I. PAGE DE GARDE

II. DEDICACES

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui quels que soient les termes utilisés, je n'arriverais jamais à leur exprimer mon amour sincère.

- ❖ A mes chers parents M VOUTI-KOTOR Kossi et Mme KARBOU Kossiwa, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,
- A ma chère sœur pour ses encouragements permanents, et son soutien moral,
- ❖ A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

III. REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord j'adresse mes remerciements à Dieu tout puissant, le père de toute chose.

J'adresse mes remerciements :

- ❖ A mes parents qui m'ont beaucoup aidé pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.
- ❖ A M. AGBETI Kodjo, Représentant Résident de l'IAI-TOGO, pour son accompagnement, ses mots d'encouragement et sa lutte acharnée pour garantir à tous les étudiants de l'IAI une meilleure formation ;
- ❖ A M. AMEYIKPO Kossi Nicolas, Directeur des Affaires Académiques et de la Scolarité pour la notion de discipline inculquée au cours de ces trois années ;
- ❖ A mon superviseur M. N'SOUGAN Folly Woèdè pour ses conseils et sa supervision ;
- ❖ A mon maitre de stage M. BILANTE Addo pour son aide, ses conseils, son encadrement et le partage d'expérience dont il a fait preuve durant la durée de cette formation.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidés mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches. Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de ma démarche.

IV. SOMMAIRE

I.	PAGE DE GARDE	
II.	DEDICACES	i
III.	REMERCIEMENTS	ii
IV.	SOMMAIRE	iv
V.	RESUME	V
VI.	GLOSSAIRE	. vi
VII.	LISTE DES FIGURES	vii
VIII.	LISTE DES TABLEAUX	×
IX.	LISTE DES PARTICIPANTS	x
INT	RODUCTION	1
PAR	TIE 1 : CAHIER DES CHARGES	2
l.	PRESENTATIONS	3
II.	THEME DU STAGE	8
III.	ETUDE DE L'EXISTANT	10
IV.	CRITIQUE DE L'EXISTANT	12
V.	PROPOSITIONS ET CHOIX DE SOLUTIONS	13
VI.	PLANNING PREVISIONNEL DE REALISATION	19
PAR	TIE 2 : ANALYSE ET CONCEPTION	20
I.	CHOIX DE LA METHODE D'ANALYSE ET JUSTIFICATION	21
II.	CHOIX DE L'OUTIL DE MODELISATION ET JUSTIFICATION	28
III.	ETUDE DETAILLEE DE LA SOLUTION	29
PAR	RTIE 3 : REALISATION ET MISE EN OEUVRE	52
l.	MATERIELS ET LOGIIELS UTILISES	53
II.	ARCHITECTURE MATERIEL ET LOGICIEL DE L'APPLICATION	63
III.	SECURITE DE L'APPLICATION	65
IV.	MISE EN PLACE DE LA BASE DE DONNEES	66
PAR	TTIE 4: EXPLOITATION (Guide d'exploitation)	70
l.	CONFIGURATION MATERIELLE ET LOGICIELLE	72
II.	DEPLOIEMENT ET SUIVI	73
III.	MAINTENANCE	74
l.	DESCRIPTION TEXTUELLE DU LOGICIEL	76
II.	PLAN DE NAVIGATION	76
III.	PRESENTATION DES DIFFERENTS INTERFACES DE L'APPLICATION	77
IV	PRESENTATION DE OLIFI OLIFS ETATS	83

PLATEFORME DIGITALE DE COLLECTE ET D'ANALYSE D'INFORMATION DES ACCIDENTS CONSTATES SUR UN TERRITOIRE (CAS DU TOGO)

CON	ICLUSION	85
V.	BIBLIOGRAPHIE INDICATIVE	I
VI.	WEBOGRAPHIE INDICATIVE	
VII.	NOTES DE COURS	. 111
VIII.	ANCIENS MEMOIRES CONSULTES	. IV
ΙX	DOCUMENTS ANNEXES	\/

V. RESUME

En vue de l'obtention du diplôme de licence, l'Institut Africaine d'Informatique, Représentation du Togo (IAI-TOGO) a mise en place des stages pratiques obligatoires dans le cadre professionnel pour permettre aux étudiants de bénéficier de l'expérience du travail en entreprise. Ce document est le mémoire d'étude de fin de formation dans la filière Génie Logiciel et Système d'Information. Sa rédaction a été faite sur une période de trois (3) mois effectués au sein de la structure informatique ODDA Technology.

VI. GLOSSAIRE

Termes Définitions

Générique	Générique signifie qu'il est nécessaire d'adapter UP au contexte du projet, de l'équipe, du domaine et/ou de l'organisation
Itératif et incrémental	La méthode est itérative dans le sens où elle propose de faire des itérations lors de ses différentes phases, ceci garantit que le modèle construit à chaque phase ou étape soit affiné et amélioré. Chaque itération peut servir aussi à ajouter de nouveaux incréments.
Index TIOBE	L'index TIOBE mesure la popularité des langages de programmation en se basant sur le nombre de pages web retournées par les principaux moteurs de recherche lorsqu'on leur soumet le nom du langage de programmation. Il est mis à jour une fois par mois et donne l'historique depuis 2002.
Micro-service	Micro Service est une architecture qui permet aux développeurs de développer et de déployer des services de manière indépendante. Chaque service en cours d'exécution a son propre processus, ce qui permet d'obtenir un modèle léger pour soutenir les applications commerciales.
REST	Representational State Transfer : Est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web
SQL	Structured Query Language : Est le principal langage des systèmes de base de données.
CLI (Angular CLI)	Command-Line Interface

VII. LISTE DES FIGURES

Figure 1: Localisation de l'IAI-TOGO	4
Figure 2: Localisation du centre d'accueil	7
Figure 3: Fiche d'enregistrement de l'accident	10
Figure 4: Fiche d'enregistrement des véhicules impliquées dans l'accident	11
Figure 5: Planning prévisionnel de réalisation et diagramme de Gantt	19
Figure 6: Hiérarchie des diagrammes UML	24
Figure 7: Le système d'information soumis à deux types de contraintes	25
Figure 8: Le processus de développement en Y	26
Figure 9: Logo de Microsoft Visio	28
Figure 10: Diagramme de cas d'utilisation de l'administrateur	31
Figure 11: Diagramme de cas d'utilisation de l'agent de police	32
Figure 12: Diagramme de cas d'utilisation de l'observateur	33
Figure 13: Diagramme de séquence du cas « S'authentifier »	38
Figure 14: Diagramme de séquence « Enregistrer un accident »	39
Figure 15: Diagramme de séquence « Enregistrer une personne »	40
Figure 16: Diagramme de séquence détaillé du cas « S'authentifier »	. 42
Figure 17: Diagramme de séquence détaillé du cas « Enregistrer un accident »	43
Figure 18: Diagramme d'activité du cas « S'authentifier »	45
Figure 19: Diagramme d'activité du cas « Enregistrer un accident »	. 46
Figure 20: Diagramme d'activité du cas « Enregistrer une personne »	47
Figure 21: Diagramme de classe du système	49
Figure 22: Diagramme de paquetage du système	. 50
Figure 23: Diagramme de déploiement du système	51
Figure 24: Logo de Spring Framework	55
Figure 25: Logo de Spring boot	. 56
Figure 26: Logo de PostgreSQL	. 57
Figure 27: Logo de Twilio	58
Figure 28: Logo de Dart	. 58
Figure 29: Logo de Flutter	. 59
Figure 30: Logo d'Angular	59

PLATEFORME DIGITALE DE COLLECTE ET D'ANALYSE D'INFORMATION DES ACCIDENTS CONSTATES SUR UN TERRITOIRE (CAS DU TOGO)

Figure 31: Logo de Bootstrap	60
Figure 32: Logo de GitHub	61
Figure 33: Logo de IntelliJ IDEA	62
Figure 34: Architecture globale de l'application	63
Figure 35: Architecture MVC	64
Figure 36: Plan de navigation	76
Figure 37: Page de connexion	77
Figure 38: Page de gestion des utilisateurs	78
Figure 39: Page de visualisation des coordonnées	78
Figure 40: Page de détails d'un accident	79
Figure 41: Page de visualisation des statistiques	79

VIII. LISTE DES TABLEAUX

Tableau 1: Participants au projet	X
Tableau 2: Parc informatique	6
Tableau 3: Tableau de coût d'achat logiciel	15
Tableau 4: Coût d'achat matériel	15
Tableau 5: Coût de formation	16
Tableau 6: Coût total	16
Tableau 7: Coût de développement	17
Tableau 8: Coût d'hébergement	17
Tableau 9: Coût de formation	17
Tableau 10: Coût total	18
Tableau 11: Listes des diagrammes UML	23
Tableau 12: Liste des cas d'utilisation	30
Tableau 13: Choix du matériel	53

IX. LISTE DES PARTICIPANTS

Ci-dessous la liste des participants au projet :

Tableau 1: Participants au projet

Noms	Fonctions	Rôles
M. VOUTI-KOTOR Nyuiwola	Etudiant en 3 ^{ème} année de Génie Logiciel et Système d'Information à l'IAI-TOGO	Maître d'œuvre
M. BILANTE Addo	DSI de ODDA Technology	Maitre de stage
M. N'SOUGAN Folly Woèdè	Chargé de cours à l'IAI-TOGO	Superviseur

INTRODUCTION

Depuis déjà quelques années, les accidents de la route constatés sur le territoire togolais ont considérablement augmenté. Cette augmentation elle-même dû à l'accroissement des véhicules mises en circulation.

Pour avoir un ordre d'idée de l'évolution du nombre d'accidents ces dernières années, au deuxième semestre de l'année deux mille vingt et un (2ème semestre 2021) il y a eu trois cent trente-quatre (334) morts liés aux accidents de la route, d'après le site www.republiquetogolaise.com, et environ six cent morts (600) tout au long de l'année. Ce nombre est nettement plus qu'en 2020 où l'on ne comptait que cinq cent cinquante-six (556) morts.

Face à cette difficulté, la nécessité de garder l'œil sur les chiffres exacts est devenue très importante. Afin de mieux gérer la situation, le ministère de la sécurité a décidé de se tourner vers des moyens beaucoup plus modernes.

Dans le cadre d'un stage de licence informatique qui se déroule du 16 Mai 2022 au 13 Août 2022 sein de la société ODDA Technology. Notre objectif est de réaliser une application en trois parties qui consistera en une gestion efficace des accidents de la route constatés sur le territoire. Le thème de notre stage est « Plateforme de collecte et d'analyse des informations sur les accidents constatés sur un territoire (Cas du Togo). ».

La première partie de ce rapport introduit le cadre d'étude et du stage suivi de la présentation du sujet. La deuxième et la troisième partie présente l'analyse, la conception et la réalisation qui va conduire le développeur soit à créer une application sur mesure, soit à choisir une solution préexistante. Et enfin la quatrième et la cinquième partie qui décrit le logiciel accompagné de la documentation.

PARTIE 1: CAHIER DES CHARGES

I. PRESENTATIONS

A. Brève présentation de l'IAI-TOGO

1. Aspect formel

La convention portant création de l'Institut Africain d'Informatique et les statuts y afférent ont été signés le 29 janvier 1972 en république du Tchad à Fort Lamy connue aujourd'hui sous l'appellation de N'Dajmena et actuelle capitale du Tchad. L'accord de siège entre l'IAI et le GABON (siège officiel de l'institut) a été signé en Janvier 1975. Il est par conséquent un établissement Inter-Etats d'Enseignement Supérieur. En application de la décision du conseil d'Administration de délocaliser l'IAI, la Représentation du TOGO a ouvert ses portes le 24 octobre 2002. L'accord d'établissement entre la République Togolaise et l'Institut Africaine d'Informatique a été signé le 12 mai 2006. L'IAI offre actuellement deux diplômes à savoir : le diplôme d'Ingénieur des Travaux Informatiques et celui de la Licence professionnelle Informatique. Trois filières sont essentiellement disponibles :

- ✓ Le Génie Logiciel;
- ✓ Le Multimédia Web et Infographie et
- ✓ Le Système et Réseaux.

Les étudiants diplômés ont la possibilité de poursuivre leurs études supérieures au siège au GABON ou dans les universités européennes en partenariat direct avec l'institut comme l'Université de Technologie de Belfort-Montbéliard (UTBM) et l'Université de Technologie de Troyes en France.

2. Aspect géographique

L'IAI-TOGO est situé à Kodjoviakopé, derrière le bâtiment de l'UTB circulaire et dans les locaux du Centre National des Etudes et Traitements Informatiques (CENETI).



Figure 1: Localisation de l'IAI-TOGO

B. Présentation de la structure ODDA Technology

Notre stage clôturant notre cycle d'Ingénieur de Travaux Informatique a été effectué au sein de la société ODDA Technology. ODDA Technology est un intégrateur de solutions informatiques et de télécommunication. Elle est considérée comme une Société de Services en Ingénierie Informatique.

1. Statut

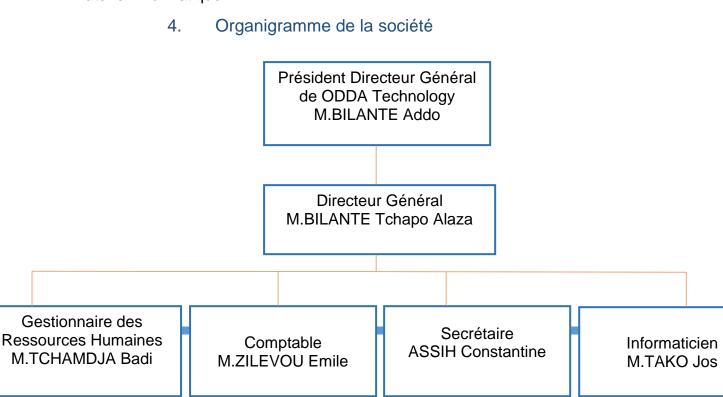
ODDA Technology est une Société Informatique spécialise dans le développement web (Site web et Application), mobile et en système réseau pour répondre au besoin fonctionnel des systèmes d'information.

