

# PROJET DE FIN D'ANNEE

*Effectué à*



*Préparé par*

**HABAIB et Ahmed**

**DÉPARTEMENT INFORMATIQUE**

**SUJET : Application Mobile Flutter et Déploiement sur CLOUD**

*Effectué pendant la période allant du 01/02/2024 à 31/05/2024*

*Soutenu le 11/06/2024, devant la commission d'examen :*

*Mme. Wafa Naifar*

*M. Mohamed Koubaa*

*M. Wael Sellami : Encadrant*

*Année Universitaire 2023 / 2024*



---

# REMERCIEMENTS

Ce travail de mémoire a été réalisé en collaboration avec l’Institut International de Technologie.

Nous tenons à remercier profondément Dieu qui nous a donné la force et la patience d’accomplir ce modeste travail.

Nous souhaitons exprimer également notre profonde gratitude et nos sincères remerciements à notre encadrant académique, Monsieur Wael Sellami. Nous le remercions chaleureusement pour la confiance qu’il a investie en acceptant de nous encadrer pour la réalisation de ce projet, pour le temps qu’il a consacré, son soutien continu et ses conseils précieux tout au long du projet.

Nous tenons également à remercier tous nos enseignants à l’Institut International de Technologie qui ont contribué à notre formation académique et qui n’ont jamais épargné le moindre effort pour nous guider et nous soutenir.

Enfin, nos remerciements vont à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.



---

# TABLE DES MATIÈRES

<b>LISTE DES FIGURES</b>	<b>vi</b>
<b>INTRODUCTION GÉNÉRALE</b>	<b>1</b>
<b>1 ÉTUDE PRÉALABLE</b>	<b>2</b>
1.1 INTRODUCTION . . . . .	3
1.2 ETUDE ET CRITIQUE DE L'EXISTANT . . . . .	3
1.3 SOLUTION PROPOSÉE . . . . .	4
1.4 OBJECTIFS À ATTEINDRE . . . . .	5
1.5 CONCLUSION . . . . .	6
<b>2 MÉTHODOLOGIE &amp; CONCEPTION</b>	<b>7</b>
2.1 INTRODUCTION . . . . .	8
2.2 MÉTHODOLOGIE . . . . .	8
2.2.1 ÉTUDE DES MÉTHODOLOGIES DE TRAVAIL . . . . .	8
2.2.2 CHOIX DE LA MÉTHODOLOGIE SCRUM . . . . .	9
2.3 LANGAGE DE CONCEPTION . . . . .	10
2.3.1 PRÉSENTATION D'UML . . . . .	10
2.3.2 OBJECTIFS DE L'UML . . . . .	11
2.4 CONCEPTION . . . . .	11
2.4.1 IDENTIFICATION DES ACTEURS . . . . .	11
2.4.2 DIAGRAMME DE CAS D'UTILISATION . . . . .	13
2.4.3 DESCRIPTION TEXTUELLE DES CAS D'UTILISATION . . . . .	13
2.4.4 DIAGRAMME DE SÉQUENCE . . . . .	16
2.4.5 DIAGRAMME DE CLASSE . . . . .	17
2.5 CONCLUSION . . . . .	18
<b>3 RÉALISATION</b>	<b>19</b>
3.1 INTRODUCTION . . . . .	20

3.2	ENVIRONNEMENT DE TRAVAIL . . . . .	20
3.2.1	ENVIRONNEMENT MATÉRIEL . . . . .	20
3.2.2	ENVIRONNEMENT LOGICIEL . . . . .	21
3.3	LANGAGE DE PROGRAMMATION ET FRAMEWORK UTILISÉS . . . . .	22
3.3.1	LANGAGE DE PROGRAMMATION . . . . .	22
3.3.2	FRAMEWORK . . . . .	22
3.4	ÉTUDE COMPARATIVE DES TECHNOLOGIES UTILISÉES . . . . .	23
3.4.1	CHOIX DE FLUTTER . . . . .	23
3.4.2	CHOIX DU FIREBASE . . . . .	23
3.4.3	POURQUOI FIRESTORE ET NON REALTIME DATABASE? . . . . .	24
3.5	INTERFACE GRAPHIQUE ET EXPÉRIENCE UTILISATEUR . . . . .	25
3.6	ARCHITECTURE . . . . .	31
3.6.1	ARCHITECTURE EN 3 TIERS . . . . .	32
3.6.2	CLIENT MOBILE (FRONTEND) . . . . .	33
3.6.3	FIREBASE . . . . .	33
3.7	DÉPLOIEMENT . . . . .	33
3.7.1	GIT ET GITHUB . . . . .	34
3.7.2	LES SOLUTIONS CLOUD . . . . .	36
3.7.3	CHOIX DE AWS . . . . .	36
3.7.4	PRÉPARATION POUR LE DÉPLOIEMENT . . . . .	37
3.7.5	DÉPLOIEMENT SUR AWS EC2 . . . . .	37
3.8	CONCLUSION . . . . .	43
	<b>CONCLUSION GÉNÉRALE</b>	<b>44</b>

# LISTE DES FIGURES

1.1	<i>Logo de notre Application</i>	5
2.1	<i>Logo UML</i>	10
2.2	<i>Diagramme du cas d'utilisation</i>	13
2.3	<i>Diagramme de séquence login</i>	17
2.4	<i>Diagramme de séquence Soumettre reclamation</i>	17
2.5	<i>Diagramme de classe</i>	18
3.1	<i>Logo Android Studio IDE</i>	21
3.2	<i>Logo Firebase</i>	21
3.3	<i>Logo Overleaf</i>	21
3.4	<i>Logo Dart</i>	22
3.5	<i>Logo Flutter</i>	22
3.6	<i>Page d'accueil</i>	25
3.7	<i>Espace Client</i>	26
3.8	<i>Espace reclamation</i>	26
3.9	<i>Message de Confirmation</i>	27
3.10	<i>Téléchargement de PDF</i>	28
3.11	<i>Remplissage des IDs</i>	28
3.12	<i>Espace reclamation pour la suivi</i>	29
3.13	<i>Identification d'administrateur</i>	30
3.14	<i>Liste des réclamations</i>	30
3.15	<i>Réponse d'admin</i>	31
3.16	<i>Architecture en 3 Tiers de l'Application "Al-Baladiya"</i>	32
3.17	<i>Logo de Git</i>	34
3.18	<i>Logo de GitHub</i>	34
3.19	<i>Interface de GitHub Desktop</i>	35
3.20	<i>Logo AWS</i>	36
3.21	<i>Groupe de sécurité</i>	38

