

REPUBLIQUE TOGOLAISE

Travail-Liberté-Patrie

PRESIDENCE DE LA REPUBLIQUE

MINISTRE DE LA
PLANIFICATION DU
DEVELOPPEMENT ET DE LA
COOPERATION



HI TECH
Informatique

Institut Africain d'Informatique,
Représentation du TOGO (IAI-TOGO)

07 BP 12456 Lomé 07, TOGO

TEL: (+228) 22 20 47 00

E-mail: iaitogo@iai-togo.tg

Site web: www.iai-togo.tg

HI TECH INFORMATIQUE

TEL : (+228) 93 65 55 25

(+228) 90 12 18 05

E-mail : contact@hitech-services.com

Site web: www.hitech-services.com

RAPPORT DE STAGE PRATIQUE EN ENTREPRISE

TYPE DE STAGE : STAGE DE PROGRAMMATION

MISE EN PLACE D'UNE APPLICATION MOBILE DE SIGNALISATION D'INCIDENTS AVEC SYSTEME D'INFORMATION GEOGRAPHIQUE

Période : Du 08 Juillet au 30 Août 2024

Rédigé et présenté par :

AMEDRO Nathanaël Essonissi

Etudiant en Deuxième année Tronc commun

W

SUPERVISEUR

**M. AGBETI Kodjo Akoro
Bitantchi**

Représentant National de
l'IAI-TOGO

MAITRE DE STAGE

M. POUH-PEKA Crésus

Développeur d'application

REMERCIEMENTS

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 guidé mes réflexions et ont accepté de répondre à mes questions durant mes recherches. Mes remerciements vont :

- À M. AGBETI Kodjo Akoro Bitantchi, Directeur Général du CENETI et Représentant National de l'IAI-TOGO ;
- À M. AMEYIKPO Kossi Nicolas, Directeur des Affaires Académiques et de la Scolarité de l'IAI-TOGO ;
- Aux membres de l'administration de l'IAI-TOGO, pour l'encadrement rigoureux ;
- À Mme. HISSEIN Virginia Directeur Générale de HITECH pour l'acceptation notre demande de stage au sein de son entreprise
- À mon maitre de stage, M. POUH-PEKA Pouwèdéou Crésus, Développeur d'application ;
- À mon superviseur, M. AGBETI Kodjo Akoro Bitantchi, Représentant National de l'IAI-TOGO;
- À M. AMEDRO Emmanuel et Mme AMEDRO Rachel mes parents pour leur soutien, et leur sacrifice.

SOMMAIRE

REMERCIEMENTS	i
SOMMAIRE	ii
LISTES DES FIGURES.....	iii
LISTES DES TABLEAUX.....	iv
INTRODUCTION	2
PARTIE1 : CAHIER DES CHARGES	3
1.1 Présentation du sujet	3
1.2 Problématique du sujet.....	3
1.3 Intérêt de sujet	4
PARTIE 2 : ANALYSE ET CONCEPTION.....	6
1.5 Étude de l'Existant	6
1.6 Critique de l'Existant	6
1.7 Planning prévisionnel et de réalisation	6
1.8 Etude détaillée de la solution.....	8
Partie 3 : REALISATION ET MISE EN ŒUVRE	19
1.9 Matériels et logiciels utilisés	19
1.10 Logiciels utilisés	19
1.11 Sécurité de l'application.....	27
1.12 Evaluation financière de la solution	27
1.13 Présentation de l'application	29
CONCLUSION	35
BIBLIOGRAPHIE INDICATIVE	I
WEBOGRAPHIE INDICATIVE	II
DOCUMENTS ANNEXES	III
TABLES DES MATIERES	IV

LISTES DES FIGURES

Fig. 1 : Logo de UML	9
Fig. 2: Logo de powerAMC	9
Fig. 3: Diagramme de cas d'utilisation	11
Fig. 4:Diagramme d'activité de «s'authentifier».....	13
Fig. 5: Diagramme d'activité de «signaler un incident».....	13
Fig. 6:Diagramme d'activité «valider un signalement».....	14
Fig. 7: Diagramme de séquence s'authentifier	15
Fig. 8 : Diagramme du cas d'utilisation crée un compte	16
Fig. 9 Diagramme de classe	17
Fig. 10: Logo Git.....	20
Fig. 11 :Logo GitHub	20
Fig. 12 : Logo Flutter.....	21
Fig. 13: Logo de Dart.....	22
Fig. 14: Logo de Firebase	23
Fig. 15: Logo de open street map.....	24
Fig. 16 : Logo de OpenWheaterMap	26
Fig. 17 : Logo VS Code	26
Fig. 18 : Exemple de règles	27
Fig. 19 : Code pour création de champ pour les utilisateur à la création de compte.....	29
Fig. 20 : Exemple de champ d'un utilisateur	30
Fig. 21 : Code pour création de champ pour les incidents au signalement d'un incident.....	30
Fig. 22 : Exemple de champ pour les incidents	30
Fig. 23 : Plan de navigation.....	31
Fig. 24 : Page de création de compte	31
Fig. 25 : Page de connexion.....	32
Fig. 26 : Dashboard.....	32
Fig. 27 : Liste des utilisateurs	33
Fig. 28 : Graphes du nombre d'incidents par type	33
Fig. 29 : Liste des incidents.....	34

LISTES DES TABLEAUX

Tableau 1:Planning Prévisionnel	7
Tableau 2: Les acteurs et leurs cas d'utilisation	10
Tableau 3: Cas d'utilisation s'authentifier	12
Tableau 4: Cas d'utilisation signaler un incident.....	12
Tableau 6 : Matériels utilisés.....	19
Tableau 7 : Coût matériel	28
Tableau 8 : Coût d'exploitation	28
Tableau 9 : Coût de conception	28
Tableau 10 : Coût total	29

PARTIE 1 : CAHIER DES CHARGES

INTRODUCTION

L'informatique, définie comme la science du traitement automatique et rationnel de l'information, joue un rôle crucial dans les domaines techniques, économiques et sociaux. Elle englobe la conception, le développement, la gestion et l'utilisation de systèmes informatiques, qu'il s'agisse de matériel (hardware) ou de logiciels (software). En tant que pilier fondamental de la société moderne, l'informatique facilite l'innovation, améliore la productivité et soutient le développement technologique.

Au sein de l'Institut Africain d'Informatique (IAI), et plus particulièrement de sa représentation au Togo, cette discipline est enseignée avec rigueur et ambition. Fondé en 1971 et présent au Togo depuis 2002, l'IAI-TOGO forme des ingénieurs de travaux informatiques et des spécialistes en génie logiciel, systèmes et réseaux. L'objectif est de doter les étudiants des compétences nécessaires pour répondre aux besoins croissants en professionnels qualifiés dans le domaine de l'informatique.

Dans le cadre de ce cursus, un stage pratique est organisé à la fin de la deuxième année d'études pour permettre aux étudiants de mettre en application leurs connaissances théoriques et de s'immerger dans le monde professionnel. Ce rapport présente l'expérience de notre stage, réalisé du 08 juin au 30 août au centre HI TECH informatique.

Durant ce stage, nous avons travaillé sur la création d'une application permettant aux utilisateurs de signaler divers incidents en temps réel, facilitant ainsi une réponse rapide et efficace aux situations d'urgence. Ce document se divise en trois parties principales : le cahier des charges, la phase de pré-programmation, et la réalisation et mise en œuvre de la solution développée. L'objectif de ce rapport est de détailler le processus suivi, les défis rencontrés et les solutions apportées, tout en soulignant l'importance de l'informatique dans la gestion efficace des incidents.

PARTIE1 : CAHIER DES CHARGES

Dans la première partie de notre document, ou cahier des charges, il est question de la formalisation et de la clarification des besoins. Le cahier de charge est le résultat de la phase d'analyse des besoins et contient obligatoirement les résultats attendus en termes de spécifications fonctionnelles et non fonctionnelles. Il comporte la description du cadre de stage, les besoins et exigences des utilisateurs futurs, les fonctionnalités attendues ainsi que les contraintes techniques. En grandes lignes de ce document, nous aurons :

- La présentation du thème de stage,
- La problématique du sujet,
- L'intérêt du sujet,
- Les objectifs et
- Résultats attendus.

1.1 Présentation du sujet

L'évolution rapide de la technologie mobile offre des opportunités uniques pour améliorer la gestion des incidents en temps réel. La signalisation rapide et efficace des incidents est cruciale pour assurer la sécurité et le bien-être des communautés. Ce projet vise à concevoir et développer une application mobile intuitive pour la signalisation des incidents, facilitant ainsi une réponse rapide et coordonnée des services d'urgence et des autorités locales.