2. Missions

La structure ODDA Technology s'est donné pour mission d'être la référence dans les domaines suivants : Conseils, Audit, Développement des systèmes et des solutions dans le domaine informatique et des réseaux.

3. Activités

Depuis sa création, elle s'est positionnée comme partenaire de choix dans les solutions informatiques. Afin d'apporter des solutions complètes et de haute qualité aux multiples besoins des entreprises soucieuses d'accroitre leur productivité et leur compétitivité, elle a bâti son activité autour de l'informatique, le réseau et la vente de matériel Informatique.



5. Quelques réalisations

- ✓ Une application dédiée à la gestion des écoles publiques et privées, et est utilisé dans plus de 100 écoles sur tout l'étendue du territoire : SCHOOL MANAGER WEB 2.0.
- ✓ Application pour la gestion des écoles supérieures : UNIVERSITY MANAGER
- ✓ Un système de calcul actuarielle des employés (Calcul des indemnités de licenciement et des indemnités de retraite) staffACTUARY

6. Service d'accueil

Notre service d'accueil pour ce stage au sein de la structure ODDA Technology fut la Direction Générale. A travers notre passage, nous avons pu reconstituer le parc informatique de la structure suivant le tableau ci-dessous :

Tableau 2: Parc informatique

Matériels	Caractéristiques techniques	Quantité
	HP Probook i3 RAM 8Go; ROM:	
Ordinateur de bureau	500Go	2
	HP OfficeJet Pro	
Imprimante multifonctions		1
	Box Nokia G-2400W-C	
Routeur		2

7. Plan de localisation



Figure 2: Localisation du centre d'accueil

II. THEME DU STAGE

Ce chapitre comporte essentiellement trois parties qui consiste à explorer les points généraux à aborder au cours du projet.

A. Présentation du sujet

Le ministère de la sécurité pour le recensement des accidents, les analyses statistiques et les prises de décision à la fin d'année utilisait jusque-là le logiciel Excel.

La société ODDA Technology, spécialisé dans la conception des logiciels informatiques et dans la vente de matériel, a reçu le projet. Dans le but de nous faire découvrir l'expérience du développement d'application en milieu professionnel, elle nous a confié le projet : « REALISATION D'UNE PLATEFORME DE COLLECTES ET D'ANALYSE DES INFORMATIONS SUR LES ACCIDENTS CONSTATES SUR UN TERRITOIRE : (Cas du Togo). ».

B. Problématique du sujet

Pour la collecte des informations des accidents constatés, les agents de police procédaient avec des bulletins d'analyse. Les résultats sont alors transférés dans un fichier Excel. Les feuilles n'étant pas très résistant aux intempéries, on fait face à un problème de perte de données. Les statistiques en fin d'années n'étant donc pas précises.

Après analyse de ces problèmes et dans le but de les résoudre, on se pose les questions suivantes :

- ✓ Comment recueillir de manière efficace les données ?
- ✓ Comment éviter la perte d'information ?
- ✓ Comment faire une gestion centralisée des données ?
- ✓ Comment optimiser le temps de traitement de l'information ?
- ✓ Comment avoir un aperçu visuel des accidents et des dégâts ?

C. Intérêt du sujet

Objectifs

Objectifs généraux

L'objectif global de ce projet est la recherche d'une solution optimale que celle utilisée présentement, en utilisant les Technologies de l'Information et de la Communication (TIC). Notre travail s'articule donc autour de l'automatisation en prenant en compte le fait que les données doivent être persistantes puisqu'on n'aura plus à utiliser les registres physiques pour l'enregistrement de données, mais plutôt des fichiers numériques (Base de données). Nous devons réaliser une base de données ainsi qu'une interface graphique associée, qui rend transparent pour l'utilisateur la gestion des données.

Objectifs spécifiques

Le système devra offrir les services suivants :

- ✓ Sécuriser les données manipulées et traitées
- √ Faciliter la cohérence des données
- ✓ Bénéficier d'un gain en temps au niveau des opérations effectuées
- √ Réduire les coûts et les risques d'erreurs
- ✓ Gestion des accidents
- ✓ Gestion des lieux

2. Résultats attendus

L'application fournira les fonctionnalités suivantes :

- ✓ Recensement des accidents sur le terrain
- ✓ Recensement des personnes impliquées et les dégâts
- ✓ Identification des lieux
- ✓ Consultation des accidents, des lieux et des dégâts
- ✓ Elaboration de statistiques et des états
- Elaboration des tâches planifiées

III. ETUDE DE L'EXISTANT

L'étude de l'existant décrit la manière dont les besoins fonctionnels sont actuellement pris en compte avant la mise en place d'une solution optimale.

Nous allons essentiellement revoir les méthodes de collecte d'information et les problèmes liés à ceux-ci.

Processus d'acquisition des données

Les agents de police se rendent sur le terrain avec des bulletins d'analyse, demande les informations sur l'accident et procèdent de façon manuscrite.

♣ Processus d'analyse des informations et de l'élaboration des statistiques

Les bulletins d'analyse préalablement remplis sont alors retranscrits dans un fichier Excel et les statistiques et états sont faits à l'aide du logiciel Microsoft Excel.

Pour avoir une idée générale sur comment se présente les bulletins d'analyse les illustrations ci-dessous vous sont présenté.

	BULLETIN D'ANALYSE D	ES ACCIDENTS	CONSTATES	- BA
Nom de l'entité rapportant :				
Numéro de la fiche :	BA:	_		
Année :				
	CARACTERIST	TIQUES DE L'ACCID	ENT	
Date de l'accident (jj mm aaaa Heure de l'accident (hh:mm) :		Région : Ville village :		Préfecture :
Tronçon :		Nombre de véhi Nombre de tués	ssés graves :	II II II
Zone 1 Urbaine 2 Rurale en agglomération 3 Rurale hors agglomération	Classification de la route 2 Route Nationale Inter-Etats (RNIE) 3 Route Nationale (RN) 4 Voirie urbaine 5 Piste répertoriée 6 Autres	Intersection 1 En intersection 2 Hors intersection 3 Passage à niveau	Type de route 1 Revêtue 2 Non revêtue 3 mixte	X: _ _ _ _ _ Y: _ _ _ _ _ _
Circonstances résumées de l'ac Ressortir les genres et immatriculation des corporelles et matérielles.	cident véhicules ; les mouvements des véhicules juste avant l'acc	ident ; les conséquences		Type du jour 1 Week-end 2 Veille de fête 3 Jour de fête

Figure 3: Fiche d'enregistrement de l'accident

PLATEFORME DIGITALE DE COLLECTE ET D'ANALYSE D'INFORMATION DES ACCIDENTS CONSTATES SUR UN TERRITOIRE (CAS DU TOGO)

BULLETIN D'ANALYSE DES ACCIDENTS CONSTATES - BA									
Nom de l'entité rapportant :	om de l'entité rapportant :								
Année :			Mois :						
		V	EHICULES IM	PLIQUES					
Caractéristi	iques du (des) véhicule(s)	Véhcule 1	Véhcule 2	Véhcule 3	Véhcule 4	Véhcule 5	Véhcule 6	Véhcule n-1	Véhcule n
Numéro d'immatriculation	Inscrire le numéro du véhicule sur le pointillé								
Pays d'immatriculation	Inscrire le pays d'immatriculation sur le pointillé								
Benre*	Selon table des genres du véhicule	1_1_1	1_1_1	1_1_1	1_1_1	1_1_1	1_1_1	1_1_1	1_1_1
Appartenance*	Selon table d'appartenance	1_1	1_1	1_1	1_1	1_1	1_1	1_1	1_1
remière mise en circulation	Inscrire les quatre chiffres de l'année	_ _ _	_ _ _	1_1_1_1_1	_ _ _	_ _ _	1_1_1_1_1	1_1_1_1_1	_ _ _
/alidité contrôle technique	1. Oui 2.Non	1_1	1_1	1_1	1_1	1_1	1_1	1_1	1_1
/alidité assurance	1. Oui 2.Non	1_1	1_1	1_1	1_1	1_1	1_1	1_1	1_1
iociété d'assurance	Inscrire le nom de la société sur le pointillé								
tat général (avant accident)	1. Bon 2. Mauvais	1_1	1_1	1_1	1_1	1_1	1_1	1_1	1_1
Pneus (avant accident)	1. Bon 2. Mauvais	1_1	1_1	1_1	1_1	1_1	1_1	1_1	1_1
eux (avant accident)	1. Bon 2. Mauvais	1_1	1_1	1_1	1_1	1_1	1_1	1_1	1_1

Figure 4: Fiche d'enregistrement des véhicules impliquées dans l'accident

IV. CRITIQUE DE L'EXISTANT

La gestion des accidents constitue une tâche importante qui implique la coopération de plusieurs services. Une mauvaise gestion de ces services ne va donc pas dans l'avantage des habitants et de leur sécurité. Le défaut d'automatisation dans le processus de gestion des accidents entraine donc plusieurs maux que sont :

- ✓ Le manque ou l'insuffisance de données
- √ Les statistiques annuelles erronées
- ✓ Les prises de décision douteuses
- ✓ La sécurité de la population menacées.

V. PROPOSITIONS ET CHOIX DE SOLUTIONS

Une étude approfondie a été mené sur le sujet, étude qui a permis de relever les manquements que présente l'existant, raison pour laquelle nous présentons deux solutions pour automatiser le système de gestion des accidents sur le territoire togolais.

A. Evaluation technique des solutions

1. Première solution

La première solution consiste à utiliser le logiciel Excel associé à une base de données MySQL et l'élaboration des statistiques se fera à l'aide du logiciel Tableau.

Excel est un logiciel de la suite bureautique Office de Microsoft et permet la création de tableaux, de calculs automatisés, de plannings, de graphiques et de bases de données. Excel est depuis quelques années déjà le tableur de référence.

MySQL est un système de gestion de base de données relationnelle (SQBDR) open source soutenue par Oracle et basé sur le langage de requête structuré (SQL). MySQL fonctionne sur pratiquement toutes les plates-formes, y compris <u>Linux</u>, <u>UNIX</u> et <u>Windows</u>. Bien qu'il puisse être utilisé dans un large éventail d'applications, MySQL est le plus souvent associé aux applications Web et à la publication en ligne.

En tant que premier choix du marché pour l'informatique décisionnelle moderne, <u>Tableau</u> aide les personnes et les organisations à être davantage axées sur les données, il facilite l'exploration et la gestion des données, et accélère la découverte et le partage d'informations.

Avantages

- ✓ Les données existantes ne nécessiteront pas de migration
- ✓ Excel est l'un des logiciels de calcul automatisés les plus puissant
- ✓ Etant très répandu, les logiciels cités plus haut offrent une panoplie de formation très complète.
- ✓ Grâce à la plateforme Tableau, la collecte des données devient très facile.
- ✓ Une image visuelle du tableau de bord fournie.

- ✓ Le serveur MySQL peu être intégré à d'autres langage de programmation et offre un outil d'administration très complet.
- ✓ Le couplage très productif du SQBD MySQL et du Logiciel Tableau qui
 offrent un module d'aide à la décision très complet.

Inconvénients

- ✓ La plupart des logiciels cités plus haut offre un support technique assez médiocre donc un paramétrage douteux et pas toujours conforme
- ✓ Le logiciel Tableau ne permet pas au client de bénéficier d'un tableau de bord prospectif.

Deuxième solution

Notre solution secours consistera en la réalisation d'une API qui sera consommé par une application web et mobile ; une approche qui sera beaucoup plus conforme car sur mesure.

Avantages

- ✓ Possibilité d'avoir une application sur mesure
- ✓ Maintenabilité plus facile et plus sûr
- ✓ L'application sera développée en fonction de la réalité problème
- ✓ La méthode d'acquisition des données autre fois manuscrite va être désormais digitale.
- ✓ La notion de contrôle pour une entreprise est très importante, surtout pour ce qui touche les technologies. En tant que propriétaire du logiciel sur mesure, nous seul, somme en contrôle.
- ✓ Un logiciel sur mesure nous permettra de gagner en agilité et en résilience face aux nombreuses fluctuations d'une entreprise en constant mouvement.
- ✓ Une application sur mesure peut, être hébergée là où bon nous semble, et nous pourrons choisir (ou changer) notre fournisseur selon nos critères de sécurité bien précis.

♣ Inconvénients

✓ Créer de toute pièce un produit, dématérialisé ou non, est un travail de longue haleine qui nécessite un temps de développement conséquent.

- ✓ Inconvénients du logiciel sur mesure car le coût de démarrage est plus important
- ✓ Difficile de tester avant d'investir.

B. Evaluation financière des solutions

1. Première solution

a) Coût d'achat logiciel

Tableau 3: Tableau de coût d'achat logiciel

Désignation	Quantité	Prix unitaire	Total
Serveur MySQL	1	0	\$0
Microsoft Office Professionnel 2021	20	\$440	\$8 800
Tableau Desktop	20	\$1000	\$20 000
Total			\$28 800
			(14 400 000 XOF)

b) Coût d'achat matériel

Tableau 4: Coût d'achat matériel

Matériel	Désignation	Quantité	Prix unitaire	Total
Serveur de base de données	Intel XEON E 3.4 GH 64Go DDR4	1	\$1 300	\$1 300
	Total			\$1 300
				(715 000 XOF)

c) Coût de formation

Tableau 5: Coût de formation

Désignation	Nombre de personnes	Prix unitaire	Total
Formation Microsoft Excel	20	\$85	\$1 700
Formation MySQL	20	\$85	\$1 700
Formation Tableau	20	\$85	\$1 700
Total	\$5 100		
	(2 805 000 XOF)		

d) Coût total

Tableau 6: Coût total

Désignation	Montant
Coût d'achat logiciel	\$28 800
Coût d'achat matériel	\$1 300
Coût de formation	\$5 100
Total	\$35 200 (19 360 000 XOF)

2. Deuxième solution

a) Coût de développement

Tableau 7: Coût de développement

Nombre de développeur	Nombre d'heure	Prix unitaire	Total
1	1080 (12h * 90)	\$10	\$10 800
Total			\$10 800 (5 940 000 XOF)

b) Coût d'hébergement

Tableau 8: Coût d'hébergement

Hébergeur	Désignation	Temps	Prix unitaire	Total
Hébergeur Cloud	Google Cloud	6 mois	\$300	\$1800
Total				\$1 800 (990 000 XOF)

c) Coût de formation

Tableau 9: Coût de formation

Nombre d'heures	Nombre de personnes	Prix unitaire	Total
(2h * 10 jours)	20	\$5	\$2 000
Total			\$2 000
			(1 100 000 XOF)

d) Coût total

Tableau 10: Coût total

Désignation	Montant
Coût de développement	\$10 800
Coût d'hébergement	\$1 800
Coût de formation	\$2 000
Total	\$14 600 (8 030 000 XOF)

C. Choix de solution

Suite à l'étude des deux solutions, celle qui a retenu notre attention est la deuxième. C'est une solution informatique qui consiste à concevoir une plateforme web de gestion des accidents. Plusieurs raisons motivent notre choix, entre autres le souci d'optimiser le temps de traitement de l'information, avoir un aperçu visuel des accidents et des dégâts, de la perte d'information.