3.22 <i>Téléchargement d'SDK Flutter</i> . . . . .	40
3.23 <i>Installation NGNIX</i> . . . . .	42
3.24 <i>Fichier de configuration du serveur</i> . . . . .	43



---

## Liste des tableaux

3.1	Comparaison entre Firebase Realtime Database et Firestore . . . . .	24
3.2	Comparaison entre Amazon AWS, Google Cloud Platform et Microsoft Azure .	36



---

# INTRODUCTION GÉNÉRALE

Les applications mobiles ont radicalement changé notre façon de communiquer et d'interagir avec le monde qui nous entoure. Grâce à elles, nous pouvons accéder à une multitude de services et d'informations directement depuis nos smartphones, rendant notre vie quotidienne plus pratique et plus connectée.

Le projet "Al-Baladiya" vise à créer une application mobile qui permet aux citoyens de communiquer plus efficacement avec leurs administrations municipales. En utilisant cette application, les citoyens peuvent facilement soumettre des réclamations sur des problèmes locaux, joindre des photos pour illustrer leurs préoccupations, et suivre l'état de leurs réclamations en temps réel.

Pour réaliser ce projet, nous avons suivi une approche méthodologique rigoureuse et utilisé des technologies modernes. Ce rapport se compose des sections suivantes :

**Étude Préable :** Nous définissons les objectifs du projet, analysons les solutions existantes et leurs lacunes, et décrivons les étapes de développement de l'application.

**Méthodologie et Conception :** Nous discutons des différentes méthodologies de travail et expliquons notre choix de la méthodologie Scrum pour ce projet. Nous présentons également la conception technique de l'application, y compris les diagrammes de cas d'utilisation et les séquences d'interaction.

**Réalisation :** Nous décrivons les environnements matériel et logiciel utilisés, les langages de programmation et les frameworks, ainsi que le processus de déploiement sur AWS EC2.

**Conclusion :** Nous récapitulons les résultats obtenus et proposons des perspectives pour l'amélioration future de l'application.



---

# ÉTUDE PRÉALABLE

## Sommaire

---

1.1	INTRODUCTION . . . . .	3
1.2	ETUDE ET CRITIQUE DE L'EXISTANT . . . . .	3
1.3	SOLUTION PROPOSÉE . . . . .	4
1.4	OBJECTIFS À ATTEINDRE . . . . .	5
1.5	CONCLUSION . . . . .	6

---

### 1.1 INTRODUCTION

Ce premier chapitre, "Étude Préalable", définit les objectifs du projet, examine les solutions actuelles et leurs lacunes, et décrit les étapes de développement de l'application. Cette analyse est essentielle pour assurer une base solide et une compréhension claire des besoins pour les prochaines phases du projet.

### 1.2 ETUDE ET CRITIQUE DE L'EXISTANT

L'étude de l'existant est une étape cruciale pour comprendre le contexte actuel dans lequel l'application "Al-Baladiya" sera déployée. Cette analyse permet d'identifier les solutions actuelles, leurs forces et leurs faiblesses, et de déterminer comment notre application peut offrir une valeur ajoutée. Dans le contexte tunisien, les réclamations des citoyens aux administrations municipales sont principalement effectuées par des canaux traditionnels comme la radio, la télévision, Facebook, ou encore en personne. Cette section explore ces méthodes et évalue leur efficacité et leurs limitations.

Comme :

#### ✓ DES CANAUX TRADITIONNELS DE RÉCLAMATION

En Tunisie, la radio et la télévision sont des médias populaires et largement utilisés par les citoyens pour soumettre leurs réclamations. Les émissions interactives permettent aux auditeurs et téléspectateurs de signaler des problèmes directement aux autorités publiques ou d'attirer l'attention sur des préoccupations locales. Cependant, cette méthode présente plusieurs défaillances :

- **Efficacité Limitée** : Les réclamations faites par la radio et la télévision dépendent souvent de l'intervention des présentateurs pour être relayées aux autorités compétentes, ce qui peut entraîner des délais et une réponse non systématique.
- **Suivi Difficile** : Il est difficile pour les citoyens de suivre l'évolution de leurs réclamations, car il n'existe pas de système structuré pour assurer un retour d'information.

- **Portée Limitée** : Seuls ceux qui écoutent ou regardent l'émission au moment précis de la réclamation peuvent être au courant du problème, limitant ainsi l'impact de la communication.

### ✓ RÉCLAMATIONS EN PERSONNE

Se déplacer en personne aux bureaux municipaux reste une méthode courante pour soumettre des réclamations en Tunisie. Bien que cette méthode assure une interaction directe avec les représentants municipaux, elle comporte des désavantages notables :

- **Temps et Effort** : Se déplacer physiquement nécessite du temps et des ressources, ce qui peut être contraignant pour les citoyens.
- **Bureaucratie** : Le processus peut être complexe et bureaucratique, avec des délais potentiels dans le traitement des réclamations.
- **Accessibilité** : Les citoyens vivant dans des zones rurales ou éloignées peuvent trouver difficile de se rendre aux bureaux municipaux.

### ✓ L'ABSENCE DE SOLUTIONS TECHNOLOGIQUES EN TUNISIE

Contrairement à d'autres pays où des solutions technologiques avancées existent pour faciliter la communication entre les citoyens et les administrations municipales, la Tunisie ne dispose pas actuellement d'applications mobiles dédiées à cette fonction. Cette absence de solutions modernes laisse un vide que l'application "Al-Baladiya" vise à combler.

## 1.3 SOLUTION PROPOSÉE

L'application mobile "Al-Baladiya" est conçue pour améliorer la communication entre les citoyens et les administrations municipales en Tunisie. Elle permet aux citoyens de soumettre des réclamations en ligne et en temps réel, de joindre des images à leurs réclamations, et de suivre l'état de celles-ci via un système de chat interactif. L'application offre une interface

conviviale et intuitive, développée en utilisant le framework Flutter, et exploite Firebase pour la gestion des données et la synchronisation en temps réel.