1.2 Problématique du sujet

Le développement d'une application mobile de signalisation des incidents soulève plusieurs défis techniques et organisationnels. Il s'agit de créer un outil efficace et intuitif capable de répondre aux besoins des utilisateurs en temps réel tout en garantissant la sécurité et la confidentialité des données. La conception de l'application doit intégrer une interface utilisateur conviviale, permettant une navigation fluide et une utilisation sans effort. Par ailleurs, l'application doit être dotée de fonctionnalités robustes pour la gestion des notifications en temps réel, la vérification et la validation des incidents signalés, et l'intégration avec les systèmes existants des services d'urgence et des autorités locales.

- Comment concevoir et développer une application mobile de signalisation des incidents qui soit efficace, intuitive et capable de répondre aux besoins des utilisateurs en temps réel?
- Quels types d'incidents doivent être signalés via l'application ?
- Qui sont les utilisateurs cibles de cette application?
- Quelles sont les attentes des utilisateurs en matière de fonctionnalités et d'interface?

Voilà les questions auxquelles nous nous efforcerons de répondre tout le long de la conception de l'application.

1.3 Intérêt de sujet

1.3.1 Objectifs

L'objectif visé par ce projet est la mise en place d'une application qui permettra de :

- **Faciliter la signalisation:**

Permettre aux utilisateurs de signaler divers incidents (accidents, incendies, situations d'urgence) rapidement et facilement.

- **Réponse rapide:**

Assurer une transmission immédiate des informations aux services compétents pour une intervention rapide.

- **Interface intuitive:**

Développer une interface utilisateur simple et conviviale pour garantir une utilisation facile par un large public.

- **Sécurité des données:**

Mettre en place des mesures de sécurité robustes pour protéger les informations sensibles des utilisateurs.

1.3.2 Résultats

Une fois l'application créée l'utilisateur :

- Les incidents sont signalés et situés
- Des notifications sont reçues par rapport aux incidents signalés
- Des incidents sont commentés.

L'administrateur:

- Les utilisateurs sont gérés
- La signalisation des incidents est gérée.

PARTIE 2 : PRE-PROGRAMMATION

PARTIE 2 : ANALYSE ET CONCEPTION

1.5 Étude de l'Existant

Actuellement, plusieurs applications mobiles de signalisation des incidents existent sur le marché, chacune offrant des fonctionnalités variées pour répondre aux besoins des utilisateurs. Par exemple, des applications comme "PulsePoint" permettent aux citoyens de signaler des urgences médicales et d'autres types d'incidents aux services de secours.

"Citizen" est une autre application populaire qui permet aux utilisateurs de signaler des incidents en temps réel et d'obtenir des alertes sur les événements se déroulant à proximité. Ces applications utilisent des technologies telles que la géolocalisation, les notifications push et les intégrations avec les services d'urgence pour assurer une communication rapide et efficace. Elles proposent également des interfaces utilisateurs intuitives pour faciliter l'usage, même pour les utilisateurs moins technophiles.

1.6 Critique de l'Existant

Bien que ces applications soient largement utilisées et offrent des fonctionnalités utiles, elles présentent certaines limitations et défis. Par exemple, la précision de la géolocalisation peut parfois être compromise, entraînant des erreurs dans la localisation des incidents signalés. De plus, les utilisateurs peuvent rencontrer des difficultés avec les notifications en temps réel, notamment en raison de problèmes de connectivité ou de latence.

En termes de sécurité, bien que des mesures soient en place, il existe des préoccupations concernant la confidentialité des données personnelles des utilisateurs. Enfin, l'adoption et l'engagement des utilisateurs varient considérablement, certains utilisateurs étant réticents à adopter ces technologies en raison de préoccupations liées à la vie privée ou à la complexité perçue de l'application. Par conséquent, il est crucial de développer une application qui non seulement améliore ces aspects, mais aussi propose des solutions innovantes pour surmonter ces défis.

1.7 Planning prévisionnel et de réalisation

➤ Planning prévisionnel

Tâche	Début	Fin	Durée (Jours)
Intégration au sein de l'entreprise	01/07/2024	03/07/2024	03
Etude du thème (recueil d'informations)	04/07/2024	04/07/2024	01
Rédaction du cahier des charges	05/07/2024	06/07/2024	02

Élaboration et création des divers diagrammes de modélisation objet pour la conception.	07/07/2024	08/07/2024	02
Conception des interfaces utilisateurs	09/07/2024	12/07/2024	04
Développement des fonctionnalités principales (inscription, connexion, rechercher salon, réservation...)	13/07/2024	07/08/2024	25
Test et correction des bugs	08/08/2024	17/08/2024	11
Finalisation et amélioration du document	18/08/2024	31/08/2024	14

Tableau 1:Planning Prévisionnel

➤ Planning de réalisation

Tâche	Début	Fin	Durée (Jours)
Intégration au sein de l'entreprise	08/07/2024	10/07/2024	03
Etude du thème (recueil d'informations)	10/07/2024	12/07/2024	02
Rédaction du cahier des charges	13/07/2024	14/07/2024	02
Élaboration et création des divers diagrammes de modélisation objet pour la conception.	15/07/2024	16/07/2024	02
Conception des interfaces utilisateurs	17/07/2024	19/07/2024	03
Développement des fonctionnalités principales (inscription, connexion, rechercher salon, réservation...)	19/07/2024	19/08/2024	31
Test et correction des bugs	19/08/2024	22/08/2024	04
Finalisation et amélioration du document	22/08/2024	31/08/2024	09

Tableau 2 : Tableau du planning de réalisation

Entre ces 2 tableaux il y a plusieurs différences notables :

Décalage des Dates: Le second tableau montre des dates de début et de fin qui sont systématiquement décalées par rapport au premier tableau, probablement en réponse à des contraintes ou à une révision du planning initial.

Durée des Tâches: Les durées des tâches sont également modifiées dans certains cas. Par exemple, la tâche de développement des fonctionnalités principales dure plus longtemps dans le second tableau (31 jours) que dans le premier (25 jours).

Ajustement des Tâches: Le deuxième tableau semble être une version révisée du premier, avec des ajustements apportés aux durées des tâches pour mieux correspondre à la réalité du projet.

En somme, le second tableau peut être interprété comme une révision du premier, intégrant des ajustements basés sur une meilleure compréhension des besoins du projet et des contraintes temporelles.

1.8 Etude détaillée de la solution

1.8.1 Etude préliminaire de la solution

L'étude préliminaire est la première étape dans notre processus de développement, étape durant laquelle nous repérons les besoins fonctionnels et opérationnels, en utilisant des textes ou des diagrammes. Cette étude préliminaire nous permet de distinguer les besoins fonctionnels des besoins techniques.

En premier nous parlerons des outils d'analyse et de modélisation qui nous ont permis la réalisation des diagrammes ensuite nous identifierons les acteurs de notre système ensuite nous présenterons les différents diagrammes qui ont été nécessaires pour l'élaboration ou la conception de notre application.

1.8.1.1 Choix des outils d'analyse

➤ UML

L'outil d'analyse utilisé est UML. Le langage de modélisation unifié, en anglais Unified Modeling Language (UML) est un langage de modélisation graphique à base de pictogrammes (dessins figuratifs stylisés ayant fonction de signe) conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

UML est un langage formel, défini par un méta modèle. Le méta modèle d'UML décrit de manière très précise tous les éléments de modélisation (les concepts véhiculés et manipulés par le langage) et la sémantique de ces éléments (leur définition et le sens de leur utilisation). C'est en cela qu'on dit qu'UML normalise les concepts objet. Un méta modèle permet de limiter les ambiguïtés et encourage la construction d'outils. Il permet aussi de classer les

différents concepts du langage (selon leur niveau d'abstraction ou leur domaine d'application) et expose ainsi clairement sa structure



Fig. 1 : Logo de UML

1.8.1.2 Présentation de L'outils de modélisation

Il existe une multitude d'outils de modélisation parmi lesquelles nous pouvons citer : DB Designer, StarUML, Draw.io ...

Pour réaliser ce projet, nous avons utilisé Power AMC POWER AMC qui est l'un des premiers outils qui permet d'élaborer des modèles de données que cela soit MERISE, UML ou autre, de manière graphique et de les implémenter quel que soit le SGBD et ce de manière automatique. De même, l'outil permet de modéliser les processus métiers.

Avantages

- Power AMC est un outil simple à utiliser. Le déploiement d'un poste suffit à rendre l'outil efficient.
- L'outil fonctionne nativement avec tous les SGBD courants du marché (ORACLE, SQL SERVEUR, DB2/UDB).
- L'outil permet une documentation des développements.
- L'outil permet une rétro-documentation de l'existant.
- L'outil génère des graphiques exportables et importables facilement via un format XML.

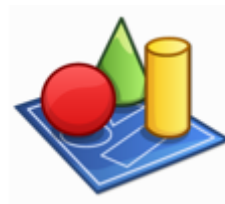


Fig. 2: Logo de powerAMC

1.8.1.3 Acteurs et rôles

Un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec notre système. L'acteur est toujours externe au système modélisé. Dans notre système nous avons précisément besoin de 2 acteurs : Utilisateurs et Administrateur.