VI. PLANNING PREVISIONNEL DE REALISATION

ID	Nom de tâche	Début	Terminer Durée	mai 2022 juin 2022 juil 2022 août 2022	
ם ו	Nom de tache			Durée	22/5 29/5 5.6 12/6 19/6 26/6 3/7 10/7 17/7 24/7 31/7 7.8
	Prise de contact avec le centre d'accueil	20/05/2022	20/05/2022	1j	ı
	Prise de connaissance du thèmeme	23/05/2022	23/05/2022		
	Compréhension du thème et identifications des besoins fonctionnels	24/05/2022	26/05/2022	3j	
4	Elaboration et validation du cahier des charges	26/05/2022	01/06/2022	5j	
5	Analyse, conception et modélisation du système	01/06/2022	14/06/2022	10j	
6	Prise en main des outils logiciels et technologies à utiliser	14/06/2022	16/06/2022	3j	
	Rédaction du rapport de stage	16/06/2022	28/07/2022	31j	
	Préprogrammation	17/06/2022	17/06/2022	1j	
	Ecriture du backend	20/06/2022	07/07/2022	14j	
	Ecriture du frontend Angular		18/07/2022	7j	
11	Ecriture du frontend Mobile		27/07/2022	7j	
12	Test fonctionnel et tests unitaires		05/08/2022	8j	
	Dépôt du rapport de stage	15/08/2022	15/08/2022	1 j	
		22/08/2022	22/08/2022	1j	

Figure 5: Planning prévisionnel de réalisation et diagramme de Gantt

PARTIE 2: ANALYSE ET CONCEPTION

I. CHOIX DE LA METHODE D'ANALYSE ET JUSTIFICATION

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles. Pour notre analyse nous utiliserons l'approche objet et le langage de modélisation UML avec le processus 2TUP.

A. Justification de de l'approche objet

L'approche objet est un ensemble de concepts stables, éprouvés et normalisés, une solution destinée à faciliter l'évolution d'applications complexes, une panoplie d'outils et de langages performants pour le développement.

L'approche objet permet :

- ❖ De réduire le besoin de maintenance et augmente à la fois la fiabilité et la flexibilité en créant de nouveaux comportements à l'aide d'objets existants.
- La réutilisabilité élevée du code.
- D'avoir une vision externe définissant les actions qu'il sera possible de faire subir au logiciel;
- D'avoir une vision interne dans laquelle seule la structure du logiciel sera considérée.

B. Langage de modélisation UML

1. Définition

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.

Adopté et standardisé par l'Object Management Group depuis 1997, il est aujourd'hui un outil de communication incontournable, utilisé sur une multitude de projets de par le monde.

En conséquence, la connaissance d'UML est désormais une des compétences qui sont exigées quasi systématiquement lors d'un recrutement. Il permet de modéliser les solutions informatiques par des diagrammes.

2. Pourquoi UML?

Le langage UML est né de la fusion de plusieurs méthodes préexistantes, il incorpore donc la plupart des fonctionnalités de ces méthodes en plus de ses propres fonctionnalités en réponse aux besoins des analystes. UML est devenu par la suite la référence en termes de modélisation objet grâce à la souplesse qu'il apporte aux projets de développement. MERISE (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise) est l'autre méthode d'analyse abordée au cours de notre formation à IAI-Togo. En comparaison à celle-ci, UML nous donne plus de possibilités, notamment au niveau de l'interface utilisateur. Grâce à sa notation graphique, UML nous permettra à l'issue de notre phase de modélisation :

- D'obtenir une modélisation de très haut niveau indépendante des langages et des environnements;
- De faire des simulations avant de construire notre système ;
- D'exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, etc....;
- De documenter le projet.

3. Les diagrammes UML

Il existe deux grands types de diagrammes ULM : les diagrammes de structure et les diagrammes de comportement (ces catégories sont elles-mêmes divisées en plusieurs sous-catégories). Ces variétés représentent les nombreux types de scénarios et de diagrammes que différents types de personnes utilisent.

Qu'il s'agisse des clients, des gestionnaires de projet, des auteurs techniques, des concepteurs, des analystes, des codeurs, des responsables assurance qualité ou encore des testeurs, tous utilisent un diagramme spécifique pour répondre à leurs besoins. Cela signifie que chaque mise en page requiert une attention et un souci du

détail différent. L'objectif de l'UML est de présenter visuellement des diagrammes faciles à comprendre.

Tableau 11: Listes des diagrammes UML

Catégories de diagrammes	Diagrammes
Diagrammes de structure	 ✓ Diagramme de classes ✓ Diagramme d'objets ✓ Diagramme de composants ✓ Diagramme de déploiement ✓ Diagramme de paquetages ✓ Diagramme de profil
Diagrammes de comportement	 ✓ Diagramme d'activités ✓ Diagramme de cas d'utilisation ✓ Diagramme global d'interaction ✓ Diagramme de temps ✓ Diagramme d'états-transition ✓ Diagramme de séquences ✓ Diagramme de communication

UML n'étant pas une méthode, l'utilisation de l'un ou l'autre des diagrammes cidessus est laissée à l'appréciation de l'analyste.

Dans notre cas, nous utiliserons successivement les diagrammes suivants :

- ✓ Diagramme de cas d'utilisation ;
- ✓ Diagramme d'activité ;
- ✓ Diagramme de séquences ;
- ✓ Diagramme de déploiement ;
- ✓ Diagramme de classes.

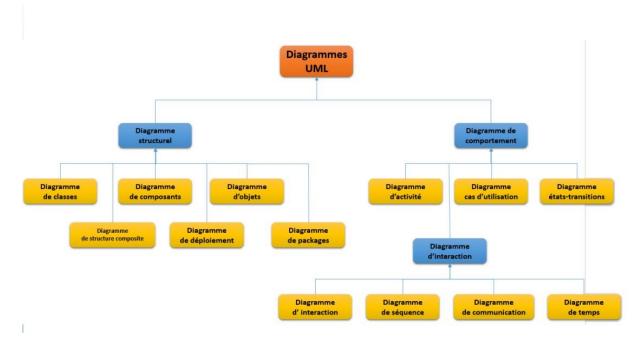


Figure 6: Hiérarchie des diagrammes UML

C. Le processus 2TUP

1. Présentation

Le Processus Unifié ou UP (Unified Process) est une méthode <u>générique</u> de développement de logiciel développée par les concepteurs d'UML, il est <u>itératif et incrémental</u>, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques. Il existe donc un certain nombre de méthodes issues de UP comme par exemple RUP (Rational Unified Process) et 2TUP (Two Track Unified Process). L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques.

La complexité croissante des systèmes informatique a conduit les concepteurs à s'intéresser aux méthodes. Définir un seul processus universel serait une grave erreur car la variété des systèmes et des techniques ne le permet pas.

Devant le nombre de méthodes disponibles, le choix parmi elles devient difficile, beaucoup de questions peuvent se poser à un chef de projet lors d'un démarrage de projet ; dans notre analyse nous utiliserons le processus 2TUP.

2TUP signifie « 2 Track Unified Process ». C'est un processus qui répond aux caractéristiques du Processus Unifié. Le processus 2TUP apporte une réponse aux contraintes de changement continuel imposées aux systèmes d'information de l'entreprise.

En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des « chemins fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.



Figure 7: Le système d'information soumis à deux types de contraintes

La branche gauche (fonctionnelle) capitalise la connaissance métier du projet. Elle constitue généralement un investissement pour le moyen et le long terme.

Les fonctions du système d'information sont en effet indépendantes des technologies utilisées. Cette branche comporte les étapes suivantes :

- ✓ La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs.
- ✓ L'analyse.

La branche droite (architecture technique) capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent être en effet indépendamment des fonctions à réaliser.

Cette branche comporte les étapes suivantes :

- ✓ La capture des besoins techniques.
- ✓ La conception générique.

La branche du milieu : à l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l'obtention d'un processus en forme de Y.

Cette branche comporte les étapes suivantes :

- ✓ La conception préliminaire.
- ✓ La conception détaillée.
- ✓ Le codage.
- ✓ L'intégration.

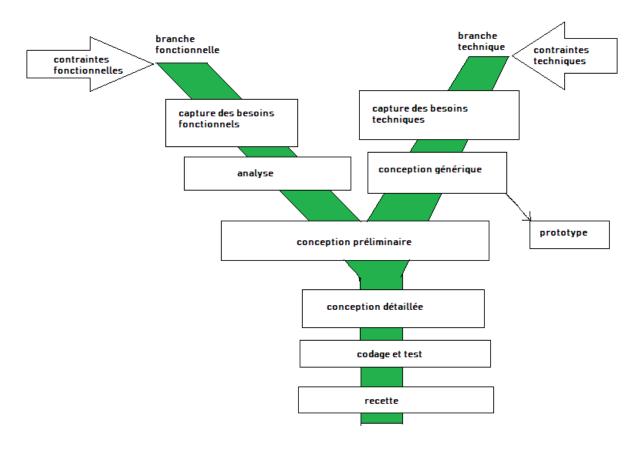


Figure 8: Le processus de développement en Y

Justification du choix du processus 2TUP

Le processus 2TUP préconise un cycle de vie en Y qui dissocie la résolution des questions fonctionnelles et techniques. Il permet également une conception générique (utilisation des Framework et pattern de conception), il est orienté utilisateur pour répondre aux besoins de celui-ci, les modèles définit tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide, en définissant des priorités pour chaque fonctionnalité, on peut minimiser les risques d'échec du projet

PLATEFORME DIGITALE DE COLLECTE ET D'ANALYSE D'INFORMATION DES ACCIDENTS CONSTATES SUR UN TERRITOIRE (CAS DU TOGO)

NB : Notons que cette démarche ne s'oppose en aucun cas à l'utilisation des processus de développement agiles. Au contraire, elle encadre leur mise en œuvre. Nous choisissons donc comme processus de développement, le processus UP (Unified Process).

II. CHOIX DE L'OUTIL DE MODELISATION ET JUSTIFICATION



Figure 9: Logo de Microsoft Visio

L'outil retenu pour la modélisation de notre analyse et conception système est Microsoft Visio 2019.

Microsoft Visio est un logiciel de diagrammes et de synoptiques pour Windows qui fait partie de la suite bureautique Microsoft Office mais se vend séparément. Il permet de créer des diagrammes et organigrammes, des plans d'architectes ou techniques, des réseaux de PERT ou encore des diagrammes IDEF0.

Le logiciel dispose d'outils de dessin généraux pour répondre à tous les besoins spéciaux. Dans le même temps, les outils de dessin généralisés nous aident à personnaliser les diagrammes en fonction de nos besoins uniques et spécifiques.

L'interface est assez intuitive et nous permet de créer des documents professionnels sans l'aide du dessinateur ou du personnel de CAO (Conception Assisté par Ordinateur). Le progiciel lui-même coûte beaucoup moins cher que ce qu'un outil de CAO pour chaque catégorie spécifique coûterait.

III. ETUDE DETAILLEE DE LA SOLUTION

Dans ce chapitre, nous modéliserons notre futur système à partir des différents diagrammes UML.

A. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système.

Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent.

1. Les acteurs

Un acteur représente « un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié ».

Un acteur peut être une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie. Le diagramme des cas d'utilisation d'UML distingue deux acteurs à savoir :

- ✓ Les acteurs principaux (qui modifient l'état du système ou qui consultent cet état);
- ✓ Les acteurs secondaires (acteurs auxquels le système fait appel pour répondre aux sollicitations d'un acteur principal.

Les acteurs de notre projet sont :

- ✓ Administrateur
- ✓ Agent de police
- ✓ Observateur

Cas d'utilisation identifiés

Un cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système. Le tableau ci-dessous, donne un récapitulatif des cas d'utilisation de notre système.