**FIGURE 1.1** – *Logo de notre Application*

### ✓ FONCTIONNALITÉS PRINCIPALES

- **Soumission de Réclamations** : Les citoyens peuvent signaler des problèmes municipaux en remplissant un formulaire détaillé, comprenant un titre, une description, une localisation, et des images.
- **Suivi des Réclamations** : Chaque réclamation reçoit un identifiant unique permettant aux citoyens de suivre l'évolution de leur réclamation via une salle de chat dédiée.
- **Communication en Temps Réel** : Les citoyens et les administrateurs municipaux peuvent échanger des messages en temps réel pour discuter des réclamations soumises.

## 1.4 OBJECTIFS À ATTEINDRE

### **Améliorer l'Accessibilité et la Commodité :**

- Permettre aux citoyens de soumettre des réclamations à tout moment et de n'importe où, sans avoir besoin de se déplacer physiquement aux bureaux municipaux.
- Offrir une plateforme centralisée pour la gestion des réclamations, réduisant ainsi la dépendance aux canaux traditionnels comme la radio, la télévision, ou les réseaux sociaux.

### **Accélérer le Traitement des Réclamations :**

- Faciliter une communication directe et rapide entre les citoyens et les administrations municipales, permettant un traitement plus efficace et plus rapide des réclamations.
- Réduire les délais de réponse grâce à la notification instantanée et à la gestion en temps réel des réclamations.

### **Augmenter la Transparence et la Responsabilité :**

- Fournir un suivi clair et transparent de l'état des réclamations, permettant aux citoyens de voir les actions entreprises par les administrations municipales.
- Accroître la responsabilité des administrations en assurant que les réclamations sont visibles et doivent être traitées dans un délai raisonnable.

### **Utilisation de Technologies Modernes :**

- Exploiter le potentiel de Flutter pour créer une application mobile performante et multiplateforme (iOS et Android).
- Utiliser Firebase pour une gestion de données robuste et une synchronisation en temps réel, garantissant une expérience utilisateur fluide et réactive.

En résumé, l'application "Al-Baladiya" vise à moderniser et à optimiser la gestion des réclamations municipales en Tunisie, en utilisant des technologies innovantes pour offrir une solution pratique, rapide et transparente aux citoyens et aux administrations.

## **1.5 CONCLUSION**

L'étude préalable du projet "Al-Baladiya" a clarifié les besoins pour développer l'application, identifié les lacunes des méthodes traditionnelles en Tunisie, et défini les étapes méthodologiques nécessaires. Cette approche garantit un développement efficace, répondant aux attentes des citoyens et améliorant la communication avec les administrations municipales. Le prochain chapitre se concentrera sur la conception détaillée de l'application.

# MÉTHODOLOGIE CONCEPTION

## Sommaire

<b>2.1</b>	<b>INTRODUCTION . . . . .</b>	<b>8</b>
<b>2.2</b>	<b>MÉTHODOLOGIE . . . . .</b>	<b>8</b>
2.2.1	ÉTUDE DES MÉTHODOLOGIES DE TRAVAIL . . . . .	8
2.2.2	CHOIX DE LA MÉTHODOLOGIE SCRUM . . . . .	9
<b>2.3</b>	<b>LANGAGE DE CONCEPTION . . . . .</b>	<b>10</b>
2.3.1	PRÉSENTATION D'UML . . . . .	10
2.3.2	OBJECTIFS DE L'UML . . . . .	11
<b>2.4</b>	<b>CONCEPTION . . . . .</b>	<b>11</b>
2.4.1	IDENTIFICATION DES ACTEURS . . . . .	11
2.4.2	DIAGRAMME DE CAS D'UTILISATION . . . . .	13
2.4.3	DESCRIPTION TEXTUELLE DES CAS D'UTILISATION . .	13
2.4.4	DIAGRAMME DE SÉQUENCE . . . . .	16
2.4.5	DIAGRAMME DE CLASSE . . . . .	17
<b>2.5</b>	<b>CONCLUSION . . . . .</b>	<b>18</b>

## 2.1 INTRODUCTION

Le deuxième chapitre de notre projet "Al-Baladiya" se concentrera sur la conception détaillée de l'application. Après avoir établi une base solide grâce à l'étude préalable, nous allons maintenant élaborer l'architecture technique, définir les choix technologiques et concevoir l'interface utilisateur de manière précise. Ce chapitre est essentiel pour assurer que l'application sera fonctionnelle, efficace et user-friendly.

## 2.2 MÉTHODOLOGIE

La méthodologie de travail est essentielle pour structurer le développement de l'application "Al-Baladiya". Dans cette section, nous allons explorer différentes méthodologies de travail couramment utilisées dans le développement de logiciels, puis nous expliquerons pourquoi nous avons choisi la méthodologie Scrum pour ce projet.

### 2.2.1 ÉTUDE DES MÉTHODOLOGIES DE TRAVAIL

Il existe plusieurs méthodologies de gestion de projet et de développement de logiciels, chacune ayant ses propres avantages et inconvénients. Les principales méthodologies incluent :

- **Modèle en Cascade** : Une méthodologie linéaire où chaque phase du projet doit être complétée avant de passer à la suivante. Cette approche est simple et facile à gérer, mais manque de flexibilité pour les changements en cours de projet.
- **Méthodologie Agile** : Un ensemble de méthodologies basées sur un développement itératif et incrémental. Agile favorise une collaboration étroite avec le client et des cycles de développement courts appelés sprints.
- **Kanban** : Une méthode visuelle de gestion des flux de travail qui utilise un tableau Kanban pour représenter les tâches à accomplir, en cours et terminées. Kanban est flexible et favorise l'amélioration continue.
- **Scrum** : Un cadre Agile qui organise le développement en sprints de durée fixe, généralement de deux à quatre semaines. Scrum met l'accent sur des réunions régulières

(comme les stand-ups quotidiens) et des revues de sprint pour évaluer les progrès et ajuster les priorités.