1.8.1.4 Captures des besoins fonctionnels

a. Identification des cas utilisations

Un cas d'utilisation, également appelé bloc fonctionnel qui décrit comment un système est utilisé de manière utile ou précieuse pour les acteurs concernés. Il représente un ensemble d'actions que le système exécute en interaction avec les acteurs pour atteindre un objectif spécifique. Les cas d'utilisation permettent de définir les exigences fonctionnelles d'un système en se plaçant du point de vue de l'utilisateur final et en utilisant son langage.

ACTEURS	CAS D'UTILISATION
Utilisateur	<ul style="list-style-type: none">- Créer un compte- S'authentifier- Recevoir notification- Gérer incident:<ul style="list-style-type: none">• Ajouter incident• Annuler incident• Modifier incident
Administrateur	<ul style="list-style-type: none">- Gérer utilisateur:<ul style="list-style-type: none">• Suspendre utilisateur• Supprimer utilisateur- Valider signalement- Générer rapport:<ul style="list-style-type: none">• Consulter statistique

Tableau 3: Les acteurs et leurs cas d'utilisation

1.8.1.4.1 Diagramme des cas d'utilisations

Un diagramme de cas d'utilisation est un diagramme représentant l'interaction entre les acteurs et le système, et leur influence sur celui-ci.

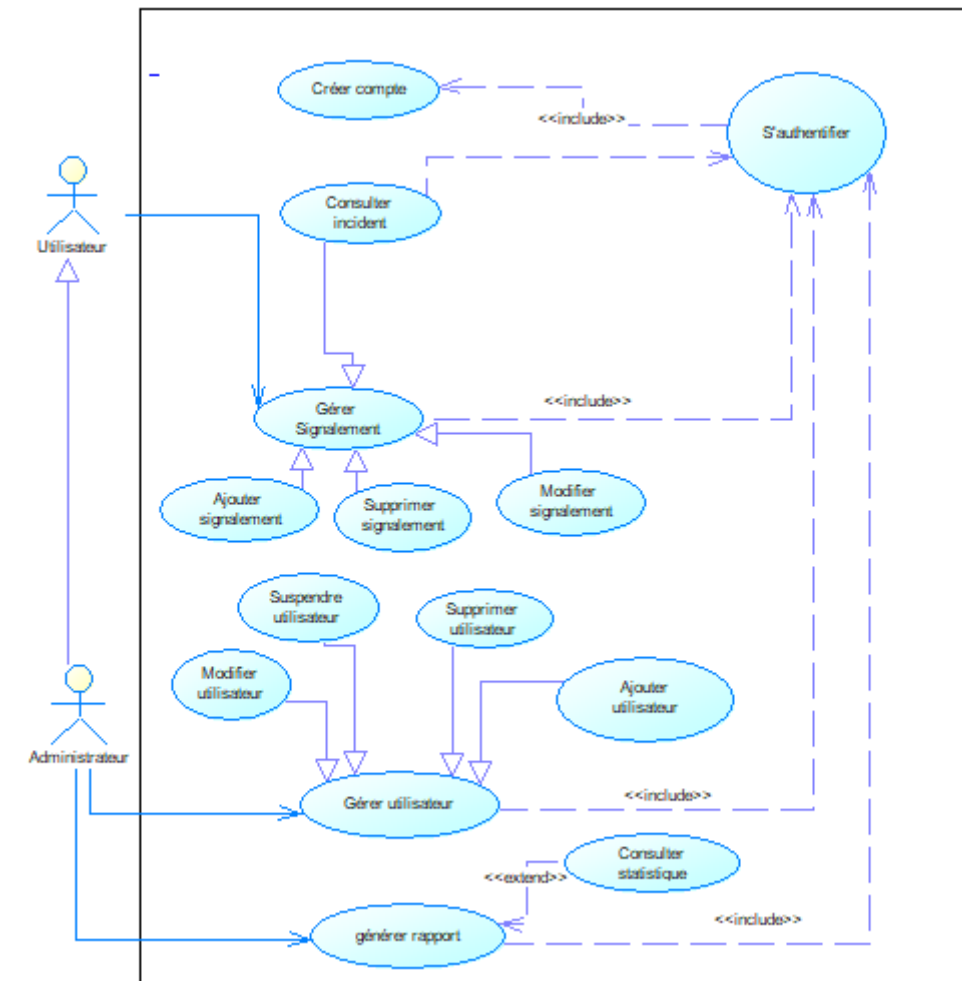


Fig. 3: Diagramme de cas d'utilisation

b. Description des cas d'utilisation

La description des cas utilisations expose de façon détaillé le dialogue entre les acteurs et les cas utilisation.

➤ Cas d'utilisation : « S'authentifier »

1-IDENTIFICATION	
-Titre: S'authentifier	
-Résumé: permet de certifier l'identité de l'utilisateur	
-Acteur: Utilisateur	
-Date de création : 16/07/2024	
-Date de mise à jour :	
-Version : 1.0	
-Responsable : AMEDRO Nathanaël	
2-DESCRIPTION DES SCENARIOS	
Pré-conditions :	
S'inscrire au préalable et avoir un compte	
Scénario nominal	Scénario alternatifs

1. L'utilisateur accède à la page d'authentification.	Les données reçues par le serveur sont erronées.
2. Le système lui affiche le formulaire d'authentification.	Ce scénario démarre au point 4 du scénario nominal lorsque les données reçues par le système sur les champs requis ne sont pas valides.
3. L'utilisateur saisit les informations requises et valide.	Le système envoie un message d'alerte : " Nom d'utilisateur ou mot de passe incorrect "
4. Le système vérifie les informations saisies	Le scénario reprend au point 3 du scénario nominal
Post-conditions : L'utilisateur est authentifié par le système.	

Tableau 4: Cas d'utilisation s'authentifier

➤ Cas d'utilisation: « Signaler un incident »

1-IDENTIFICATION	
- Titre: Signaler un incident - Résumé: l'utilisateur signal un incident qu'il a remarqué - Acteur: Utilisateur - Date de création : 16/07/2024 - Date de mise à jour : - Version : 1.0 - Responsable : AMEDRO Nathanaël	
2-DESCRIPTION DES SCENARIOS	
Pré-conditions : L'utilisateur c'est authentifié L'utilisateur est connecté à internet	
Scénario nominale	Scénario alternatifs
1. L'utilisateur remarque ou assiste à un incident 2. L'utilisateur ouvre l'application et clique sur le bouton signaler incident. 3. Le système démarre une page ou l'utilisateur peut prendre en photo, décrire, et situer l'incident sur une carte. 4. L'utilisateur sur le bouton envoyer	Lors de la signalisation de l'incident l'utilisateur remarque que l'incident est en train d'être pris en charge, il décide donc d'appuyer sur le bouton annuler
Post-conditions : L'utilisateur à signaler la catastrophe.	

Tableau 5: Cas d'utilisation signaler un incident

c. Diagramme d'activité

➤ S'authentifier

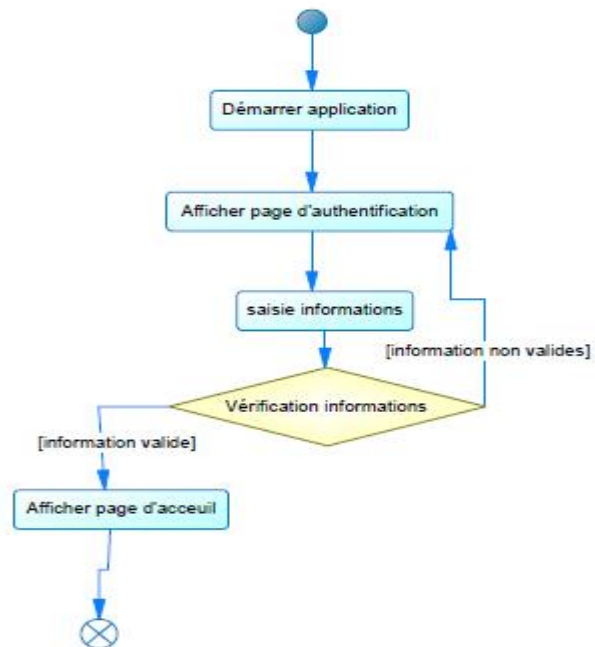


Fig. 4:Diagramme d'activité de «s'authentifier»