Tableau 12: Liste des cas d'utilisation

Acteurs	Cas d'utilisations	
Le cas d'utilisation S'authentifier est commun à tous les acteurs du système		
Administrateur	 ✓ Gérer les utilisateurs du système ✓ Gérer les statistiques ✓ Consulter les détails ✓ Générer des états 	
Agent de police	 ✓ Enregistrer les véhicules impliqués dans un accident; ✓ Enregistrer les détails d'un accident; ✓ Enregistrer les données de l'état des lieux d'un accident; ✓ Enregistrer des données du conducteur d'un véhicule; ✓ Enregistrer les passagers des véhicules ✓ Enregistrer les piétons 	
Observateur	✓ Voir statistique✓ Consulter détails✓ Générer des états	

a) Cas d'utilisation par acteurs

♣ Diagramme de cas d'utilisation de l'administrateur

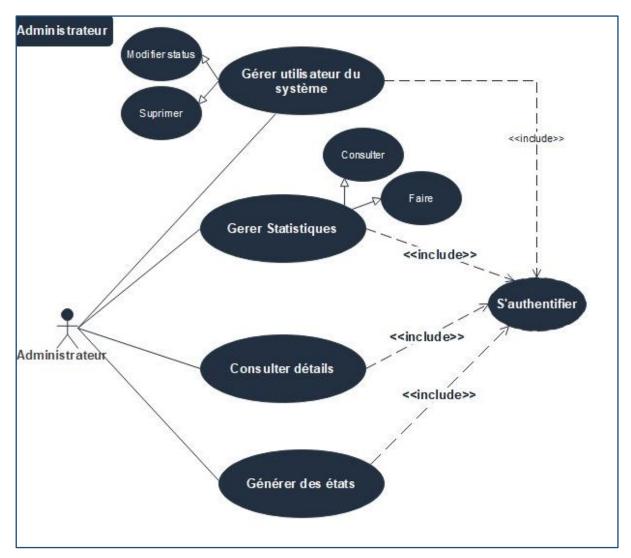


Figure 10: Diagramme de cas d'utilisation de l'administrateur

♣ Diagramme de cas d'utilisation de l'agent de police

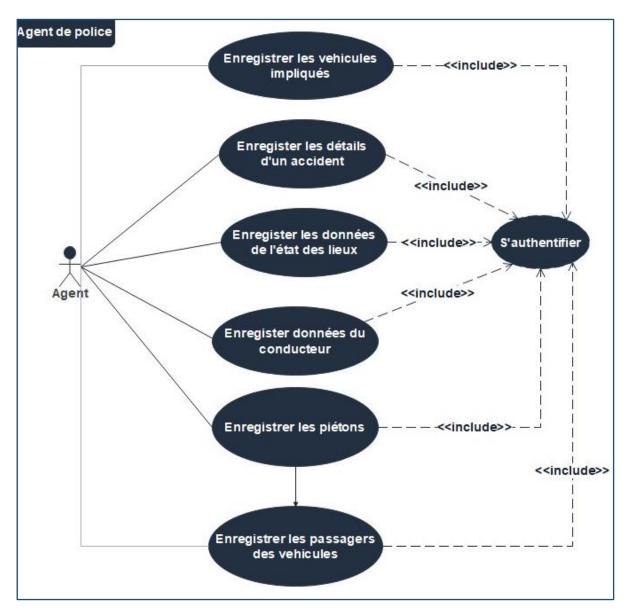


Figure 11: Diagramme de cas d'utilisation de l'agent de police

♣ Diagramme de cas d'utilisation de l'observateur

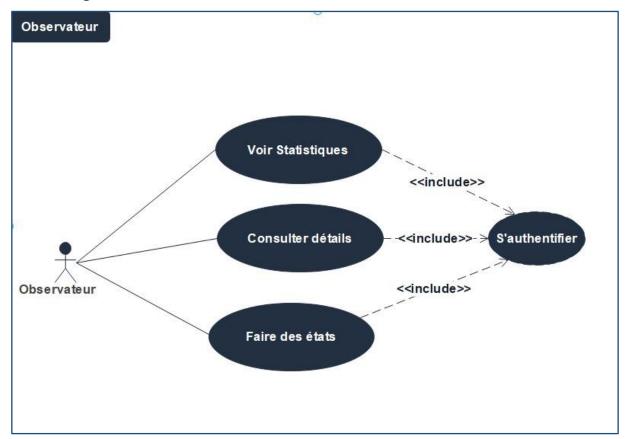


Figure 12: Diagramme de cas d'utilisation de l'observateur

b) Description textuelle de quelques cas d'utilisation

La description textuelle d'un cas d'utilisation permet de clarifier le déroulement de la fonctionnalité et de décrire la chronologie des actions qui devront être reliées. Nous allons utiliser la structure suivante :

- ✓ Un sommaire d'identification qui inclut le titre, le résumé, les dates (création et mise à jour), la version, les acteurs responsables.
- ✓ La description des enchainements qui décrit les préconditions, le scenario nominal, les enchainements alternatifs, les enchainements d'exception et les post conditions.

Ainsi, voici les cas d'utilisation que nous allons décrire textuellement dans le cadre de ce projet :

- ✓ S'authentifier
- ✓ Enregistrer un accident
- ✓ Générer un état

Cas d'utilisation s'authentifier

SOMMAIRE D'IDENTIFICATION

Nom: S'authentifier

Objectif : Permettre à l'utilisateur de se connecter au système

Acteur(s) : **Administrateur**

Date: 14/07/2022 Responsable: VOUTI-KOTOR Nyuiwola Version: 1.0

PRE-CONDITION

- ✓ Le système doit être fonctionnel
- ✓ L'utilisateur doit être connecté à internet

DESCRIPTION DES ENCHAINEMENTS

- 1- Scénario nominal
- 2- L'utilisateur se rend sur l'adresse web de l'application
- 3- L'utilisateur saisit son matricule ensuite son mot de passe
- 4- L'utilisateur clique sur le bouton se connecter
- 5- L'utilisateur est redirigé vers la page d'accueil
- 6- Scénario alternatif
- A1 : Le matricule de l'utilisateur ne se trouve pas dans le système
 - 7- Un message d'avertissement es envoyé à l'utilisateur
 - 8- Le scénario reprend au point 2 du scénario nominal.
- A2: Le mot de passe saisi n'est pas correct
 - 9- Un message d'avertissement es envoyé à l'utilisateur
 - 10- Le scénario reprend au point 2 du scénario nominal.
 - 11-Scénario d'exception
- E1: L'utilisateur annule la connexion
 - 1- Le scénario nominal est interrompu

POST-CONDITION

√ L'utilisateur connecté est ajouté au contexte de connexion du Backend

Cas d'utilisation enregistrer un accident

SOMMAIRE D'IDENTIFICATION

Nom: Enregistrer un accident

Objectif : Permettre à l'agent de police d'enregistrer un accident

Acteur(s): Agent de police

Date: 14/07/2022 Responsable: VOUTI-KOTOR Nyuiwola Version: 1.0

PRE-CONDITION

- ✓ Le système doit être fonctionnel
- ✓ L'utilisateur doit être connecté à internet
- ✓ L'utilisateur doit être authentifié

DESCRIPTION DES ENCHAINEMENTS

Scénario nominal

- 1- L'utilisateur se rend sur l'écran d'enregistrement de l'accident sur l'application mobile
- 2- L'utilisateur choisit la région, la préfecture et la ville
- 3- L'utilisateur choisit le type de route et la classification de la route et prend les coordonnées géographiques de l'états des lieux.
- 4- L'utilisateur saisit les informations relatives à l'accident
- 5- L'utilisateur prend des images de l'état des lieux et des dégâts.
- 6- L'accident est enregistré.

Scénario alternatif

A1: L'utilisateur omet un champ

- 1- Un message d'avertissement es envoyé à l'utilisateur
- 2- Le scénario reprend au point 2 du scénario nominal.

POST-CONDITION

✓ L'accident est enregistré et l'utilisateur est redirigé vers l'écran d'enregistrement des véhicules impliqués.

B. Diagramme de séquence

Un diagramme de séquence système est une représentation graphique des interactions entre les acteurs d'un cas d'utilisation et le système selon un ordre chronologique.

Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système ; le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

1. Quelques concepts

- ✓ Une Ligne de vie est une ligne verticale en-dessous d'un objet pour matérialiser la vie de cet objet.
- ✓ Une période d'activité est une bande rectangulaire juxtaposée sur la ligne de vie pour illustrer la période pendant laquelle l'objet peut recevoir ou émettre un appel.
- ✓ Un message est l'exécution d'une méthode par un objet émetteur sur un autre objet récepteur. La méthode exécutée doit être dans la classe de l'objet récepteur.

On distingue divers types de messages :

- ✓ Les messages simples : ce sont des messages qui n'ont aucune spécificité particulière d'envoi et de réception ;
- ✓ Les messages avec durée de vie : pour ces types de message, l'expéditeur attend une réponse du récepteur pendant un certain temps et reprend ses activités si aucune réponse n'a lieu dans le délai prévu ;
- ✓ Les messages synchrones : pour ces types de message, l'expéditeur est bloqué jusqu'au signal de prise en compte par le destinataire ;
- ✓ Les gardes : ce sont des messages qui ne sont émis qu'après le remplissage d'un certain nombre de conditions

2. Quelques diagrammes de séquences système de notre projet

a) Diagramme de séquence du cas « S'authentifier »

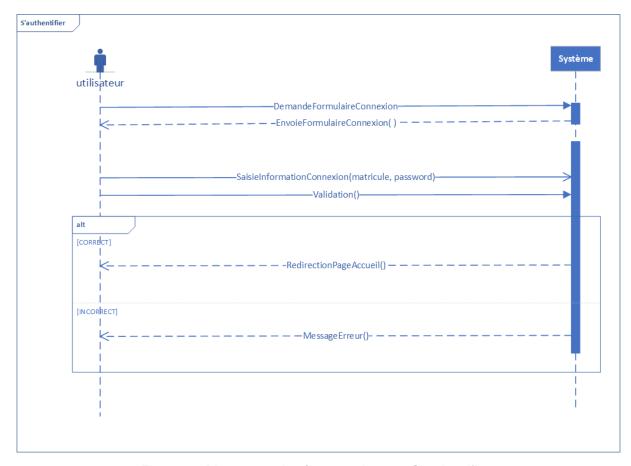


Figure 13: Diagramme de séquence du cas « S'authentifier »

b) Diagramme de séquence du cas « Enregistrer accident »

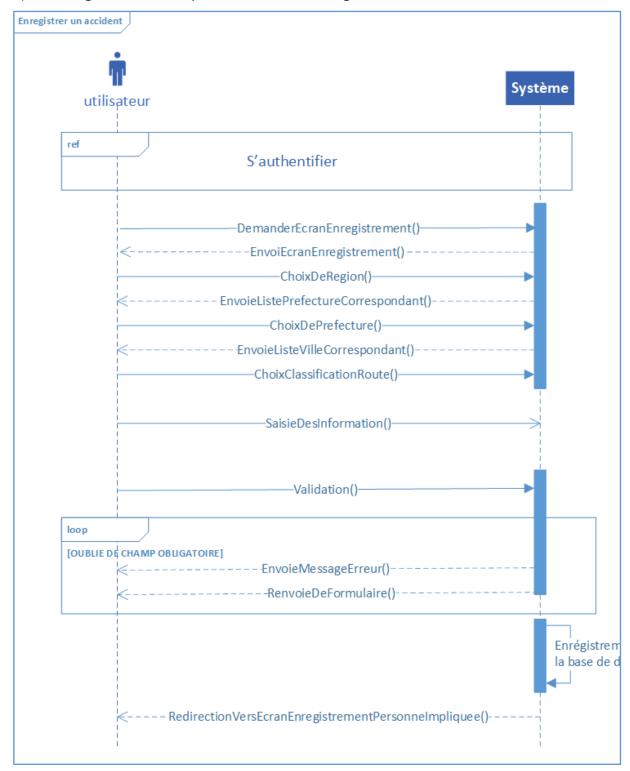


Figure 14: Diagramme de séquence « Enregistrer un accident »

c) Diagramme de séquence du cas « Enregistrer personne »

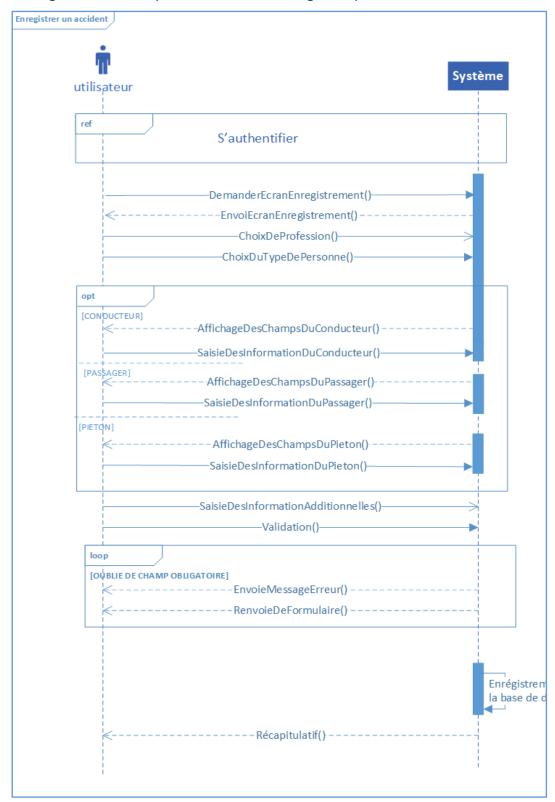


Figure 15: Diagramme de séquence « Enregistrer une personne »

3. Quelque diagramme de séquence détaillé de notre système

Nous rappelons que ce dernier est utilisé pour voir les interactions entre les classes de votre système.

Il y a donc très souvent des classes qui ont un rôle bien précis qu'il est intéressant de bien visualiser dans un diagramme de séquence.

C'est notamment le cas :

- ✓ Pour les classes qui représentent des composants de l'IHM.
- ✓ Pour la classe qui contrôle globalement le système avec la prise en compte de la gestion événementielle.
- ✓ Pour les classes qui implémentent la persistance des attributs (associées à une base de données).

Jackobson distinguent les trois stéréotypes suivants :

- « boundary » : classes qui servent à modéliser les interactions entre le système et ses acteurs.
- « control » : classes utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.
- « entity » : classes qui servent à modéliser des informations durables et souvent persistantes.