### 2.2.2 CHOIX DE LA MÉTHODOLOGIE SCRUM

Pour le développement de l'application "Al-Baladiya", nous avons choisi d'utiliser la méthodologie Scrum. Scrum est particulièrement adapté aux projets où les exigences peuvent évoluer et où une collaboration étroite avec le client est essentielle. Voici comment nous avons adapté Scrum à notre projet :

Bien que Scrum soit généralement utilisé par des équipes, il peut également être adapté pour un développement individuel en suivant les principes de base :

- **Sprints** : Le projet est divisé en sprints de deux semaines. À la fin de chaque sprint, une révision est effectuée pour évaluer les progrès et planifier le sprint suivant.
- **Backlog de Produit** : Toutes les fonctionnalités à développer sont listées dans un backlog de produit. Les éléments du backlog sont priorisés en fonction de leur importance et de leur valeur pour le projet.
- **Stand-ups Quotidiens** : Bien que les stand-ups quotidiens soient traditionnels dans une équipe, pour un développement individuel, il est utile de prendre un moment chaque jour pour évaluer les progrès et ajuster les tâches.
- **Revue de Sprint** : À la fin de chaque sprint, une revue est menée pour démontrer les fonctionnalités complétées et obtenir des retours. Même en solo, cela peut impliquer la revue par des mentors ou des parties prenantes externes.
- **Rétrospectives** : Après chaque sprint, une rétrospective est réalisée pour identifier ce qui a bien fonctionné, ce qui n'a pas fonctionné, et comment les processus peuvent être améliorés pour le prochain sprint.

#### Application de Scrum au Projet "Al-Baladiya"

En appliquant Scrum à notre projet, nous avons suivi les étapes suivantes :



- **Initialisation du Projet** : Définition du backlog de produit avec toutes les fonctionnalités souhaitées pour l'application "Al-Baladiya".
- **Planification des Sprints** : Décomposition du backlog en sprints de deux semaines, avec des objectifs clairs pour chaque sprint.
- **Développement Itératif** : Mise en œuvre des fonctionnalités en sprints, avec des revues régulières pour évaluer les progrès.
- **Revues et Rétrospectives** : À la fin de chaque sprint, une revue et une rétrospective sont effectuées pour assurer l'amélioration continue du processus de développement.

En utilisant la méthodologie Scrum, nous avons pu maintenir une approche structurée et flexible pour le développement de l'application "Al-Baladiya", assurant ainsi une livraison progressive et l'intégration des retours en temps réel.

## 2.3 LANGAGE DE CONCEPTION

### 2.3.1 PRÉSENTATION D'UML

UML (Unified Modeling Language) est un langage de modélisation visuelle utilisé pour spécifier, visualiser, construire et documenter les artefacts d'un système logiciel. Développé par l'Object Management Group (OMG), UML est un standard international largement accepté pour la modélisation des systèmes orientés objets.

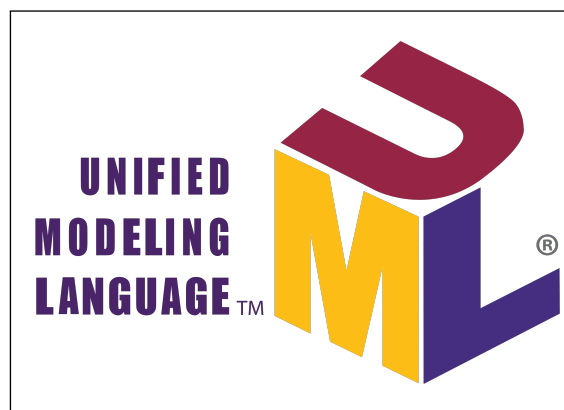


FIGURE 2.1 – Logo UML

### 2.3.2 OBJECTIFS DE L'UML

**Visualisation :** UML permet de représenter graphiquement les différentes composantes d'un système logiciel, facilitant ainsi la compréhension de la structure et du comportement du système.

**Spécification :** Il aide à définir clairement les exigences et les fonctionnalités du système, permettant aux développeurs et aux parties prenantes de partager une compréhension commune.

**Construction :** UML fournit des guides et des outils pour la construction effective du système logiciel.

**Documentation :** Il permet de créer des documents détaillés qui décrivent les aspects du système, utiles pour la maintenance et l'évolution du système.

## 2.4 CONCEPTION

L'architecture système de l'application "Al-Baladiya" est conçue pour être robuste, scalable et sécurisée. Elle comprend plusieurs composants clés qui interagissent pour offrir une expérience utilisateur fluide et efficace. Cette section détaille les différents aspects de l'architecture, en incluant des diagrammes et des descriptions détaillées des composants.

### 2.4.1 IDENTIFICATION DES ACTEURS

L'application "Al-Baladiya" vise à améliorer la communication entre les citoyens et les administrations municipales en permettant aux citoyens de soumettre des réclamations de manière anonyme et de suivre l'état de leurs réclamations. Les acteurs principaux de cette application sont les citoyens et les administrateurs municipaux. Voici une description détaillée

des rôles et responsabilités de chaque acteur :

### **Acteurs Principaux :**

#### **Citoyen**

Le citoyen est l'utilisateur principal de l'application. Son rôle est de soumettre des réclamations concernant des problèmes municipaux et de suivre l'état de ces réclamations. Les citoyens peuvent effectuer les actions suivantes :

- **Soumettre une Réclamation :** Les citoyens peuvent signaler des problèmes municipaux en soumettant des réclamations via l'application. La soumission peut inclure une description du problème et des images pour illustrer la réclamation.
- **Suivi des Réclamations :** Chaque réclamation est associée à un identifiant unique de la salle de chat (ID de la room) et à un identifiant utilisateur (ID de l'utilisateur). Les citoyens peuvent utiliser ces identifiants pour suivre l'état de leurs réclamations.
- **Interaction Anonyme :** Actuellement, les citoyens peuvent soumettre des réclamations de manière anonyme. Cependant, la perspective d'ajouter la possibilité de créer des comptes utilisateurs dans le futur est envisagée, ce qui permettrait une gestion plus personnalisée des réclamations et des interactions.