➤ Signaler un incident

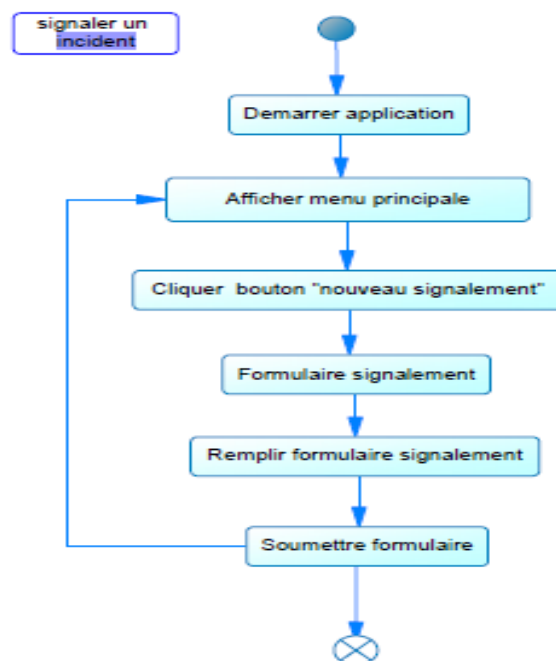


Fig. 5: Diagramme d'activité de «signaler un incident»

➤ Valider un signalement

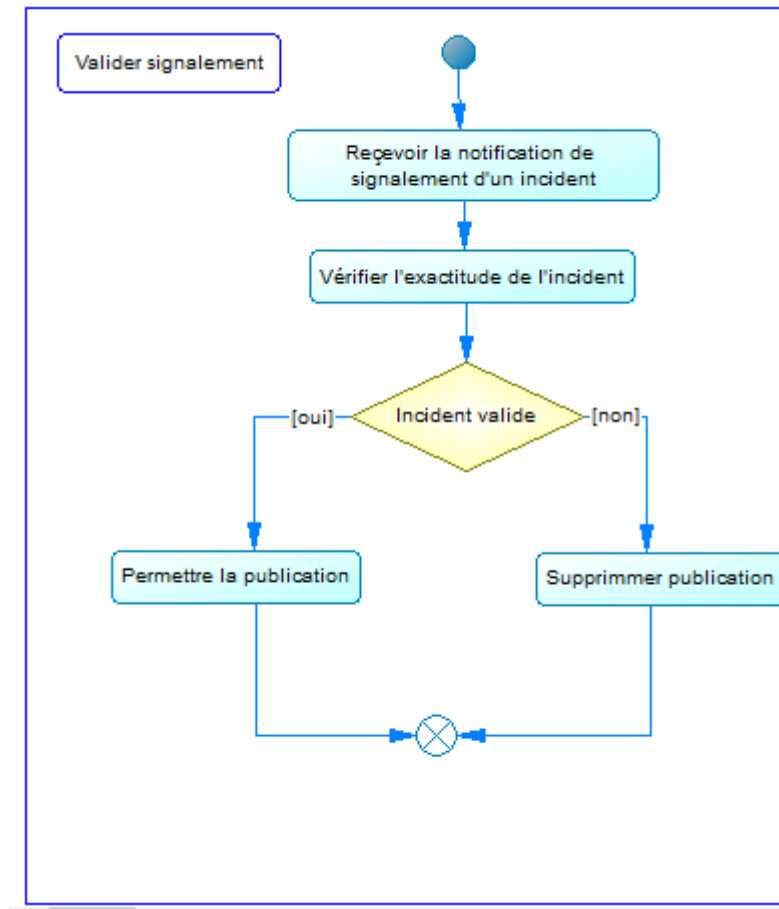


Fig. 6:Diagramme d'activité «valider un signalement»

d. Diagramme de séquence

A la suite de la spécification des besoins à l'aide des divers diagrammes correspondants, nous pouvons modéliser de manière adéquate et logique la structure de notre système. Le diagramme de séquence permettra donc de représenter cette structure en mettant en évidence les échanges entre les différents objets et acteurs de ce système en fonction du temps.

➤ Diagramme de séquence du cas d'utilisation «s'authentifier»

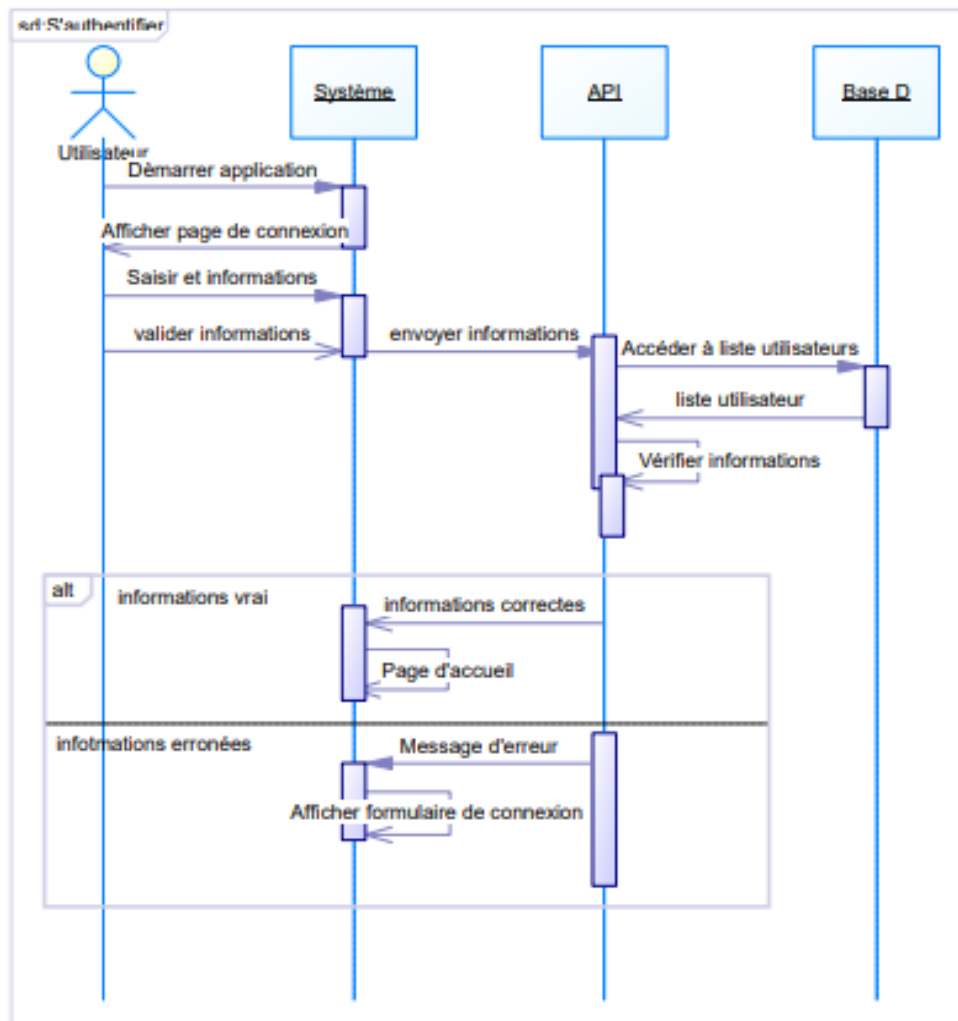


Fig. 7: Diagramme de séquence s'authentifier

➤ Diagramme de séquence du cas d'utilisation créer un compte

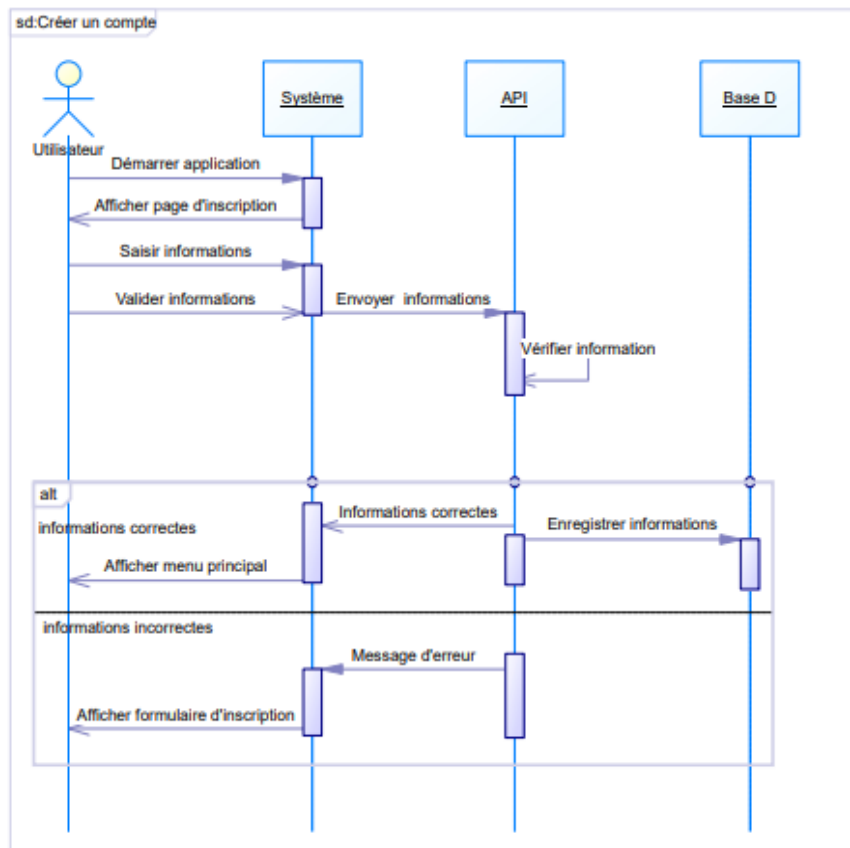


Fig. 8 : Diagramme du cas d'utilisation crée un compte

e. Diagramme de Classe

Le diagramme de classe est le diagramme le plus important de la modélisation orientée objet. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classe en montre la structure interne. Le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Les principaux éléments de cette vue statique sont les classes (description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes) et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation.