Voici quelques diagrammes de séquence détaillés de notre système.

a) Diagramme de séquence détaillé du cas « S'authentifier »

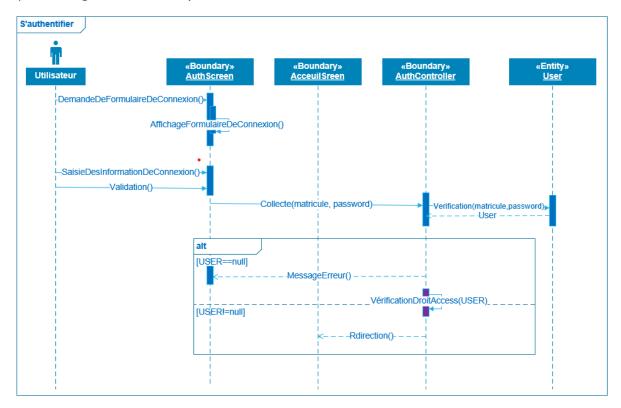


Figure 16: Diagramme de séquence détaillé du cas « S'authentifier »

b) Diagramme de séquence détaillé du cas « Enregistrer un accident »

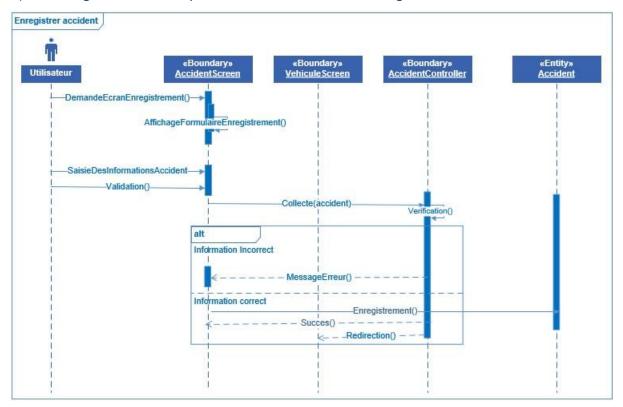


Figure 17: Diagramme de séquence détaillé du cas « Enregistrer un accident »

C. Diagramme d'activité

Dans le langage UML, les diagrammes d'activités fournissent une vue d'ensemble du comportement du système en décrivant une série d'actions dans un processus. Les diagrammes d'activité sont similaires aux organigrammes de traitement de l'information en ce sens qu'ils montrent le flux entre les actions au sein d'une activité. Cependant, les diagrammes d'activité peuvent également représenter des flux parallèles et alternés simultanés.

Les diagrammes d'activités permettent de modéliser un processus interactif, global ou partiel d'un système spécifique (logiciel, système d'information).

Décrire un cas d'utilisation par un graphe d'activité correspond à sa traduction algorithmique. L'activité est une implémentation partielle d'un cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Les diagrammes d'activité présentent une vue globale et temporelle du système modélisé.

- Quelque Diagramme d'activité de notre système
- a) Diagramme d'activité du cas « S'authentifier »

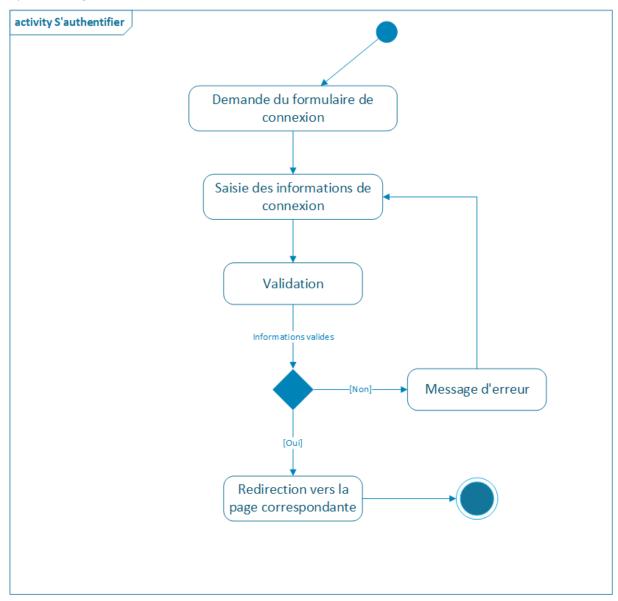


Figure 18: Diagramme d'activité du cas « S'authentifier »

b) Diagramme d'activité du cas « Enregistrer accident »

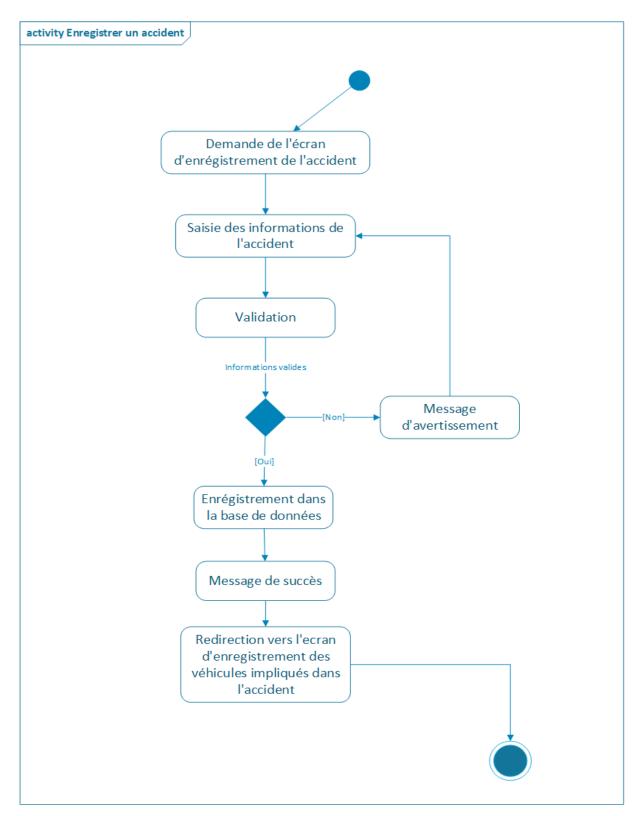


Figure 19: Diagramme d'activité du cas « Enregistrer un accident »

c) Diagramme d'activité du cas « Enregistrer une personne »

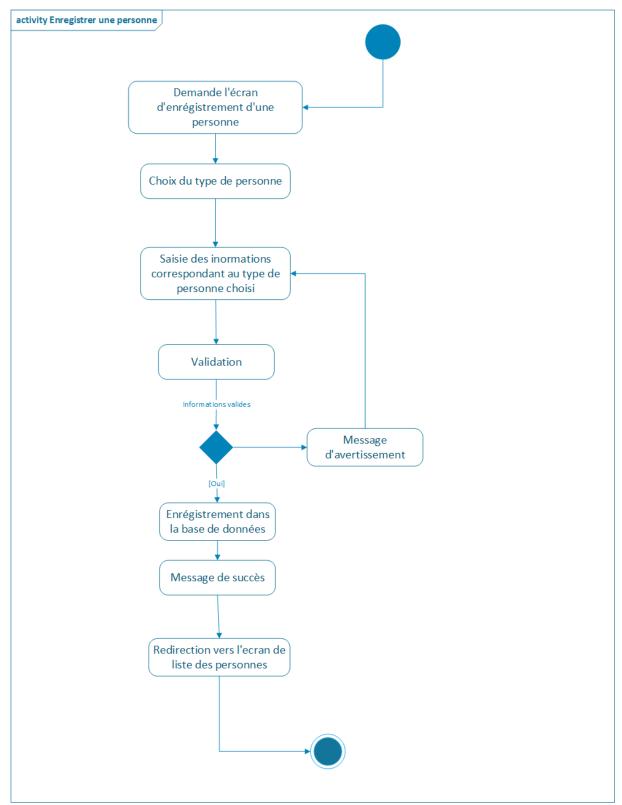


Figure 20: Diagramme d'activité du cas « Enregistrer une personne »

D. Diagramme de classe

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. Il s'agit d'un schéma présentant les classes et les interfaces du système ainsi que les différentes relations entre elles. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, 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.

Le diagramme de classe de notre application se présente comme suit :

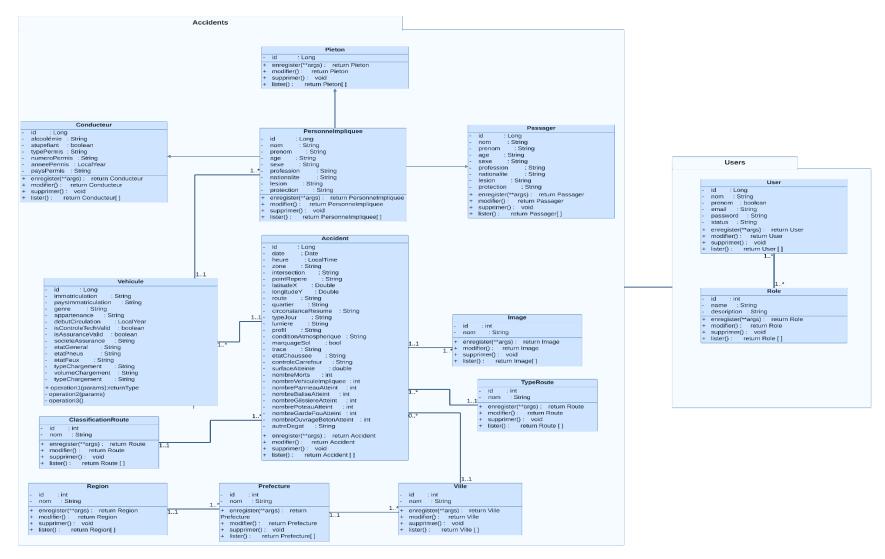


Figure 21: Diagramme de classe du système

E. Diagramme de paquetage

De la manière qu'un système peut être découpé en sous-systèmes, UML propose de décomposer un système d'informations en paquetages. Un paquetage regroupe des éléments de la modélisation (les classes par exemple), portant sur un sous-ensemble du système d'information. Le diagramme de paquetage de notre application se présente comme suit :

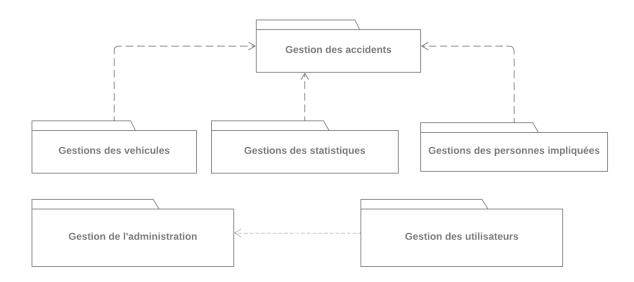


Figure 22: Diagramme de paquetage du système

F. Diagramme de déploiement

Les diagrammes de déploiement montrent la disposition physique des matériels qui composent le système et la répartition des composants sur ces matériels. Ils correspondent à la vue de déploiement d'une architecture logicielle. A travers ce diagramme, le concepteur identifie les matériels nécessaires pour le déploiement de l'application et surtout les parties du système devant se retrouver sur les matériels. Le diagramme de déploiement de notre application se présente comme suit :

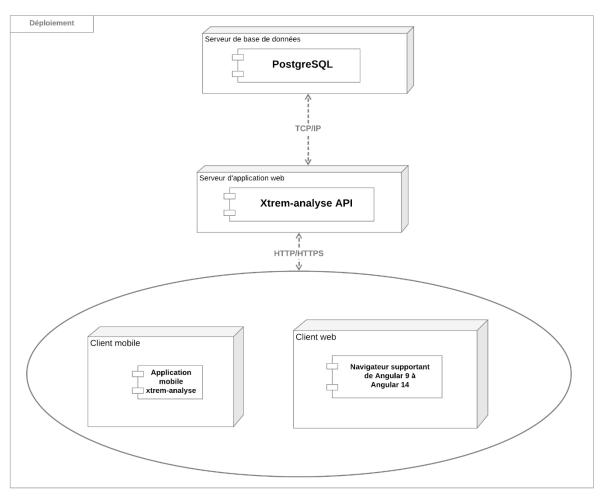


Figure 23: Diagramme de déploiement du système

PARTIE 3 : REALISATION ET MISE EN OEUVRE

I. MATERIELS ET LOGIIELS UTILISES

A. Choix du matériel

Tableau 13: Choix du matériel

Désignation	Propriétés	Caractéristiques
Ordinateur portable	Marques et modèle	ASPIRE A515-52
	Processeur et fréquence	Intel(R) Core (TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
	Mémoire RAM	8,00 Go (7,85 Go utilisable)
	Type de système	Système d'exploitation 64bits, processeur x64
	Disque dur	1TB
	Système d'exploitation	Windows 10 Professionnel

B. Choix du logiciel utilisé

1. Langage de programmation et technologies utilisées

a) Langage de programmation

Le monde de l'informatique est en constante évolution et l'émergence continue de langages de programmation en est un bon exemple. Il existe donc plusieurs langages de programmation notamment dans le développement d'application web.

On peut citer entre autres :

- ✓ Java : Java est un langage de programmation populaire et est utilisé à grande échelle dans le monde entier pour le développement d'applications. Il présente des avantages tels que le multithreading, l'extensibilité, la gestion de la mémoire, la haute sécurité, le support communautaire, etc. qui le rendent extrêmement pertinent pour différents projets de développement d'applications.
- ✓ Javascript : JavaScript est un langage de programmation orienté objet. Il a été conçu pour rendre le développement web plus facile et plus attrayant. Dans la plupart des cas, JavaScript est utilisé pour créer des éléments réactifs et interactifs pour les pages web, améliorant ainsi l'expérience de l'utilisateur.
- ✓ Python (avec son framework Django): Python pour le développement web, la science des données, les scripts et l'automatisation. Comme JavaScript, Python bénéficie également d'un énorme soutien communautaire et dispose de nombreux frameworks, bibliothèques et outils utiles qui peuvent nous aider à créer une application web en un temps rapide.
- ✓ PHP: Acronyme désignant initialement Personal Home Page, est actuellement connu sous le nom de Hypertext Preprocessor. Il s'agit d'un langage de script côté serveur, utilisé pour le développement de sites Web statiques et dynamiques. PHP peut également développer des applications web.

Pour le développement de notre système nous avons opté pour l'utilisation du langage de programmation Java.

Justification

Java détient de manière fiable la meilleure situation absolue du record TIOBE parmi tous les dialectes de programmation restants. Quelques avantages de Java :

- ✓ Il nous permet de créer des projets standard et du code réutilisable.
- ✓ Il s'exécute sur n'importe quelle machine qui n'a pas besoin de s'embêter avec une programmation unique à introduire, cependant, la JVM doit être disponible sur la machine.
- ✓ Il n'a pas de pointeur sans équivoque. En plus de cela, c'est un administrateur de sécurité qui caractérise l'entrée des classes.

- ✓ Distribution de la mémoire : En Java, la mémoire est divisée en deux sections, l'une est stockée et l'autre est empilée. À n'importe quel moment que nous prononçons, une variable JVM donne de la mémoire à partir de l'une ou l'autre pile ou espace de pile. Il aide à conserver l'information et à la rétablir sans aucun problème.
- ✓ Java fourni un nombre important de frameworks et d'outils très puissants répondants généralement à tous les besoins.