#### **Administrateur Municipal**

L'administrateur municipal est responsable de la gestion et du traitement des réclamations soumises par les citoyens. Son rôle inclut les responsabilités suivantes :

- **Réception et Gestion des Réclamations :** L'administrateur reçoit les réclamations soumises par les citoyens et les classe en fonction de leur nature et de leur urgence.
- **Réponse aux Réclamations :** L'administrateur peut répondre aux réclamations directement via l'application. Cette interaction se fait dans une salle de chat dédiée à chaque réclamation, permettant une communication en temps réel.
- **Suivi et Mise à Jour :** L'administrateur peut mettre à jour le statut des réclamations et informer les citoyens des actions entreprises pour résoudre les problèmes signalés.

## 2.4.2 DIAGRAMME DE CAS D'UTILISATION

Le diagramme de cas d'utilisation illustre les interactions entre les acteurs et le système, mettant en évidence les principales fonctionnalités accessibles à chaque acteur.

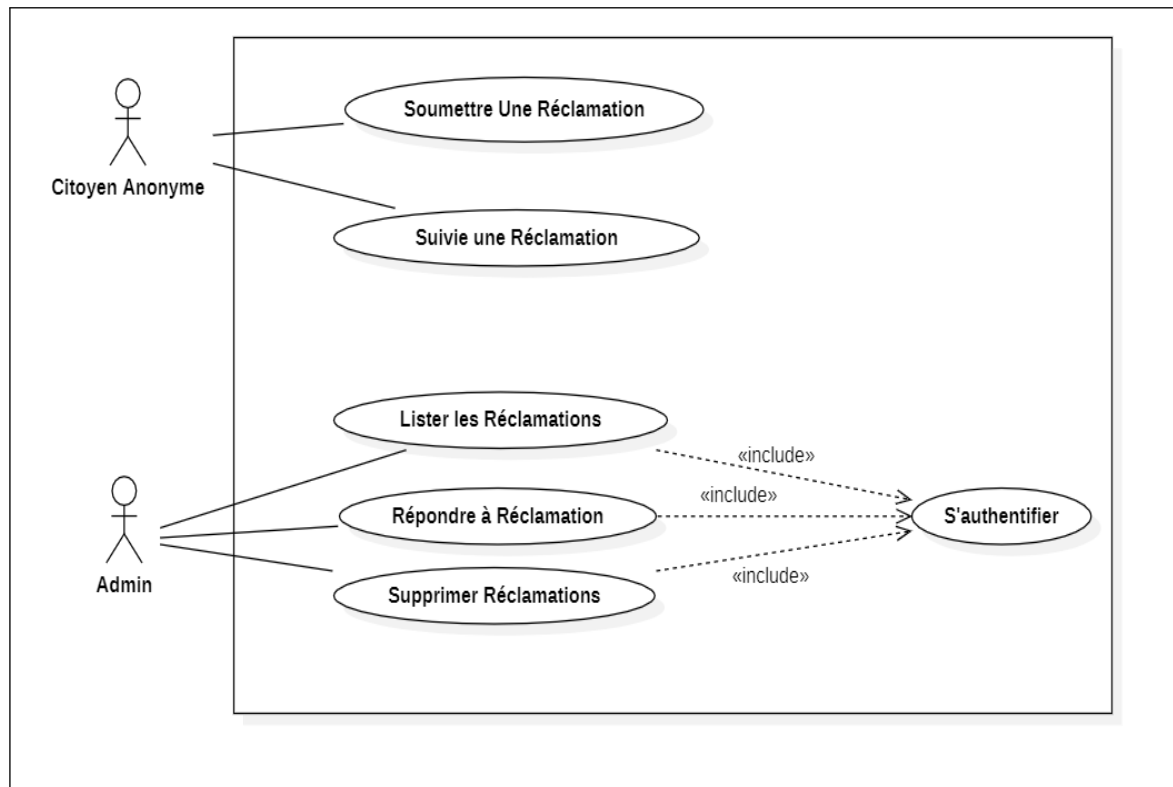


FIGURE 2.2 – Diagramme du cas d'utilisation

## 2.4.3 DESCRIPTION TEXTUELLE DES CAS D'UTILISATION

Cette section fournit une description textuelle détaillée des principaux cas d'utilisation de l'application "Al-Baladiya". Pour chaque cas d'utilisation, nous décrirons le scénario nominal (scénario principal) et les scénarios alternatifs (variantes ou exceptions).

### Cas d'Utilisation 1 : Soumettre une Réclamation

**Description :** Un citoyen soumet une réclamation concernant un problème municipal via l'application.

**Acteur Principal :** Citoyen

**Préconditions :** L'application est installée sur le dispositif du citoyen et l'utilisateur a accès à

Internet.

### **Scénario Nominal :**

- Le citoyen ouvre l'application "Al-Baladiya".
- Le citoyen sélectionne l'option "Soumettre une réclamation".
- L'application affiche un formulaire de réclamation.
- Le citoyen remplit les champs du formulaire (titre, description du problème, localisation).
- Le citoyen a la possibilité de joindre une ou plusieurs images pour illustrer la réclamation.
- Le citoyen soumet le formulaire.
- L'application envoie la réclamation à Firebase Firestore, où elle est stockée.
- L'application affiche un message de confirmation avec l'ID de la réclamation (ID de la salle de chat et ID de l'utilisateur) pour le suivi ultérieur.

### **Scénarios Alternatifs :**

- **Erreur de Connexion :** Si l'utilisateur n'a pas accès à Internet, l'application affiche un message d'erreur indiquant qu'une connexion Internet est requise pour soumettre une réclamation.
- **Champs Obligatoires Non Remplis :** Si le citoyen tente de soumettre le formulaire sans remplir tous les champs obligatoires, l'application affiche un message d'erreur demandant de compléter les informations manquantes.
- **Problème de Téléchargement d'Image :** Si l'application rencontre un problème lors du téléchargement des images, un message d'erreur s'affiche et le citoyen peut réessayer de soumettre la réclamation sans les images ou avec des images corrigées.