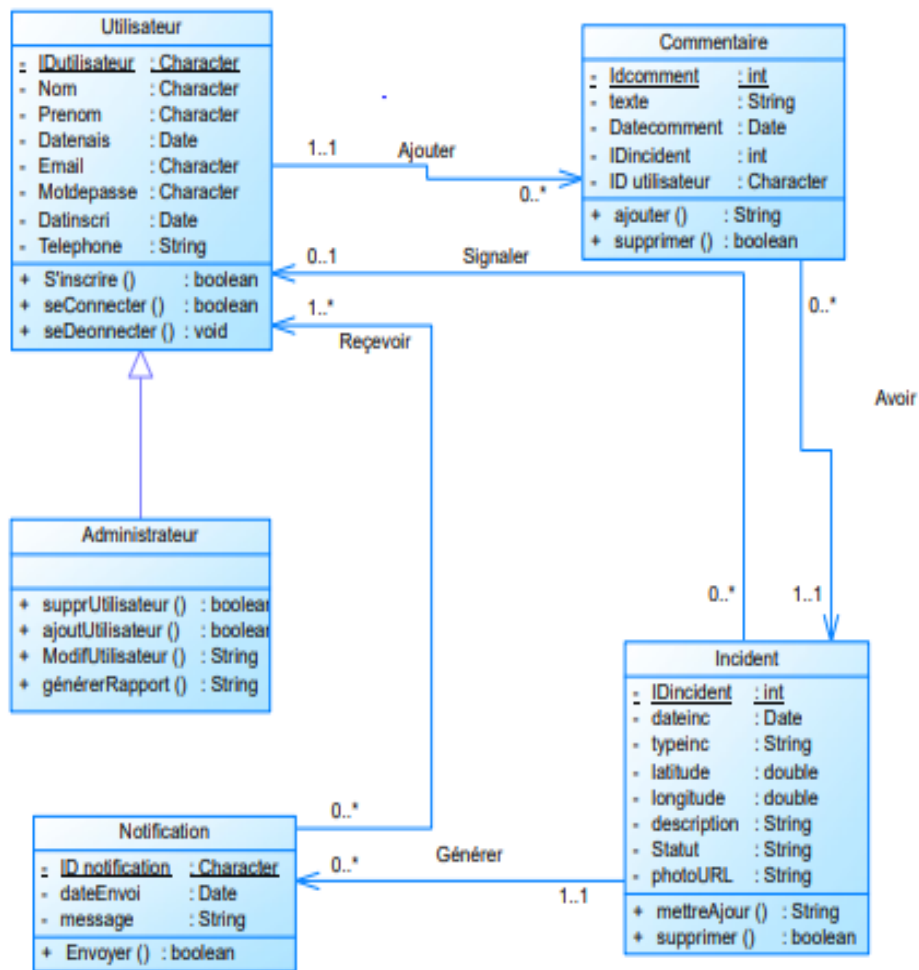


Fig. 9 Diagramme de classe

PARTIE 3 : REALISATION ET MISE EN ŒUVRE

Partie 3 : REALISATION ET MISE EN ŒUVRE

L'étape de réalisation de projet consiste à convertir en critères techniques les objectifs et les exigences du projet approuvé afin de permettre de concevoir dans les détails et de mettre en œuvre intégralement le produit final et la mise en œuvre vise à permettre de transporter la solution de conception approuvée dans la documentation pour acquérir et livrer le produit qui répondra aux objectifs et aux exigences du projet.

1.9 Matériels et logiciels utilisés

Désignation	Propriétés	Caractéristiques
Ordinateur	Marque et modèle	HP EliteBook Folio 9470m
	Processeur et fréquence	Core i5 1.90Ghz
	Type de système	64 bits
	Mémoire RAM	8 Go
	Disque dur	256 Go
	Système d'exploitation	Windows 10 Professionnel

Tableau 6 : Matériels utilisés

1.10 Logiciels utilisés

1.10.1 Git

Git est un logiciel de gestion de versions décentralisées. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. Le principal contributeur actuel de Git et depuis plus de 16 ans est Junio C Hamano. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes. Git ne repose pas sur un serveur centralisé, mais il utilise un système de connexion pair à pair. Le code informatique développé est stocké non seulement sur l'ordinateur de chaque contributeur du projet, mais il peut également l'être sur un serveur dédié. C'est un outil de bas niveau, qui se veut simple et performant, dont la principale tâche est de gérer l'évolution du contenu d'une arborescence.

Git dispose d'une multitude de commandes et parmi ces commandes nous pouvons énumérer :

- Git config : cette commande est utilisée pour configurer les préférences de l'utilisateur (mail, nom d'utilisateur, etc.).
- Git init : utilisée pour créer un nouveau dépôt Git.
- Git add : ajoute de nouveaux objets dans la base des objets pour chaque fichier modifié depuis le dernier commit. Les objets précédents restent inchangés.
- Git commit : elle permet de valider les modifications apportées.
- Git push : elle est utilisée pour publier des changements locaux et les charger vers un dépôt centralisé.



Fig. 10: Logo Git

1.10.2 GitHub

Le nom GitHub est composé du mot « git » faisant référence à un système de contrôle de version open-source et le mot « hub » faisant référence au réseau social bâti autour du système Git, mais aussi à une plateforme de correspondance qui est appelée en anglais un « hub ». GitHub est donc un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath, PJ Hyett et Tom Preston-Werner. Le site assure 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 nous permet l'hébergement du projet avec Git. Il nous permet d'effectuer une intégration continue et la gestion des versions du projet.



Fig. 11 :Logo GitHub

1.10.3 Flutter

Flutter est un kit de développement logiciel (SDK) d'interface utilisateur open-source développé par Google permettant de concevoir des applications multiplateformes pour Android, iOS ou encore Web. Il utilise un langage de programmation créé par Google : le Dart. C'est un langage de programmation orientée-objet comme Swift ou Kotlin. Les applications Flutter sont écrites en Dart et utilisent de nombreuses fonctionnalités avancées du langage.

Sur Windows, macOS et Linux, Flutter est exécuté par la machine virtuelle Dart et dispose d'un moteur d'exécution just-in-time. Lors de l'écriture et du débogage d'une application, Flutter utilise la compilation "Just In Time", qui permet de la recharge à chaud (hot reload), avec laquelle les modifications des fichiers sources peuvent être injectées dans une application en cours d'exécution. Flutter étend cette possibilité avec la prise en charge de la recharge à chaud stateful, où dans la plupart des cas les modifications du code source peuvent être reflétées

Partie 3 : Réalisation et mise en œuvre

immédiatement dans l'application en cours d'exécution sans nécessiter de redémarrage ou de perte d'état.

Flutter permet de développer 2 versions d'une application mobile (Android et iOS) en utilisant le même code source. Flutter est une bonne solution si vous souhaitez réduire les coûts de développement de votre application mobile. En effet, au lieu d'effectuer deux développements natifs en parallèle, développer une application mobile en Flutter permet de réduire le temps de développement.



Fig. 12 : Logo Flutter

1.10.4 Dart

Dart est un langage de programmation développé par Google, principalement utilisé pour le développement d'applications web et mobiles. Voici une vue d'ensemble de ses caractéristiques :

- **Langage Orienté Objet** : Dart est un langage orienté objet, ce qui signifie qu'il utilise des objets et des classes pour structurer le code. Cela facilite la création et la gestion de la logique complexe dans les applications.
- **Compilé et Interprété** : Dart peut être compilé en code natif pour des performances optimales, ce qui est particulièrement utile pour les applications mobiles et les logiciels de bureau. Il peut également être interprété, ce qui le rend adapté pour le développement web.
- **Utilisation Principale** : Dart est le langage principal utilisé pour développer des applications avec Flutter, un framework de développement d'applications mobiles et web également développé par Google. Flutter utilise Dart pour créer des interfaces utilisateur riches et performantes.
- **Syntaxe Moderne** : Dart a une syntaxe moderne et expressive qui ressemble à celle de nombreux autres langages de programmation populaires comme JavaScript, Java et C#. Cela facilite la prise en main pour les développeurs venant de ces langages.

