoins du métier, et puis le développement des différentes interfaces et objets, dans le respect strict des différentes étapes du cycle de développement Scrum.

Ce stage nous a permis de mettre nos connaissances théoriques à l'épreuve de la pratique dans un environnement d'entreprise. Reste que la nouvelle solution pourrait avec le temps et l'usage être améliorée afin de répondre encore plus aux besoins et exigences de Société Générale Burkina Faso.

Bibliographie

- Pr Abdoulaye OUEDRAOGO, *Guide de rédaction de mémoire de fin d'études*, 2020.
- Charles MOUMOUNI, Article de la revue juridique *Thémis*, *Faculté de droit, Université de Montréal*, 1997, page 781.
- Joseph KAMGA, *Droit et pratique hypothécaires dans l'espace OHADA*, 2020, page 21, 35, 39.
- Mehdi Kettani, *Présentation générale des garanties bancaires conventionnelles usuelles du crédit*, 2017, page 1,2,4.
- Hervé Diaz, *La gestion du risque de crédit bancaire sur les portefeuilles professionnels et particuliers*, 2016, page 5,6.
- F.X. Fornari et P. Manoury, *Introduction aux tests du logiciel*, 2011, page 41,42.
- OUALID KTATA, *Estimation de projets web : application et analyse de fiabilité des modèles COCOMO II et WebMo*, 2007, page 25.
- Mlle. Joairia Lafhal et Mlle. Soumaya. Réalisation et développement d'une plateforme pour la gestion et le suivi des projets des Entreprises, *Mémoire de Fin d'Etudes, Université Abdelmalek Essaadi*, 2016.

Webographie

[https://bpifrance-creation.fr/moment-de-vie/garanties-bancaires-quoi-sagit-il#:~:text=Qu'est%2Dce%20qu',banque%20ou%20par%20l'emprunteur.Annexe\(s\)](https://bpifrance-creation.fr/moment-de-vie/garanties-bancaires-quoi-sagit-il#:~:text=Qu'est%2Dce%20qu',banque%20ou%20par%20l'emprunteur.Annexe(s)), 05/03/2021

<https://www.lesclesdelabanque.com/web/Cdb/Particuliers/Content.nsf/DocumentsByIDWeb/6WGED4?OpenDocument>, 06/03/2021

<https://www.sofinco.fr/lexique-credit/qu-est-ce-qu-un-pre-bancaire.html>, 06/03/2021

<https://www.empruntis.com/financement/lexique/echeance.php>, 07/03/2021

<https://www.observatoiredefranchise.fr/dossier-franchise/suretes-cautions-garanties-nantissements-gages-545.htm>, 08/03/2021

<https://www.journaldunet.fr/patrimoine/guide-des-finances-personnelles/1201671-le-compte-courant/>, 08/03/2021

<https://www.journaldunet.fr/patrimoine/guide-de-l-immobilier/1202139-titre-de-propriete-comment-l-obtenir/>, 15/03/2021

<http://www.blog-gestion-de-projet.com/henry-laurence-gantt-et-son-diagramme/>, 16/04/2021

<http://www.clipindustrie.com/fr/pages/gestion-de-projet>, 17/04/2021

<https://www.newave.be/blog/2017/02/apprendre-facilement-a-utiliser-la-methodologie-scrum/>, 17/04/2021

<http://agilemanifesto.org/iso/fr/principles.html>, 17/04/2021

Annexe

Le capital social de Société Générale Burkina Faso (SGBF)

12 800 000 000 FCFA au 10/03/2017	
Groupe Société Générale (SG siège 47,6% - SGBCI 5%)	52,6%
Groupe CASTEL (FINADEI 6% - BRAKINA 9%)	15%
Etat Burkinabé	15%
Autres actionnaires burkinabé	17,4%

Sommaire	i
Biographie	iv
Dédicace	v
Remerciements	vi
Sigles et Abréviations	vii
Glossaire	ix
Résumé	xi
Liste des Figures	xii
Liste des Tableaux	xiv
Préambule	xv
I. Présentation de l'ESTA	xv
II. Formations de l'ESTA	xv
1. Formations diplômantes	xv
2. Masters	xvi
3. Formation Continue Professionnelle	xvii
4. Certifications en TIC de l'ESTA	xvii
Introduction Générale	1
I. Introduction du sujet	1
II. Contexte et justification (ou problématique)	1
III. Objectifs	1
1. Objectif général	1
2. Objectifs spécifiques	1
IV. Hypothèses (?)	2
1. Hypothèse Principale	2
2. Hypothèses Spécifiques	2

V.	Questions de recherche	2
VI.	Intérêt de l'étude	2
VII.	Méthodologie de collecte des données.....	3
1.	Interview	3
2.	Documents	3
3.	Recherche bibliographique.....	3
4.	Traitements des résultats	3
VIII.	Structure d'accueil	4
1.	Présentation	4
2.	Historique	4
3.	Les valeurs du groupe.....	5
4.	Le capital social	5
5.	L'organigramme	6
Chapitre I : Revue de littérature		7
Introduction.....		8
I.	Les garanties bancaires.....	8
II.	Le risque de crédit.....	10
Conclusion		10
Chapitre II : Etude fonctionnelle du système		11
Introduction.....		12
I.	Critique de l'existant.....	12
1.	Amplitude Delta	12
2.	Microsoft Excel.....	13
II.	Les fonctionnalités identifiées	15
1.	Le module de gestion des garanties.....	15
2.	Le module de gestion des assurances	16