### **Cas d'Utilisation 2 : Suivre une Réclamation**

**Description :** Un citoyen suit l'état de sa réclamation en utilisant l'ID de la salle de chat et son ID utilisateur.

**Acteur Principal :** Citoyen

**Préconditions :** Le citoyen a soumis une réclamation précédemment et dispose de l'ID de la salle de chat et de son ID utilisateur.

### **Scénario Nominal :**

- Le citoyen ouvre l'application "Al-Baladiya".
- Le citoyen sélectionne l'option "Suivre une réclamation".
- L'application affiche un écran de suivi des réclamations.
- Le citoyen entre l'ID de la salle de chat et son ID utilisateur.
- L'application récupère les informations de la réclamation depuis Firebase Firestore.
- L'application affiche le statut actuel de la réclamation et toutes les communications de l'administrateur associées.

### Scénarios Alternatifs :

- **ID Incorrect** : Si le citoyen entre un ID de la salle de chat ou un ID utilisateur incorrect, l'application affiche un message d'erreur indiquant que les informations fournies sont invalides.
- **Réclamation Non Trouvée** : Si aucune réclamation correspondant aux ID fournis n'est trouvée, l'application informe le citoyen qu'aucune réclamation correspondante n'existe.

### Cas d'Utilisation 3 : Répondre à une Réclamation

**Description** : Un administrateur municipal répond à une réclamation soumise par un citoyen.

**Acteur Principal** : Administrateur Municipal

**Préconditions** : Une réclamation a été soumise par un citoyen et l'administrateur est connecté à l'application.

### Scénario Nominal :

- L'administrateur reçoit une notification via Firebase Cloud Messaging lorsqu'une nouvelle réclamation est soumise.
- L'administrateur ouvre l'application "Al-Baladiya".
- L'administrateur sélectionne l'option "Gérer les réclamations".
- L'application affiche la liste des réclamations soumises.
- L'administrateur sélectionne une réclamation spécifique à traiter.
- L'application affiche les détails de la réclamation.
- L'administrateur rédige une réponse et l'envoie via l'interface de chat.

- L'application enregistre la réponse dans Firebase Firestore et la notifie au citoyen via Firebase Cloud Messaging.

### Scénarios Alternatifs :

- **Réclamation Déjà Fermée** : Si l'administrateur tente de répondre à une réclamation qui a déjà été résolue et fermée, l'application affiche un message indiquant que la réclamation est déjà fermée.
- **Problème de Connexion** : Si l'administrateur perd la connexion Internet, l'application affiche un message d'erreur indiquant qu'une connexion est nécessaire pour envoyer la réponse.

## 2.4.4 DIAGRAMME DE SÉQUENCE

Avant de passer à l'implémentation de notre application "Al-Baladiya", il est crucial de visualiser les interactions entre les différents acteurs et le système. Les diagrammes de séquences ci-dessous illustrent les étapes et les échanges de messages lors de la soumission des réclamations, ainsi que l'identification des administrateurs.

### Authentification de l'Administrateur

Le diagramme suivant illustre le processus d'authentification pour un administrateur qui tente de se connecter à l'application.

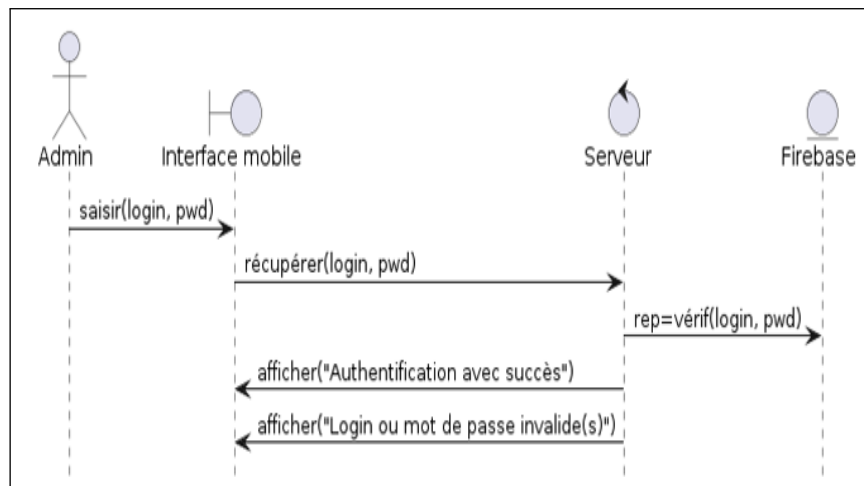


FIGURE 2.3 – Diagramme de séquence login

### Soumission d'une Réclamation

Le diagramme ci-dessous montre le processus de soumission d'une réclamation par un citoyen. L'utilisateur saisit les informations nécessaires, qui sont ensuite récupérées par l'interface mobile. Le serveur vérifie les informations avant de les enregistrer dans Firebase.

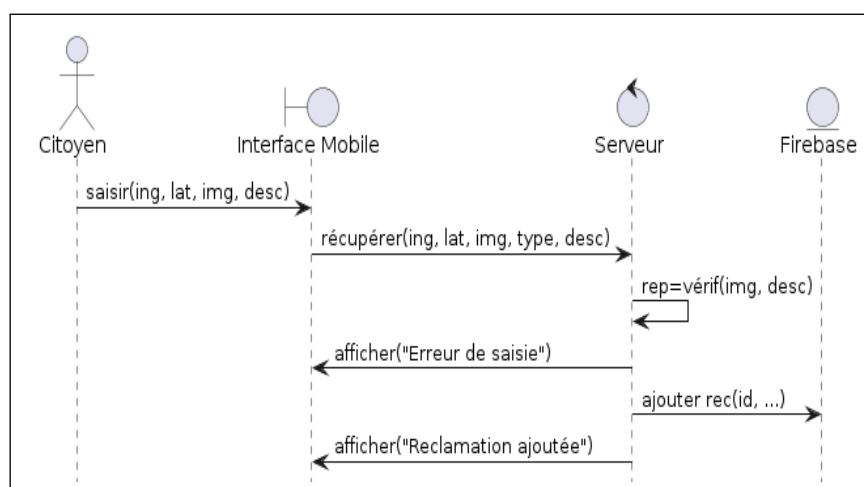


FIGURE 2.4 – Diagramme de séquence Soumettre réclamation

## 2.4.5 DIAGRAMME DE CLASSE

Le diagramme de classes ci-joint représente la structure statique du système en illustrant les classes, leurs attributs, méthodes et les relations entre elles. Pour l'application "Al-Baladiya", les principales classes incluent Personne, Citoyen, Administrateur, Municipalité, Compte,



Réclamation et Intervention. Voici une description détaillée de ces classes et de leurs relations.