Malgré tous les avantages cités plus haut, Java comme tous les autres langages de programmation présente des inconvénients.

On peut noter entre autres :

- ✓ La lenteur par rapport à différents langages.
- ✓ Java nécessite une mesure critique ou significative d'espace mémoire par rapport à différents dialectes.
- ✓ Il n'y a pas de support pour la programmation de bas niveau en Java, car les pointeurs manquent.
- ✓ Courbe d'apprentissage plus rudes et plus longues.

b) Les technologies utilisées

Pour ce qui est des technologies utilisées, nous en avons plusieurs qui sont tout autant important l'une que l'autre. Ainsi à la vue du langage choisit en l'occurrence le **Java**, nous nous devons d'utiliser une technologie qui nous permettent d'utiliser ce langage pour concevoir une application web, raison pour laquelle nous avons opté pour les technologies suivantes :

♣ Spring Framework



Figure 24: Logo de Spring Framework

Le framework Spring (Spring Framework) est un framework d'application opensource qui fournit un support d'infrastructure pour le développement d'applications Java. L'un des frameworks Java Enterprise Edition (Java EE) les plus populaires,

Spring aide les développeurs à créer des applications très performantes en utilisant de simples objets Java.

La logique de base de Spring est l'injection de dépendances. L'injection de dépendances est un modèle de programmation qui permet aux développeurs de construire des architectures plus découplées. L'injection de dépendances signifie que Spring comprend les différentes annotations Java qu'un développeur ajoute aux classes. Spring sait que le développeur veut créer une instance d'une classe et que Spring doit la gérer. Spring comprend également les dépendances et s'assure que toutes les instances créées ont des dépendances correctement remplies.

♣ Le framework Spring boot



Figure 25: Logo de Spring boot

Spring Boot est un cadre open source basé sur Java utilisé pour créer un microservice. Il est développé par l'équipe Pivotal et est utilisé pour construire des applications Spring autonomes et prêtes pour la production.

Nous avons Spring Boot en raison des caractéristiques et des avantages qu'il offre, comme indiqué ici :

- ✓ Il offre un moyen flexible de configurer les Java Beans, les configurations XML et les transactions de base de données.
- ✓ Il offre un puissant traitement par lots et gère les points d'extrémité REST.
- ✓ Dans Spring Boot, tout est configuré automatiquement ; aucune configuration manuelle n'est nécessaire.
- ✓ Il offre une application Spring basée sur des annotations
- ✓ Il facilite la gestion des dépendances
- ✓ Il inclut un conteneur de servlets intégré.

JasperReports



JasperReports est un outil de reporting open source, offert sous forme d'une bibliothèque qui peut être embarquée dans tous types d'applications Java. Jasper Reports est un outil conçu pour les applications Java et les bases de données Oracle. Nous l'avons utiliser pour générer les états de notre projet.

♣ PostgreSQL



Figure 26: Logo de PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD Ce SGBDRO utilise des types de données modernes, dits composés ou enrichis suivant les terminologies utilisées dans le vocable informatique usuel. Ceci signifie que PostgreSQL peut stocker plus de types de données que les types simples traditionnels entiers, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type, etc.

PostgreSQL est plus avancé que ses concurrents dans la conformité aux standards SQL (énoncé vrai entre 2012 et 2015). PostgreSQL est pratiquement conforme (de plus en plus conforme) aux normes ANSI SQL 89, SQL 92 (SQL 2), SQL 99 (SQL 3), SQL :2003 et SQL :200814. Il fonctionne sur diverses plates-formes matérielles et sous différents systèmes d'exploitation. PostgreSQL est largement reconnu pour son comportement stable, proche d'Oracle, mais aussi pour ses possibilités de programmation étendues, directement dans le moteur de la base de données, via PL/pgSQL. Le traitement interne des données peut aussi être couplé à d'autres modules externes compilés dans d'autres langages.

Twilio Api



Figure 27: Logo de Twilio

Twilio est une entreprise américaine spécialisée dans les communications unifiées via une plateforme cloud. Twilio permet aux développeurs de logiciels de créer et de recevoir des appels téléphoniques, d'envoyer et de recevoir des messages texte et d'effectuer d'autres fonctions de communication à l'aide de ses API de service Web.

Twilio est l'un des meilleurs sur le marché grâce à sa documentation très bien fourni, à un prix abordable et très simple d'utilisation.

Nous avons utilisé twilio dans notre application notamment pour la récupération de mot de passe et pour l'envoi de notification en cas d'activation du compte d'un utilisateur.

♣ Le langage de programmation Dart



Figure 28: Logo de Dart

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web.

Dart est un langage orienté objet, basé sur les langages JavaScript, Java et Objective C. IL utilise la classe, récupérateur de mémoire avec une syntaxe de type C1. Dart peut se compiler en code natif ou en JavaScript. Il prend en charge les interfaces, les mixins, les classes abstraites, les génériques réifiés et l'inférence de type.

Notre développement est effectué avec la dernière version de ce langage à savoir le Dart 3.0.0.

Le framework Flutter



Figure 29: Logo de Flutter

Flutter est un kit de développement de logiciel (SDK) d'interface utilisateur opensource créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code.

La première version de Flutter était connue sous le nom de code "Sky" et fonctionnait sur le système d'exploitation Android. Elle a été dévoilée lors du sommet des développeurs Dart de 2015, avec l'intention déclarée de pouvoir effectuer un rendu cohérent à 120 images par seconde.

Nous avons utilisé ce framework pour notre développement parce qu'il permet d'écrire du code en peu de lignes et surtout nous permet de pouvoir gérer les applications multiplateformes.

Le framework Angular



Figure 30: Logo d'Angular

Angular est une plateforme et un cadre permettant de créer des applications client à page unique en utilisant HTML et TypeScript. Angular est écrit en TypeScript. Il implémente les fonctionnalités principales et optionnelles sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans vos applications.

L'architecture d'une application Angular repose sur certains concepts fondamentaux. Les éléments de base du cadre Angular sont des composants Angular organisés en NgModules. Les NgModules rassemblent le code connexe en ensembles fonctionnels ; une application Angular est définie par un ensemble de NgModules. Une application comporte toujours au moins un module racine qui permet de démarrer et comporte généralement de nombreux autres modules fonctionnels.

Quelques raisons qui nous ont poussé à choisir Angular :

- ✓ Rapidité d'exécution conséquente
- ✓ Les allers-retours entre le serveur et le navigateur sont considérablement réduits. Tous les calculs se font sur la partie client
- ✓ Des compétences provenant de Google : Les développeurs qui maintiennent le code source d'Angular sont formés par Google, ce qui assure un certain niveau de qualité, de performance et de sécurité.
- ✓ La facilité de prise en main des projets : La structure du code est facile à lire
 et permet de bien se retrouver dans le projet.

↓ Le framework CSS Bootstrap.



Figure 31: Logo de Bootstrap

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. Il se base sur un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style aux différents composants d'une page web. Nous avons fait le choix de Bootstrap 4 car en plus de sa simplicité d'utilisation, c'est l'un des projets Open Source les plus populaires sur la plateforme de gestion de développement GitHub.

GitHub.



Figure 32: Logo de GitHub

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions **Git**. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque **projet**.

GitHub est la solution Git pour les équipes professionnelles ; c'est une plateforme web qui permet d'effectuer le clouding. Un petit rappel, au sein de notre entreprise nous avons décidé d'opter pour la méthode AGILE, ce qui implique un travail de groupe. GitHub nous permet alors de pouvoir travailler et de de suivre l'évolution de nos codes quelle que soit la distance. Il nous permet d'effectuer le Versionnage et de faciliter le suivi et le développement du projet.

c) Outil de développements utilisé : IntelliJ IDEA

Les environnements de développement intégré (EDI) offrent des outils permettant l'édition du code, le débogage, la génération, la mise en œuvre d'applications. Parmi les EDI les plus populaires, on retrouve en bonne position IntelliJ IDEA, EDI propriétaire, développé par Jetbrains..



Figure 33: Logo de IntelliJ IDEA

IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement

de développement intégré destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala. IntelliJ est le meilleur environnement de développement d'application JAVA.

II. ARCHITECTURE MATERIEL ET LOGICIEL DE L'APPLICATION

A. Architecture globale

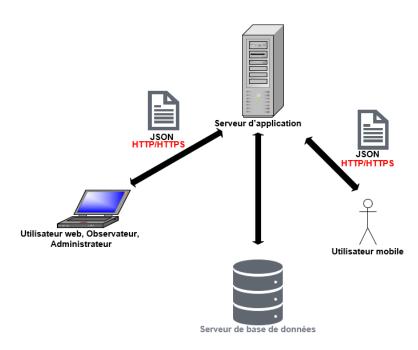


Figure 34: Architecture globale de l'application

B. Architecture matérielle

L'architecture logique du système est divisée en trois niveaux ou couches : couche de présentation ; couche de traitement ; couche d'accès aux données.

Dans l'architecture à 3 niveaux (appelée architecture 3-tier), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre:

- 1. Le client : le demandeur de ressources
- 2. **Le serveur d'application** : (appelé aussi *middleware*) le serveur chargé de fournir la ressource mais faisant appel à un autre serveur
- 3. Le serveur secondaire : (Généralement un serveur de base de données), fournissant un service au premier serveur.

Dans notre cas, l'application web sera déployé sur un serveur d'application (Heroku), la base de données sera mise sur un serveur de type DaaS (Database as Service).

C. Architecture logicielle

L'architecture que nous avons utilisée pour la mise en œuvre de notre application est l'architecture MVC : Model-View-Controller. Le MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs :

- ✓ Un modèle (Model) contient les données à afficher.
- ✓ Une vue (View) contient la présentation de l'interface graphique.
- ✓ Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

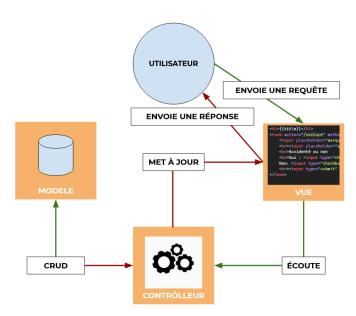


Figure 35: Architecture MVC

III. SECURITE DE L'APPLICATION

Un certain nombre de règles de sécurités et de bonnes pratiques ont été mises en place au cours de notre développement pour assurer un niveau de sécurité acceptable de nos applications :

- ✓ Aucune fonctionnalité n'est réalisable sans authentification : l'utilisateur sera systématiquement redirigé vers la page d'authentification s'il n'est pas authentifié ;
- ✓ La gestion de session s'effectue coté client grâce à JWT (Json Web Token);
- ✓ Chaque compte créer attendra une validation des administrateurs avant d'être actif, un sms est envoyé à l'utilisateur en cas d'activation.
- ✓ Toute donnée sensible est cryptée ou hachée comme les mots de passe utilisateur;
- ✓ Les sessions inactives ont une durée de vie de 30 minutes ;
- ✓ Aucune action critique n'est exécutable sans confirmation explicite de l'utilisateur;
- ✓ Chaque utilisateur a un rôle précis, ce qui l'empêche d'avoir accès à des informations sensibles et dont il n'a pas les droits.
- ✓ Sauvegardes et restaurations pour rendre l'information disponible et accessible à tout moment.

IV. MISE EN PLACE DE LA BASE DE DONNEES

Lorsque nous travaillons avec un système orienté objet, il existe un décalage entre le modèle objet et la base de données relationnelle. Les SGBDR représentent les données sous forme de tableaux alors que les langages orientés objet, tels que Java ou C#, les représentent sous forme d'un graphe d'objets interconnectés.

ORM (Object-Relational Mapping) est une technique de programmation permettant de convertir des données entre des bases de données relationnelles et des langages de programmation orientés objet tels que Java, C#, etc.

La mise en place de notre base de données s'est faite grâce à l'ORM Hibernate. Quelques avantages de Hibernate :

- ✓ Permet au code métier d'accéder aux objets plutôt qu'aux tables de la base de données.
- ✓ Cache les détails des requêtes SQL de la logique OO (Orienté Objet).
- ✓ Basé sur JDBC.
- ✓ Il n'est pas nécessaire de s'occuper de l'implémentation de la base de données.
- ✓ Entités basées sur des concepts métier plutôt que sur la structure de la base de données.
- ✓ Gestion des transactions et génération automatique des clés.
- ✓ Développement rapide de l'application.