Partie 3 : Réalisation et mise en œuvre

- Asynchrone et Concurrent : Dart prend en charge les opérations asynchrones grâce à des fonctionnalités comme les `Future` et les `Stream`, permettant une gestion efficace des tâches concurrentes et des appels réseau.
- Garbage Collection : Dart dispose d'un ramasse-miettes (garbage collector) automatique pour gérer la mémoire, ce qui aide à éviter les fuites de mémoire et optimise la gestion des ressources.
- Package Manager : Dart utilise un gestionnaire de packages appelé `pub`, qui permet de gérer les dépendances, de partager des bibliothèques et d'intégrer des packages tiers dans les projets Dart.
- Support pour le Web : En plus du développement mobile avec Flutter, Dart peut être utilisé pour le développement web grâce à Dart2js, un compilateur qui convertit le code Dart en JavaScript pour une exécution dans les navigateurs.
- Outils et IDE : Dart est bien pris en charge par des outils de développement modernes comme Visual Studio Code et IntelliJ IDEA, avec des extensions spécifiques pour faciliter le développement et le débogage.

En résumé, Dart est un langage polyvalent et moderne qui, grâce à ses intégrations avec des outils comme Flutter, est particulièrement adapté pour le développement d'applications mobiles et web performantes et bien structurées.



Fig. 13: Logo de Dart

1.10.5 Firebase

Firebase est une plateforme développée par Google qui propose une multitude de services backend, principalement destinés aux applications mobiles et web. Lorsqu'il est intégré dans un projet Flutter, Firebase permet de gérer plusieurs aspects essentiels d'une application sans nécessiter de créer un backend complexe. Firebase a permis de gérer plusieurs fonctionnalités de notre application comme :

- **Authentification (Firebase Authentication):** Firebase Authentication permet d'ajouter facilement des fonctionnalités d'authentification à votre application Flutter, que ce soit via email/mot de passe, téléphone, Google, Facebook, Twitter, ou d'autres fournisseurs OAuth. Firebase gère la sécurité, les sessions utilisateurs, et offre une interface unifiée pour différents types de connexion.
- **Base de Données en Temps Réel (Firebase Realtime Database) :** Firebase Realtime Database est une base de données NoSQL hébergée dans le cloud, qui permet de stocker et synchroniser des données en temps réel entre vos utilisateurs. Les données sont automatiquement mises à jour pour tous les utilisateurs en temps réel. C'est idéal pour les applications nécessitant des mises à jour en direct, comme les chats ou les applications collaboratives.
- **Stockage (Firebase Storage) :** Firebase Storage permet de stocker et de servir des fichiers générés par les utilisateurs, tels que des photos, des vidéos, et d'autres fichiers binaires. Les fichiers sont stockés de manière sécurisée et peuvent être facilement accessibles via des URL publiques ou privées. Firebase Storage s'intègre bien avec Firebase Authentication pour gérer les autorisations.
- **Notifications Push (Firebase Cloud Messaging - FCM) :** Firebase Cloud Messaging permet d'envoyer des notifications push à votre application Flutter, qu'elle soit en arrière-plan ou au premier plan. Vous pouvez facilement engager les utilisateurs avec des messages personnalisés, des alertes, ou des promotions, sans avoir à configurer un serveur de notification.



Fig. 14: Logo de Firebase

1.10.6 OpenStreetMap

OpenStreetMap (OSM) est une carte collaborative et gratuite du monde, créée par des bénévoles qui ajoutent et mettent à jour les données cartographiques. Voici quelques points clés à son sujet :

- **Collaboration Ouverte :** OSM est une plateforme ouverte où tout le monde peut contribuer. Les utilisateurs peuvent ajouter des informations telles que des routes, des bâtiments, des parcs, des points d'intérêt, et bien plus encore.

Partie 3 : Réalisation et mise en œuvre

- **Données Géographiques** : Les données sont collectées à partir de diverses sources, y compris des relevés GPS, des photos aériennes et des cartes anciennes. Elles sont ensuite vérifiées et améliorées par la communauté.
- **Utilisation Libre** : Les données d'OSM sont disponibles sous une licence ouverte (Open Database License - ODbL), ce qui permet à quiconque de les utiliser, modifier et redistribuer gratuitement, tant qu'ils attribuent correctement les contributions et partagent les dérivés sous la même licence.
- **Applications** : OSM est utilisé dans de nombreuses applications, des services de navigation aux outils d'analyse géospatiale, en passant par les cartes personnalisées sur les sites web et les applications mobiles.
- **Cartographie Locale** : Les détails peuvent être plus précis dans certaines régions en fonction de la densité de contributeurs, ce qui peut offrir des informations très détaillées dans certaines zones.

En gros, OSM est une ressource précieuse pour obtenir des cartes et des données géographiques à jour, grâce à la contribution de nombreux utilisateurs passionnés partout dans le monde.



Fig. 15: Logo de open street map

1.10.7 OpenWheaterMap

OpenWeatherMap est un service en ligne qui fournit des données météorologiques et climatiques en temps réel ainsi que des prévisions pour diverses localités dans le monde. Voici un aperçu des principales fonctionnalités d'OpenWeatherMap :

Partie 3 : Réalisation et mise en œuvre

- **Données Météorologiques** : OpenWeatherMap offre des informations sur les conditions météorologiques actuelles, y compris la température, l'humidité, la pression atmosphérique, la vitesse du vent et la couverture nuageuse.
- **Prévisions** : Le service propose des prévisions météorologiques à court terme (jusqu'à 7 jours) ainsi que des prévisions horaires, permettant aux utilisateurs de planifier en fonction des conditions météorologiques attendues.
- **Alertes Météorologiques** : Il fournit des alertes sur les conditions météorologiques extrêmes, telles que les tempêtes, les vagues de chaleur ou les gelées, ce qui peut aider les utilisateurs à se préparer à des événements climatiques inhabituels.
- **Cartes Météorologiques** : OpenWeatherMap propose des cartes interactives montrant les conditions météorologiques dans différentes régions, y compris des cartes de température, de précipitations et de vent.
- **API** : Les développeurs peuvent accéder aux données météorologiques via une API (Interface de Programmation d'Applications) pour intégrer les informations météorologiques dans leurs applications web ou mobiles. L'API permet de récupérer des données en temps réel, des prévisions et des historiques météorologiques.
- **Données Historique** : Le service propose également des données historiques sur les conditions météorologiques passées, ce qui peut être utile pour les analyses climatiques et les études de tendances.
- **Plan de Tarification** : OpenWeatherMap propose différents plans, y compris un plan gratuit avec des fonctionnalités de base et des plans payants offrant des fonctionnalités supplémentaires, des quotas plus élevés et un accès aux données plus détaillées.

En résumé, OpenWeatherMap est un outil puissant pour obtenir des informations météorologiques à jour et détaillées, idéal pour les développeurs, les chercheurs et les utilisateurs intéressés par les conditions climatiques locales et globales.



Fig. 16 : Logo de OpenWheaterMap

1.10.8 Visual Studio Code

Visual Studio Code est un éditeur extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent :

- La prise en charge du débogage.
- La mise en évidence de la syntaxe.
- La complétion intelligente du code.
- Les snippets,
- La refactorisation du code
- Git intégré.

Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. Le code source de Visual Studio Code provient du projet logiciel libre et open-source VS Code de Microsoft publié sous la licence MIT permissive, mais les binaires compilés sont des logiciels gratuits pour toute utilisation. Dans le StackOverflow 2019 Developer Survey, Visual Studio Code a été classé comme l'outil d'environnement de développement le plus populaire, avec 50,7% des 87.317 répondants déclarant l'utiliser.

Visual Studio Code est un éditeur de code source qui peut être utilisé avec une variété de langages de programmation, notamment Dart, HTML, CSS, JavaScript etc.



Fig. 17 : Logo VS Code

1.11 Sécurité de l'application

La sécurisation des applications est le processus qui consiste à développer, ajouter et tester des fonctionnalités de sécurité au sein des applications, afin d'éviter les vulnérabilités face à des menaces telles que les accès et les modifications non autorisés.

Cette partie nous permettra de présenter quelques mesures de sécurité prises par rapport à notre système :

Authentification : C'est un processus permettant au système de s'assurer de la légitimité de la demande d'accès faite par une entité (être humain ou un autre système...) afin d'autoriser l'accès de cette entité à des ressources du système conformément au paramétrage du contrôle d'accès. Dans notre application nous avons utilisé Firebase authentication qui permet de valider la légitimité de l'accès de l'entité.

A part l'authentification, Firebase nous permet d'écrire des règles concernant la lecture et l'écriture dans la base de données. Elles permettent de restreindre l'accès aux données en fonction de l'authentification de l'utilisateur, du rôle de l'utilisateur, ou d'autres critères définis.

```
1  {
2    "rules": {
3      ".read": "now < 1735689600000", // 2024-12-31
4      ".write": "now < 1735689600000", // 2024-12-31
5      "users": {
6        "$userId": {
7          ".read": "$userId === auth.uid",
8          ".write": "$userId === auth.uid",
9        }
10     }
11   }
12 }
13
```

Fig. 18 : Exemple de règles

1.12 Evaluation financière de la solution

1.12.1 Coût Matériel

L'évaluation financière consiste à valoriser les flux résultant des études précédentes pour déterminer le financement du projet.