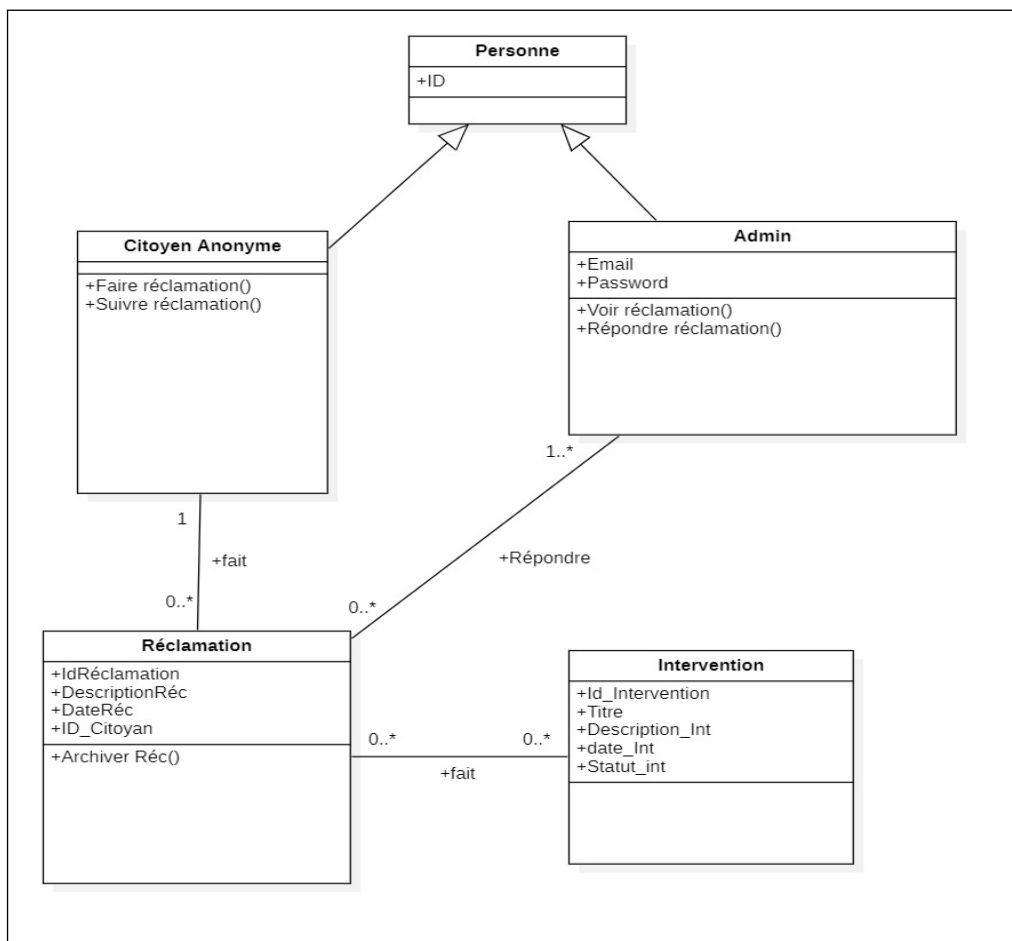


FIGURE 2.5 – Diagramme de classe

## 2.5 CONCLUSION

Ce chapitre a établi les bases de la conception détaillée de l'application "Al-Baladiya". Nous avons introduit l'UML comme outil de modélisation essentiel et décrit l'architecture technique, incluant le choix de Firebase Firestore pour la gestion des données. Les rôles des principaux acteurs, citoyens et administrateurs municipaux, ont été définis, ainsi que les cas d'utilisation principaux, détaillant les scénarios nominaux et alternatifs. Enfin, le diagramme de classes a illustré la structure statique du système. Cette approche structurée et méthodique garantit que l'application sera robuste, scalable et user-friendly, prête pour les phases de développement et de mise en œuvre.

---

# RÉALISATION

## Sommaire

---

<b>3.1</b>	<b>INTRODUCTION . . . . .</b>	<b>20</b>
<b>3.2</b>	<b>ENVIRONNEMENT DE TRAVAIL . . . . .</b>	<b>20</b>
3.2.1	ENVIRONNEMENT MATÉRIEL . . . . .	20
3.2.2	ENVIRONNEMENT LOGICIEL . . . . .	21
<b>3.3</b>	<b>LANGAGE DE PROGRAMMATION ET FRAMEWORK UTILISÉS . . . . .</b>	<b>22</b>
3.3.1	LANGAGE DE PROGRAMMATION . . . . .	22
3.3.2	FRAMEWORK . . . . .	22
<b>3.4</b>	<b>ÉTUDE COMPARATIVE DES TECHNOLOGIES UTILISÉES . . . . .</b>	<b>23</b>
3.4.1	CHOIX DE FLUTTER . . . . .	23
3.4.2	CHOIX DU FIREBASE . . . . .	23
3.4.3	POURQUOI FIRESTORE ET NON REALTIME DATABASE? . . . . .	24
<b>3.5</b>	<b>INTERFACE GRAPHIQUE ET EXPÉRIENCE UTILISATEUR . . . . .</b>	<b>25</b>
<b>3.6</b>	<b>ARCHITECTURE . . . . .</b>	<b>31</b>
3.6.1	ARCHITECTURE EN 3 TIERS . . . . .	32
3.6.2	CLIENT MOBILE (FRONTEND) . . . . .	33
3.6.3	FIREBASE . . . . .	33
<b>3.7</b>	<b>DÉPLOIEMENT . . . . .</b>	<b>33</b>
3.7.1	GIT ET GITHUB . . . . .	34
3.7.2	LES SOLUTIONS CLOUD . . . . .	36
3.7.3	CHOIX DE AWS . . . . .	36
3.7.4	PRÉPARATION POUR LE DÉPLOIEMENT . . . . .	37
3.7.5	DÉPLOIEMENT SUR AWS EC2 . . . . .	37
<b>3.8</b>	<b>CONCLUSION . . . . .</b>	<b>43</b>

---

### 3.1 INTRODUCTION

Ce troisième chapitre est consacré à la réalisation pratique de l'application "Al-Baladiya". Après avoir établi une conception détaillée dans le chapitre précédent, nous allons maintenant passer à la mise en œuvre technique. Cette phase comprend l'installation et la configuration des environnements de développement, le développement de l'application selon les spécifications définies, et la mise en place des différentes composantes techniques nécessaires pour assurer le bon fonctionnement de l'application.