Script de création de la base de données

```
CREATE DATABASE xtrem_analyse_db
WITH
OWNER = xtrem_analyse_user
ENCODING = 'UTF8'
LC_COLLATE = 'French_France.1252'
LC_CTYPE = 'French_France.1252'
TABLESPACE = pg_default
CONNECTION LIMIT = -1;
```

La classe User

```
@AllArgsConstructor
@NoArgsConstructor
@Getter @Setter
@Entity
@Table(name = "users")
public class User implements Serializable {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "id", nullable = false, updatable = false)
  private Long id;
  @Column(name = "user_matricule", nullable = false, length = 50, unique = true)
  private String userMatricule;
  @Column(name = "nom", length = 50)
  private String nom;
  @Column(name = "phone_number")
  private String phoneNumber;
  @Column(name = "prenom", length = 50)
  private String prenom;
  @Column(name = "email", length = 50)
  private String email;
  @Column(name = "address", length = 200)
  private String address;
  @JsonProperty(access = JsonProperty.Access.WRITE_ONLY)
  @Column(name = "password", nullable = false)
  private String password;
  @Column(name = "last_login_date")
  private Date lastLoginDate;
  @Column(name = "last_login_date_display")
  private Date lastLoginDateDisplay;
  @Column(name = "join date")
  private Date joinDate:
  @Column(name = "is_active", length = 1, nullable = false)
  private Boolean isActive;
  @Transient
  private String token;
  @Transient
  private boolean isTokenValid=false;
  @ManyToMany(cascade = CascadeType.PERSIST)
  private List<Role> roles=new ArrayList<>();
```

↓ La classe Accident (Quelques attributs)

```
@AllArgsConstructor
@NoArgsConstructor
@Getter @Setter
@Entity
@Table(name = "accidents")
public class Accident {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "id", nullable = true)
  private Long id;
  @Column(name = "date_accident", nullable = true)
  private LocalDate dateAccident;
  @Column(name = "heure", nullable = true)
  private LocalDateTime heure;
  @Column(name = "zone", nullable = true)
  private String zone;
  @Column(name = "quartier", nullable = true, length = 70)
  private String quartier;
  @Column(name = "intersection_accident", nullable = true)
  private String intersectionAccident;
  @Column(name = "nombre_panneaux_atteints", nullable = true)
  private Integer nombrePanneauxAtteints;
  @Min(value = 0)
  @Column(name = "nombre_balises_atteintes", nullable = true)
  private Integer nombreBalisesAtteintes;
  @Column(name = "nombre_garde_fou_atteints", nullable = true)
  private Integer nombreGardeFouAtteints;
  @OneToMany(orphanRemoval = true, cascade = CascadeType.ALL)
  private List<Image> images = new ArrayList<>();
  @ManyToOne
  private Ville ville;
  @ManyToOne
  private TypeRoute typeRoute;
  @ManyToOne
  private ClassificationRoute classificationRoute;
  @ManyToOne
  private User user;
  @JsonIgnore
  @OneToMany(mappedBy = "accident",
       cascade = CascadeType.ALL,
       orphanRemoval = true)
  private List<Vehicule> vehicules = new ArrayList<>();
  @Column(name = "created_at", nullable = false)
  private LocalDateTime createdAt;
  @Column(name = "update_at", nullable = false)
  private LocalDateTime updateAt;
```

La classe Vehicule

```
@AllArgsConstructor
@NoArgsConstructor
@Getter
@Setter
@Entity
@Table(name = "vehicules")
public class Vehicule {
  @ld
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "id", nullable = true)
  private Long id:
  @Column(name = "immatriculation", nullable = true, length = 20)
  private String immatriculation;
  @Column(name = "pays_immatriculation", nullable = true, length = 20)
  private String paysImmatriculation;
  @Column(name = "genre+", nullable = true, length = 20)
  private String genre;
  @Column(name = "is_premier_choc", nullable = true, length = 1)
  private Boolean isPremierChoc;
  @Column(name = "debut_circulation", nullable = true)
  private String debutCirculation;
  @Column(name = "is controle tech valid", nullable = true, length = 1)
  private Boolean isControleTechValid;
  @Column(name = "is_assurance_valid", nullable = true, length = 1)
  private Boolean isAssuranceValid;
  @Column(name = "etat_general", nullable = true, length = 30)
  private String etatGeneral;
  @Column(name = "etat pneus", nullable = true, length = 30)
  private String etatPneus;
  @Column(name = "etat_feux", nullable = true, length = 40)
  private String etatFeux;
  @Column(name = "type_chargement", nullable = true, length = 40)
  private String typeChargement;
  @Column(name = "volume_chargement", nullable = true, length = 40)
  private String volumeChargement;
  @Column(name = "created_at", nullable = true)
  private LocalDateTime createdAt;
  @Column(name = "update_at", nullable = true)
  private LocalDateTime updateAt;
  @ManyToOne
  private Accident accident:
  @ManyToOne
  private Assurance assurance;
  @JsonIgnore
  @OneToMany(mappedBy = "vehicule", cascade = CascadeType.ALL, orphanRemoval = true)
  private List<PersonneImpliquee> personneImpliquees = new ArrayList<>();
  @JsonIgnore
  @OneToMany(mappedBy = "vehicule", cascade = CascadeType.ALL, orphanRemoval = true)
  private List<PersonneImpliquee> personneImpliquees = new ArrayList<>();
```

PARTIE 4 : EXPLOITATION (Guide d'exploitation)

PLATEFORME DIGITALE DE COLLECTE ET D'ANALYSE D'INFORMATION DES ACCIDENTS CONSTATES SUR UN TERRITOIRE (CAS DU TOGO)

La phase de réalisation et de mise en œuvre nous a permis de mettre en œuvre le système conçu au départ. Le présent document vise alors à décrire la procédure à suivre pour permettre la mise en production de l'application. Nous présenterons dans ce document, les configurations matérielles et logicielles requises pour déployer l'application puis nous exposerons les procédures ainsi que les actions à mener en cas d'erreurs.

I. CONFIGURATION MATERIELLE ET LOGICIELLE

A. Configuration matérielle

Appareil	Ressources minimales	Ressources recommandées
Serveur hébergeant l'API «	Processeur : core i3,2.0GHz	Processeur : core i9,3.0GHz
Xtem-analyse »	Disque dur :500Go	Disque dur :500Go
	<u>RAM</u> : 8 Go	<u>RAM</u> : 16 Go
Serveur hébergeant la	Processeur : core i3,2.0GHz	Processeur : core i9,3.0GHz
plateforme web « Xtem-	Disque dur :500Go	Disque dur :500Go
analyse »	<u>RAM</u> : 8 Go	<u>RAM</u> : 16 Go
Serveur de base de données	Processeur : core i3,2.0GHz	Processeur : core i9,3.0GHz
	Disque dur :500Go	Disque dur :500Go
	<u>RAM</u> : 8 Go	<u>RAM</u> : 16 Go

B. Configuration logicielle

Appareil	Ressources minimales
Serveur hébergeant l'API «Xtem-analyse»	Système d'exploitation : Au choix entre Linux et
	Windows server
Serveur hébergeant la plateforme web	Système d'exploitation : Au choix entre Linux et
«Xtem-analyse»	Windows server
Serveur de base de données	Système d'exploitation : Au choix entre Linux et
	Windows server

II. DEPLOIEMENT ET SUIVI

Le suivi d'une application après son déploiement permet d'éviter au cours de son utilisation, des problèmes qui pourraient entraîner une indisponibilité de cette dernière. Les principales actions à mener pour le suivi d'une application sont le suivi des bugs et la gestion de versions.

1. Suivi en cas d'erreur

Vu la structure dispose d'un développeur informatique en l'occurrence notre maître de stage, l'application pourra être suivi, entretenue et maintenue par ce dernier.

2. Gestions de versions

Un gestionnaire de versions est un système qui enregistre l'évolution d'un fichier ou d'un ensemble de fichiers d'un projet au cours du temps de manière à ce qu'on puisse rappeler une version antérieure d'un fichier ou du projet tout entier. Utiliser un gestionnaire de versions nous permet de suivre les différents changements qui ont été effectué et surtout, nous garantit qu'on puisse revenir à un état « stable » des fichiers à tout moment. Pour notre projet, nous avons utilisé le logiciel Git avec le serveur de gestion de développement logiciel GitHub (tous présentés dans la partie Réalisation et mise en œuvre). Muni de ce guide d'exploitation, l'exploitant est en mesure de mettre en place notre application et a connaissance des principaux moyens de suivi de l'application mis en place. Il faut ensuite former les utilisateurs afin que ceux-ci puissent utiliser pleinement l'application. La documentation qui suivra présentera le guide d'utilisation de notre application et les actions à mener en cas d'erreur.

III. MAINTENANCE

La maintenance concerne la documentation des erreurs qui peuvent survenir dans l'utilisation de l'application et des différentes actions à mener lorsque celles-ci surviennent. Les erreurs les plus courantes sont entre autres les exceptions de type :

Code d'erreur	Nom de l'erreur	Description	Action à mener
403	Unauthorized	Un niveau	Le système
		d'accréditation est	déconnecte
		nécessaire pour avoir	l'utilisateur et lui
		accès à cette	demande de se
		ressource	reconnecter avec le
			rôle requis.
404	Not Found	Ressource introuvable	Indiquer à l'utilisateur
			que la ressource
			demandée est
			introuvable.
401	Bad Request	La requête n'a pas été	Envoie de message
		effectuée car il	d'avertissement ou
		manque	d'erreur.
		desinformations	
		d'authentification	
		valides pour la	
		ressource	
500	Internal Server Error	Erreur interne du	Signaler à l'utilisateur
		serveur	que le serveur est
			temporairement
			indisponible

PARTIE 5: GUIDE D'UTILISATION

I. DESCRIPTION TEXTUELLE DU LOGICIEL

« Xtrem-analyse » est une application de gestion des accidents répertoriés sur le territoire togolais. L'application permet de gérer des opérations telles que : enregistrer les accidents constatés ainsi que les personnes impliquées dans l'accident, les villes, préfectures et régions, les statistiques.

II. PLAN DE NAVIGATION

Nous présenterons ici l'arborescence des menus de notre plateforme web.

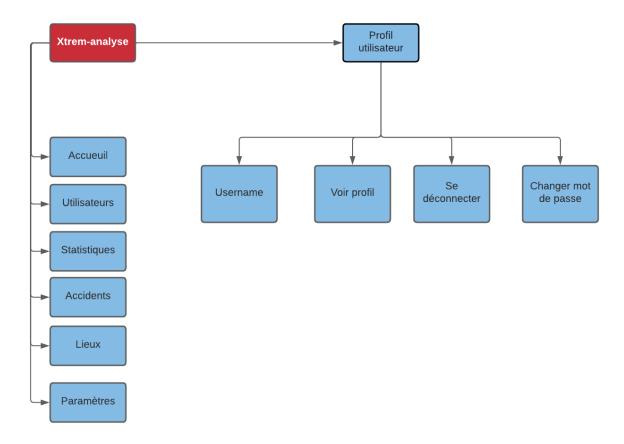


Figure 36: Plan de navigation

III. PRESENTATION DES DIFFERENTS INTERFACES DE L'APPLICATION

Nous considérons le scenario nominal ou les conditions sont bien remplies et les applications sont parfaitement opérationnelles :

- ✓ Le système doit être fonctionnel
- ✓ L'utilisateur doit être connecté à internet
 - A. Application web
- Interface de connexion

Une authentification est requise avant d'effectuer une quelconque action sur l'application.

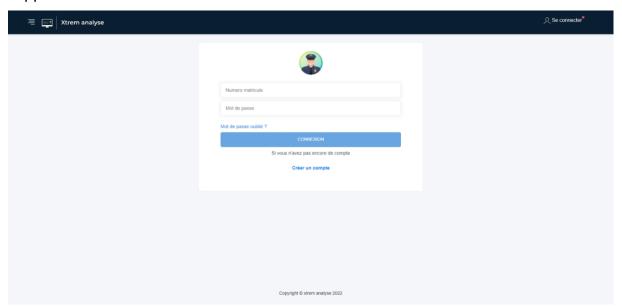


Figure 37: Page de connexion

Les deux champs permettent à l'utilisateur de saisir respectivement son matricule de connexion et son mot de passe

Interface de gestion des utilisateurs

Cette interface est destinée au SUPER ADMINISTRATEUR et aux Administrateurs. Elle permet de gérer les utilisateurs en l'occurrence : ajouter et supprimer des rôles, modifier les informations, et modifier le statut (Compte actif ou compte inactif) de l'utilisateur.

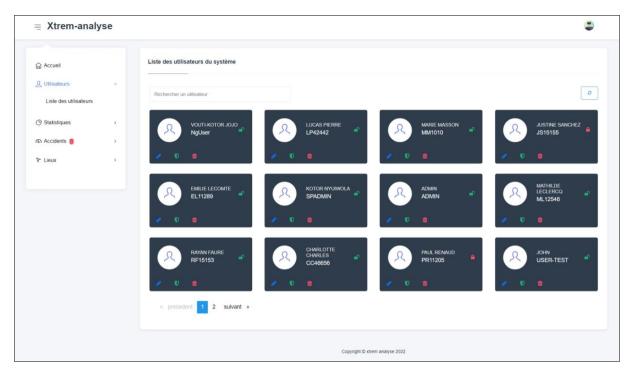


Figure 38: Page de gestion des utilisateurs

❖ Page de visualisation des coordonnées géographiques d'un accident
Cette vue présente à l'Observateur la vue géographique de l'accident sélectionné.