Ainsi, dans cette partie de notre document, nous allons expliciter les divers coûts liés à la mise en place de la solution retenue. Nous allons débiter par la présentation du coût matériel, ensuite poursuivre par celui du coût d'hébergement, puis passer au coût d'exploitation et au coût de conception, et enfin un récapitulatif de tous les coûts.

Partie 3 : Réalisation et mise en œuvre

Désignation	Description	Prix unitaire (FCFA)	Quantité	Coût (FCFA)
Ordinateur	Machine pour la conception, la mise en œuvre et pour l'administration de la plateforme : HP EliteBook core i5	230000	01	230000
Total				230000

Tableau 7 : Coût matériel

1.12.2 Coût d'exploitation

Désignation	Description	Prix unitaire (FCFA)	Quantité	Coût (FCFA)
Forfait internet	Fibre optique Bande passante : 50 Mb/s	30000	12	360000
Total				360000

Tableau 8 : Coût d'exploitation

1.12.3 Coût de conception

Désignation	Description	Prix unitaire (FCFA)	Quantité	Coût (FCFA)
Main d'œuvre	Montant à percevoir en tant que concepteur et réalisateur	15000/jour	75 jours	1125000
Total				1125000

Tableau 9 : Coût de conception

1.12.4 Coût total

DESIGNATION	COUT (FCFA)
Coût Matériel	230000
Coût d'exploitation	360000
Coût de conception	1125000
TOTAL	1815000

Tableau 10 : Coût total

1.13 Présentation de l'application

1.13.1 Mise en place de la base de données

Dans notre application nous avons utilisé Firebase Realtime Database qui est une base de données No SQL dans le cloud qui stocke les données sous forme de JSON et les synchronise en temps réel avec les clients connectés. Dans Firebase Realtime Database, un champ correspond à une paire clé-valeur dans une structure JSON. Chaque champ représente un nœud dans l'arbre de la base de données.

Les champs sont créés en ajoutant des paires clé-valeur dans cet arbre. Une clé est le nom du champ, et la valeur peut être un simple type de données (comme une chaîne, un nombre, un booléen) ou un objet JSON contenant d'autres paires clé-valeur (un sous-arbre).

- Création du champ users

```
final User? user = credential.user;
if (user != null) {
  // Enregistrer les données de l'utilisateur dans Firebase Realtime Database
  await _database.ref().child('users').child(user.uid).set({
    'email': email,
    'username': username, // Enregistrer le nom d'utilisateur
    'createdAt': DateTime.now().toIso8601String(),
    'phone': phone,
  });
}
```

Fig. 19 : Code pour création de champ pour les utilisateur à la création de compte

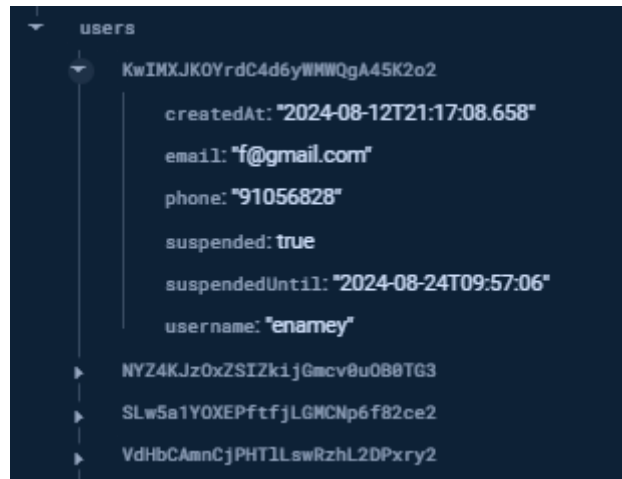


Fig. 20 : Exemple de champ d'un utilisateur

➤ Création du champ incident

```
await FirebaseDatabase.instance.ref().child('incidents').push().set({
  'userId': userId,
  'comment': _commentController.text,
  'imageUrl': imageUrl,
  'incidentType': _selectedIncident,
  'location': {
    'latitude': widget.draggableMarkerPosition.latitude,
    'longitude': widget.draggableMarkerPosition.longitude,
  },
  'timestamp': DateTime.now().toIso8601String(), // Enregistre la date et l'heure actuel
});
```

Fig. 21 : Code pour création de champ pour les incidents au signalement d'un incident