3. Le module de production de statistiques	17
4. Le module de gestion de la sécurité	17
4.1. La gestion des utilisateurs	17
4.2. La gestion des profils ou des rôles	17
4.3. La gestion de l'historique ou la traçabilité	17
4.4. La gestion des sauvegardes.....	18
5. Le module de paramétrage	18
5.1. La gestion des agences	18
5.2. La gestion des services	18
5.3. La gestion des fonctions.....	18
5.4. La gestion des collaborateurs.....	18
5.5. La gestion des institutions garantes.....	18
5.6. La gestion des devises.....	19
5.7. La gestion des types de concours	19
5.8. La gestion des types de garanties.....	19
5.9. La gestion des natures de garanties	19
5.10. La gestion des évaluateurs	19
5.11. La gestion des notaires.....	19
5.12. La gestion des originaux titres.....	20
5.13. La gestion des localités immeubles	20
5.14. La gestion des types d'assurances.....	20
5.15. La gestion des paramètres généraux.....	20
III. Démarche de développement	20
1. Tableau comparatif des différents cycles de vie	21
2. Méthode de Scrum.....	24
3. Approche Agile	24
4. Pourquoi la méthode Scrum ?.....	25

5. Fonctionnement de Scrum.....	25
IV. Planification du projet	28
1. Planification.....	28
2. Diagramme de Gantt.....	29
Conclusion	30
Chapitre III : Etude technique du système.....	31
Introduction.....	32
I. Architecture logicielle.....	32
1. Couche Présentation.....	32
2. Couche ViewModel	32
3. Couche Service	33
4. Couche d'accès aux données.....	33
5. Couche stockage	33
6. Couche sécurité	34
II. Architecture Réseau	34
III. Méthodologie de développement : Démarche MVC	34
IV. La plateforme PHP	36
V. Frameworks	38
1. Laravel.....	38
2. Angular	40
3. HTML5/CSS3.....	41
4. Bootstrap	43
VI. Environnement de travail	43
1. Environnement de développement.....	43
2. Outil de modélisation.....	45
3. Système de Gestion de Base de Données	46

Conclusion	47
Chapitre IV : Conception détaillée du futur système	48
Introduction.....	49
I. Diagramme de cas d'utilisation	49
1. Identification des acteurs.....	50
2. Identification des cas d'utilisations.....	51
3. Diagramme de cas d'utilisation principal ou de base	58
4. Diagramme de cas d'utilisation : pour le Chargé Garanties	60
5. Diagramme de cas d'utilisation : pour le Responsable Garanties	61
6. Diagramme de cas d'utilisation : pour le Conseiller Clientèle.....	62
7. Diagramme de cas d'utilisation : pour l'administrateur.....	63
II. Diagramme d'activité	64
III. Diagramme de séquence	67
1. Diagramme de séquence du scénario : « Authentification »	68
2. Diagramme de séquence du scénario : « Gestion des garanties » .	68
IV. Diagramme de classes	71
V. Diagramme déploiement	77
Conclusion	78
Chapitre V : Implémentation ou mise en œuvre du futur système.....	79
Introduction.....	80
I. Déploiement de Lafia Garanties	80
II. Les tests.....	81
1. Les tests unitaires (TU)	81
2. Les tests d'intégrations (TI)	81
3. Les tests de validations (TV)	82
4. Les tests de recette	82

III.	Politique de sécurité.....	82
1.	Sécurisation contre les virus.....	83
2.	Confidentialité des données.....	83
3.	Sécurisation contre les catastrophes.....	83
IV.	Coût de mise en œuvre	84
1.	Méthode de calcul de coûts	84
2.	Evaluation du coût de mise en œuvre	86
2.1.	Evaluation du coût de matériels	87
2.2.	Coût de développement de l'application	87
2.3.	Coût de formation des utilisateurs	88
2.4.	Coût de mise en œuvre	88
V.	Charte et quelques interfaces graphiques	88
1.	Charte graphique	88
1.1.	Définition.....	88
1.2.	Objectifs	89
1.3.	Contenu de la charte.....	89
2.	Quelques interfaces graphiques	90
2.1.	Page d'authentification	90
2.2.	Page d'accueil	91
2.3.	Liste des garanties saisies	92
2.4.	Formulaire de saisie d'une garantie	93
VI.	Bilan et perspectives	95
1.	Bilan (Post mortem du projet).....	95
1.1.	Contraintes et difficultés	95
1.2.	Synthèse.....	95
2.	Perspectives	95
3.	Planning réel.....	96

Conclusion	97
Conclusion Générale	98
Bibliographie	99
Webographie.....	100
Annexe.....	101
Table des matières	102