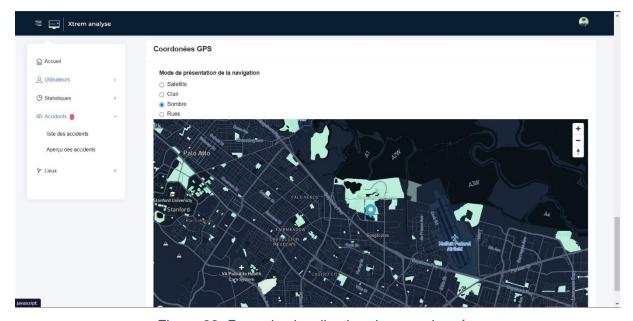


Figure 39: Page de visualisation des coordonnées

Page de détails d'un accident

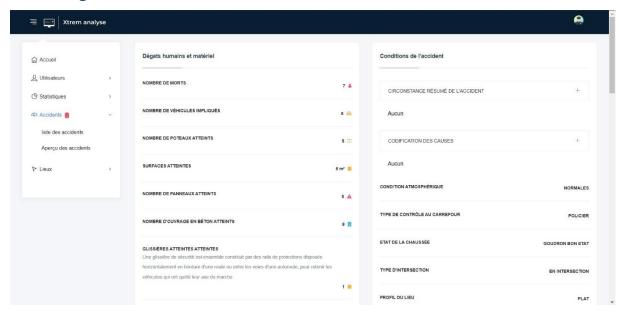


Figure 40: Page de détails d'un accident

Page de présentation des statistiques

Sur cette page, l'observateur pourrait avoir une vue sur les statistiques

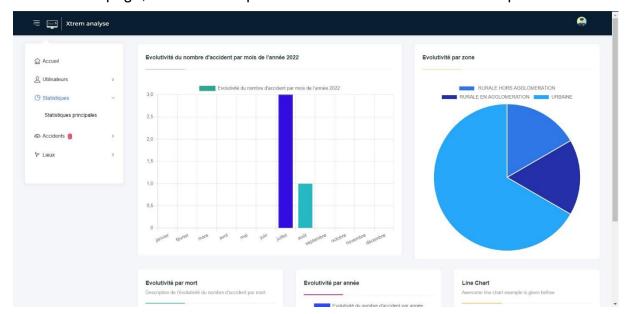


Figure 41: Page de visualisation des statistiques

B. Application mobile

❖ Ecran de connexion et écran d'enregistrement d'un accident

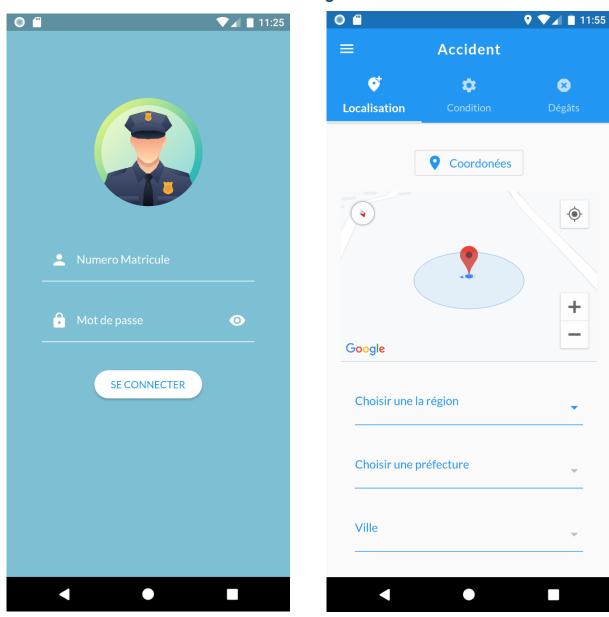


Figure 42: Page de connexion et page d'enregistrement d'un accident

❖ Ecran d'enregistrement d'un véhicule

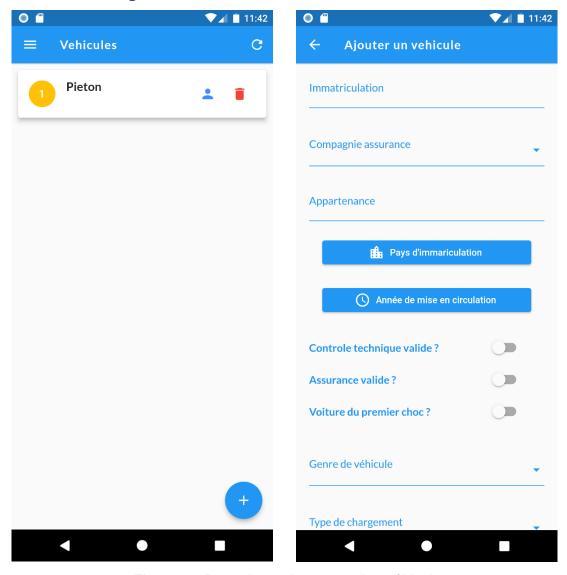


Figure 43: Page d'enregistrement d'un véhicule

Ecran d'enregistrement d'une personne

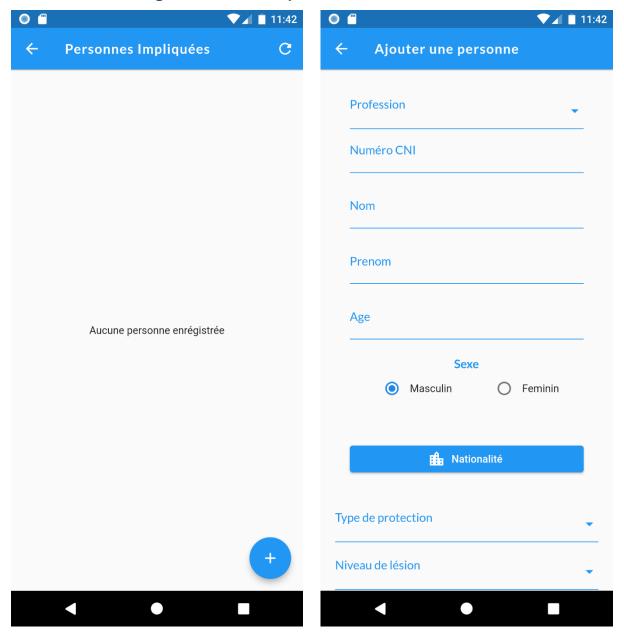


Figure 44: Page d'enregistrement d'une personne

IV. PRESENTATION DE QUELQUES ETATS

La liste des utilisateurs du système

Cet état est généré par notre application en fichier PDF ou HTML au choix.

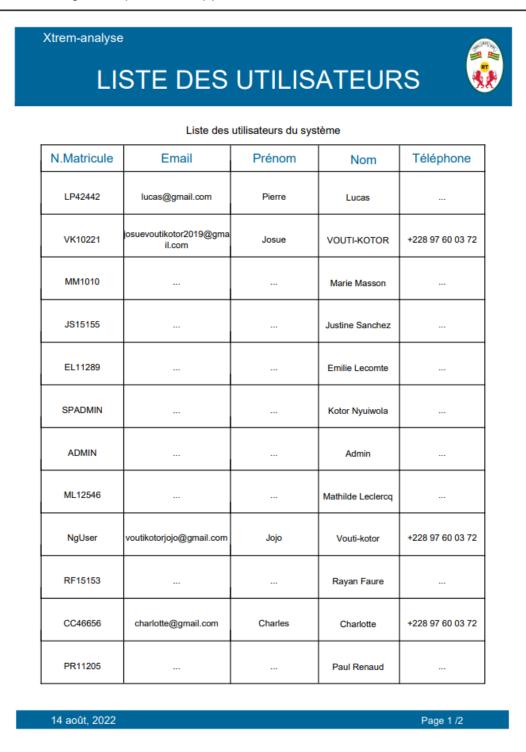


Figure 45: Etat, liste des utilisateurs

Xtrem-analyse

Liste des accidents



Accidents selon les conditions atmospherique

Date	N. morts	N. véhicules	Quartier	Route
2022-07-26	7	5	Atigangomé	Boulevard du 30 Aout
2022-07-28	2	2	Agoue	Boulevard
2016-07-27	6	9	Adidogome	Segbe
2022-08-04	9	2	Logope	Boulevard
2022-07-26	5	4	Agougan	sentier

14 août, 2022 Page 1 /1

Figure 46: Etat, Listes des accidents en fonction des conditions atmosphériques

CONCLUSION

Ce stage a été très enrichissant pour moi car il m'a permis de découvrir dans le détail le secteur de l'ingénierie informatique ses acteurs, contraintes et il m'a permis de participer concrètement à ses enjeux au travers de mes missions variées comme celle du développement de l'application **Xtrem-analyse** que j'ai particulièrement apprécié. Ce stage m'a aussi permis de comprendre que les missions créatives n'étaient pas les plus adaptées pour moi et je préfère m'orienter vers l'analyse et la conception qui me conviennent mieux. Fort de cette expérience et en réponse à ses enjeux, j'aimerai beaucoup par la suite essayer de m'orienter via un prochain stage, vers le secteur de l'Administration des Systèmes et Réseaux avec des acteurs de petites tailles, et un important développement d'avenir. L'objectif de ce stage était la mise en place d'une plateforme de collectes et d'analyse des informations des accidents constatés.

La réalisation s'est faite en trois étapes (décrites plus haut) dont la plus intéressante était l'implémentation de la solution. J'ai donc rempli les objectifs fixés, à savoir DEVELOPPER UNE APPLICATION DE GESTION DES ACCIDENTS.

V. BIBLIOGRAPHIE INDICATIVE

- ❖ UML 2 par la pratique. Edition EYROLLES 5eme édition de Pascal ROQUES
- ❖ Gestion de projets 2 Tracks Unified Process d'El MAZOUZI Nadia, FILALI Abderrahmane et TAMAGNON Olivier
- Comprendre Spring-boot, Mohamed Youssfi

VI. WEBOGRAPHIE INDICATIVE

Sites web consultés	Période
https://stackoverflow.com	16 Mai 2022 au 13 Août 2022
https://www.wikipedia.com	16 Mai 2022 au 13 Août 2022
https://www.youtube.com/	16 Mai 2022 au 13 Août 2022
https://pub.dev/	16 Mai 2022 au 13 Août 2022
https://www.baeldung.com/	16 Mai 2022 au 13 Août 2022
https://github.io	Dernière consultation : 11 Août 2022

VII. NOTES DE COURS

- ❖ Analyse avec le langage de modélisation UML2 de Pierre Gerard (2020-2021)
- ❖ Implémentation des bases de données de M Gbodui Roland (2020-2021)
- Cours de Gestion de projets Informatiques de M. AMEVOR Kossi (2021-2022);

VIII. ANCIENS MEMOIRES CONSULTES

❖ VOUTI-KOTOR Nyuiwola (2020-2021), Mise en place d'une plateforme de réservation de place pour des évènements

IX. DOCUMENTS ANNEXES

- ❖ Le bulletin d'analyse des accidents fournis par le ministère de la sécurité
- ❖ La fiche de statistiques de l'OMS (Organisation Mondiale de la Santé)
- ❖ Le canevas de rédaction de la mémoire fourni par l'IAI TOGO

X. TABLE DES MATIERES

l.	PAC	GE DE GARDE	i		
II.	DEDICACESi				
III.	REMERCIEMENTSi				
IV.	SON	MMAIRE	. iv		
٧.	RESUMEvi				
VI.	GLC	DSSAIRE	vii		
VII.	LIST	TE DES FIGURES	√iii		
VIII.	LIST	TE DES TABLEAUX	x		
IX.	LIST	TE DES PARTICIPANTS	. Xi		
INT	ROD	UCTION	. 1		
PAF	RTIE	1 : CAHIER DES CHARGES	. 2		
l.	PRE	SENTATIONS	. 3		
Α	•	Brève présentation de l'IAI-TOGO	. 3		
	1.	Aspect formel	. 3		
	2.	Aspect géographique	. 4		
В		Présentation de la structure ODDA Technology	. 5		
	1.	Statut	. 5		
	2.	Missions	. 5		
	3.	Activités	. 5		
	4.	Quelques réalisations	. 6		
	5.	Service d'accueil	. 6		
	6.	Plan de localisation	. 7		
II.	THE	EME DU STAGE	. 8		
Α	•	Présentation du sujet	. 8		
В		Problématique du sujet	. 8		
С		Intérêt du sujet	. 8		
	1.	Objectifs	. 8		
	2.	Résultats attendus	. 9		
III.	ETU	JDE DE L'EXISTANT	10		
IV.	CRI	TIQUE DE L'EXISTANT	12		
V. PROPOSITIONS ET CHOIX DE SOLUTIONS					
Α		Evaluation technique des solutions	13		
	1.	Première solution	13		
	2	Deuxième solution	14		

PLATEFORME DIGITALE DE COLLECTE ET D'ANALYSE D'INFORMATION DES ACCIDENTS CONSTATES SUR UN TERRITOIRE (CAS DU TOGO)

	B.	Evaluation financière des solutions	15
	1.	Première solution	15
	2.	Deuxième solution	17
	C.	Choix de solution	18
VI.	PLA	NNING PREVISIONNEL DE REALISATION	19
PΑ	RTIE	2 : ANALYSE ET CONCEPTION	20
I.	CHO	DIX DE LA METHODE D'ANALYSE ET JUSTIFICATION	21
	A.	Justification de de l'approche objet	21
	B.	Langage de modélisation UML	21
	1.	Définition	21
	2.	Pourquoi UML ?	22
	3.	Les diagrammes UML	22
	C.	Le processus 2TUP	24
	1.	Présentation	24
	2.	Justification du choix du processus 2TUP	26
ΙΙ.	CHO	DIX DE L'OUTIL DE MODELISATION ET JUSTIFICATION	28
III.	ETL	JDE DETAILLEE DE LA SOLUTION	29
	A.	Diagramme de cas d'utilisation	29
	1.	Les acteurs	29
	2.	Cas d'utilisation identifiés	29
	B.	Diagramme de séquence	37
	1.	Quelques concepts	37
	2.	Quelques diagrammes de séquences système de notre projet	38
	3.	Quelque diagramme de séquence détaillé de notre système	41
	C.	Diagramme d'activité	44
	D.	Diagramme de classe	48
	E.	Diagramme de paquetage	50
	F. D	Piagramme de déploiement	50
PΑ	RTIE	3 : REALISATION ET MISE EN OEUVRE	52
I.	MA	TERIELS ET LOGIIELS UTILISES	53
	A.	Choix du matériel	53
	В.	Choix du logiciel utilisé	53
	1.	Langage de programmation et technologies utilisées	53
II.	ARC	CHITECTURE MATERIEL ET LOGICIEL DE L'APPLICATION	63
	Δ	Architecture globale	63

PLATEFORME DIGITALE DE COLLECTE ET D'ANALYSE D'INFORMATION DES ACCIDENTS CONSTATES SUR UN TERRITOIRE (CAS DU TOGO)

В		Architecture matérielle	63
C		Architecture logicielle	64
III.	SEC	CURITE DE L'APPLICATION	65
IV.	MI	SE EN PLACE DE LA BASE DE DONNEES	66
PAR	RTIE	4 : EXPLOITATION (Guide d'exploitation)	70
I.	СО	NFIGURATION MATERIELLE ET LOGICIELLE	72
Α		Configuration matérielle	72
В	•	Configuration logicielle	72
II.	DE	PLOIEMENT ET SUIVI	73
	1.	Suivi en cas d'erreur	73
	2.	Gestions de versions	73
III.	MA	AINTENANCE	74
I.	DE:	SCRIPTION TEXTUELLE DU LOGICIEL	76
II.	PLA	AN DE NAVIGATION	76
III.	PRI	ESENTATION DES DIFFERENTS INTERFACES DE L'APPLICATION	77
Α		Application web	77
В		Application mobile	80
CON	ICLU	JSION	85
IV.	BIB	SLIOGRAPHIE INDICATIVE	I
V.	WE	BOGRAPHIE INDICATIVE	. II
VI.	NO	TES DE COURS	Ш
VII.	AN	CIENS MEMOIRES CONSULTES	IV
VIII.	DO	CUMENTS ANNEXES	. V