Fig. 22 : Exemple de champ pour les incidents

1.13.2 Plan de navigation

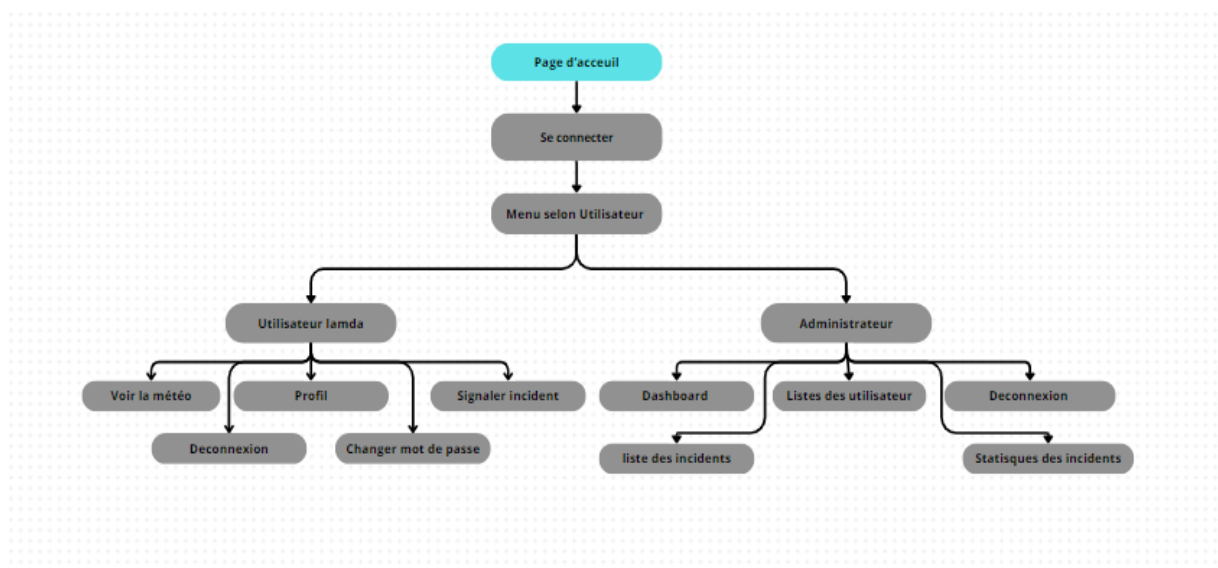


Fig. 23 : Plan de navigation

1.13.3 Quelques Masques de saisies

1.13.3.1 Page de création de compte de l'utilisateur



Fig. 24 : Page de création de compte

Partie 3 : Réalisation et mise en œuvre

1.13.3.2 Page de connexion de l'utilisateur

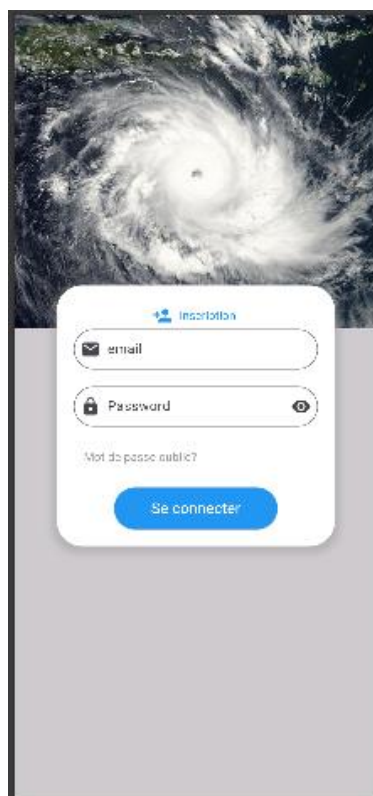


Fig. 25 : Page de connexion

1.13.3.3 Dashboard de l'administrateur

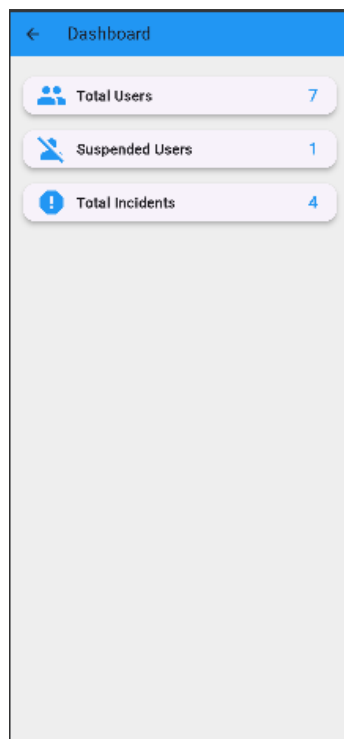


Fig. 26 : Dashboard

Partie 3 : Réalisation et mise en œuvre

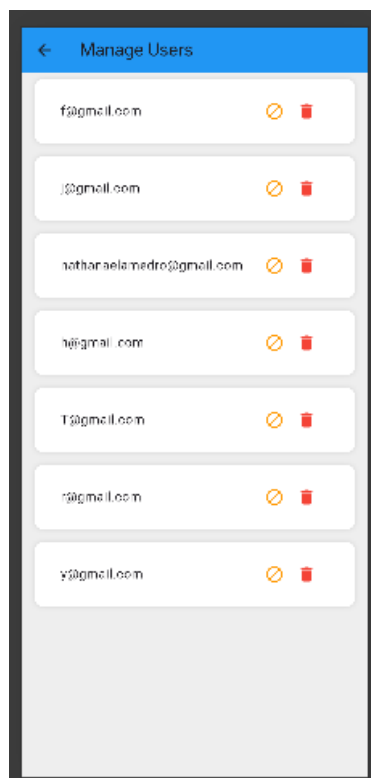


Fig. 27 : Liste des utilisateurs

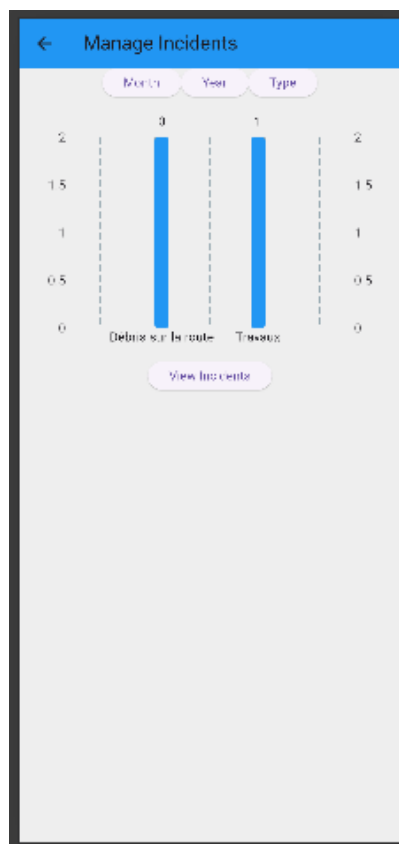


Fig. 28 : Graphes du nombre d'incidents par type

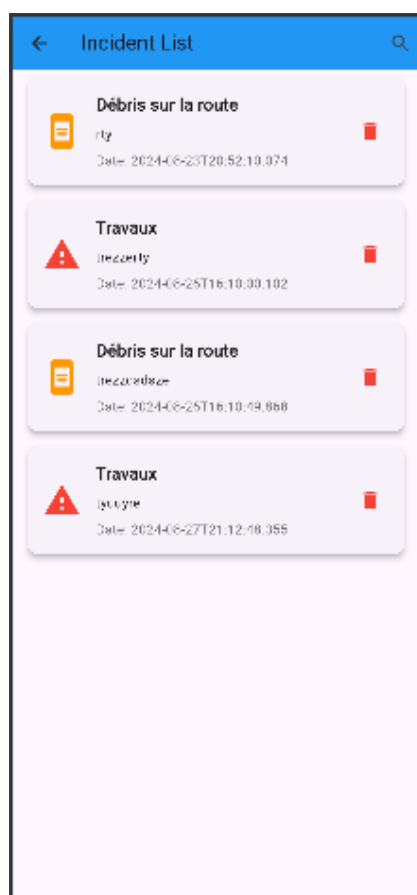


Fig. 29 : Liste des incidents

CONCLUSION

Notre projet avait pour objectif de développer une application de gestion des incidents, axée sur la localisation en temps réel et la communication instantanée entre les utilisateurs. Ce projet a été une excellente opportunité de mettre en pratique les connaissances acquises, tout en développant de nouvelles compétences techniques en utilisant des outils comme Flutter, Firebase et OpenStreetMap.

Dans la première phase, nous avons réalisé une étude descriptive de notre projet, en analysant les besoins spécifiques liés à la gestion des incidents et en définissant les solutions adaptées à ces problématiques. Ensuite, lors de la phase d'analyse et de conception, nous avons appliqué des méthodes rigoureuses, telles que la modélisation UML, pour structurer et planifier les différentes fonctionnalités de l'application.

La troisième phase a consisté à décrire l'environnement technique et à sélectionner les outils de développement appropriés. Nous avons configuré l'intégration avec Firebase pour la gestion des données en temps réel, et l'affichage des itinéraires en utilisant le SDK OpenRouteService. Nous avons également présenté la structure de la base de données ainsi que les interfaces utilisateur, tout en prenant soin d'optimiser l'expérience utilisateur avec des éléments visuels et des fonctionnalités pratiques.

Cependant, comme toute première version, cette application ne peut prétendre répondre à tous les besoins des utilisateurs. Ainsi, à mesure que de nouveaux besoins émergeront, des fonctionnalités supplémentaires pourront être intégrées pour enrichir et perfectionner l'application. Ce projet représente une base solide sur laquelle nous pourrions continuer à innover et à apporter des améliorations, tout en garantissant la sécurité et l'efficacité de la gestion des incidents pour les utilisateurs.

BIBLIOGRAPHIE INDICATIVE

Année académique : 2021-2022

- Cours d'UML : M. SEWAVI Kokou Maurice ;
- Rédaction scientifique : M. WOAMEY Komla Mike ;

WEBOGRAPHIE INDICATIVE

Année académique : 2023-2024

- <https://www.google.com/>, tout le long du projet ;
- <https://stackoverflow.com/>, tout le long du projet ;
- <https://www.youtube.com/>, tout le long du projet ;

DOCUMENTS ANNEXES

- NABATOU Roc, **PLATEFORME DE PRESENTATION DES PROGRAMMES ET DE PUBLICATION DES EVENEMENTS : CAS DE L'ASSOCIATION TIDD**
- TCHAKOROM Nawal, **CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION MOBILE DE RESERVATION DE SERVICES DE SALONS D'ESTHETIQUE**
- AWESSO Lidaou Espoir, **MISE EN PLACE D'UNE APPLICATION WEB DE GESTION DE STOCK**

TABLES DES MATIERES

REMERCIEMENTS	i
SOMMAIRE	ii
LISTES DES FIGURES.....	iii
LISTES DES TABLEAUX.....	iv
INTRODUCTION	2
PARTIE1 : CAHIER DES CHARGES	3
1.1 Présentation du sujet	3
1.2 Problématique du sujet.....	3
1.3 Intérêt de sujet	4
1.3.1 Objectifs	4
1.3.2 Résultats	4
PARTIE 2 : ANALYSE ET CONCEPTION.....	6
1.5 Étude de l'Existant	6
1.6 Critique de l'Existant	6
1.7 Planning prévisionnel et de réalisation	6
1.8 Etude détaillée de la solution.....	8
1.8.1 Etude préliminaire de la solution	8
1.8.1.1 Choix des outils d'analyse	8
1.8.1.2 Présentation de L'outils de modélisation	9
1.8.1.3 Acteurs et rôles	9
1.8.1.4 Captures des besoins fonctionnels	10
Partie 3 : REALISATION ET MISE EN ŒUVRE	19
1.9 Matériels et logiciels utilisés	19
1.10 Logiciels utilisés	19
1.10.1 Git	19
1.10.2 GitHub	20
1.10.3 Flutter.....	20
1.10.4 Dart.....	21
1.10.5 Firebase	22
1.10.6 OpenStreetMap.....	23
1.10.7 OpenWheaterMap	24
1.10.8 Visual Studio Code.....	26

1.11	Sécurité de l'application	27
1.12	Evaluation financière de la solution	27
1.12.1	Coût Matériel	27
1.12.2	Coût d'exploitation.....	28
1.12.3	Coût de conception	28
1.12.4	Coût total.....	29
1.13	Présentation de l'application	29
1.13.1	Mise en place de la base de données	29
1.13.2	Plan de navigation	31
1.13.3	Quelques Masques de saisies.....	31
1.13.3.1	Page de création de compte de l'utilisateur.....	31
1.13.3.2	Page de connexion de l'utilisateur.....	32
1.13.3.3	Dashboard de l'administrateur	32
	CONCLUSION	35
	BIBLIOGRAPHIE INDICATIVE	I
	WEBOGRAPHIE INDICATIVE	II
	DOCUMENTS ANNEXES	III
	TABLES DES MATIERES	IV