### 3.2 ENVIRONNEMENT DE TRAVAIL

Pour développer efficacement l'application "Al-Baladiya", un environnement matériel et logiciel adéquat est essentiel. Cette section décrit les spécifications de l'environnement de développement utilisé.

#### 3.2.1 ENVIRONNEMENT MATÉRIEL

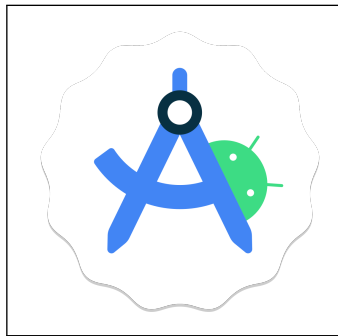
Pour le développement de l'application "Al-Baladiya", un environnement matériel adéquat est essentiel pour assurer des performances optimales. Les spécifications de l'environnement matériel utilisé sont les suivantes :

- Ordinateur : MSI
- Processeur : Intel Core i5 9ème Génération
- Carte Graphique : NVIDIA GTX 1050 TI
- Mémoire RAM : 16 GB
- Stockage : HDD : 1 TB, SSD : 256 GB

### 3.2.2 ENVIRONNEMENT LOGICIEL

L'environnement logiciel inclut les outils et les plateformes nécessaires pour le développement et le déploiement de l'application :

- **Android Studio** : IDE principal utilisé pour le développement de l'application Flutter.



**FIGURE 3.1** – *Logo Android Studio IDE*

- **Firebase** : Utilisé pour l'authentification, la base de données Firestore, le stockage et les notifications push.



**FIGURE 3.2** – *Logo Firebase*

- **Overleaf** : Utilisé pour la rédaction collaborative et la gestion des documents de projet.



**FIGURE 3.3** – *Logo Overleaf*

## 3.3 LANGAGE DE PROGRAMMATION ET FRAMEWORK UTILISÉS

Dans cette section, nous détaillerons les langages de programmation et les frameworks utilisés pour le développement de l'application "Al-Baladiya".

### 3.3.1 LANGAGE DE PROGRAMMATION

**Dart** : Utilisé pour le développement de l'application mobile avec Flutter. Dart est un langage de programmation optimisé pour les applications clientes sur plusieurs plateformes.



**FIGURE 3.4** – *Logo Dart*

### 3.3.2 FRAMEWORK

Flutter : Framework open-source développé par Google, utilisé pour créer des applications natives compilées pour mobile, web et bureau à partir d'une seule base de code.



**FIGURE 3.5** – *Logo Flutter*

## 3.4 ÉTUDE COMPARATIVE DES TECHNOLOGIES UTILISÉES

Pour le développement de l'application "Al-Baladiya", il a été nécessaire de choisir les technologies les plus appropriées pour répondre aux exigences du projet. Cette section présente une étude comparative des différentes options disponibles et explique pourquoi certaines technologies ont été préférées à d'autres.

### 3.4.1 CHOIX DE FLUTTER

**Flutter** a été choisi comme framework principal pour le développement de l'application pour les raisons suivantes :

- **Développement Multiplateforme** : Flutter permet de développer une application unique qui peut être déployée à la fois sur Android et iOS, ce qui réduit les efforts de développement et de maintenance.
- **Performance** : Les applications Flutter sont compilées en code natif, offrant des performances proches des applications natives.
- **UI Consistante** : Flutter utilise un moteur de rendu performant qui permet de créer des interfaces utilisateur riches et fluides.
- **Communauté Active** : Flutter bénéficie d'une communauté active et d'une documentation complète, facilitant le développement et la résolution des problèmes.

Comparé à d'autres frameworks comme React Native ou Xamarin, Flutter offre une combinaison unique de performance, flexibilité et facilité de développement, ce qui en fait le choix idéal pour ce projet.

### 3.4.2 CHOIX DU FIREBASE

Firebase a été choisi comme backend pour plusieurs raisons :

- **Intégration Facile** : Firebase s'intègre facilement avec Flutter, ce qui simplifie le développement.
- **Services Complets** : Firebase offre une suite complète de services backend, incluant l'authentification, la base de données en temps réel, le stockage de fichiers et les notifications push.
- **Scalabilité** : Firebase peut facilement évoluer pour gérer un nombre croissant d'utilisateurs et de données.
- **Sécurité** : Firebase propose des règles de sécurité robustes pour protéger les données utilisateur.

Par rapport à d'autres solutions backend comme AWS Amplify ou Backendless, Firebase offre une meilleure intégration avec Flutter et une suite de services plus complète.

## 3.4.3 POURQUOI FIRESTORE ET NON REALTIME DATABASE ?

Firestore a été préféré à Realtime Database pour plusieurs raisons :

Critère	Realtime Database	Firestore
Modèle de Données	✗ Simple JSON	✓ Documents et Collections
Scalabilité	✗ Limitée	✓ Excellente
Requêtes	✗ Simples	✓ Avancées
Synchronisation	✓ Temps réel	✓ Temps réel et hors ligne
Tarification	✗ Quantité de données	✓ Opérations de lecture/écriture
Support Multi-région	✗ Limité	✓ Multi-région
Transactions	✗ Limité	✓ Complet
Fonctionnalités Hors Ligne	✗ De base	✓ Robustes

**TABLE 3.1 – Comparaison entre Firebase Realtime Database et Firestore**

Firestore offre une meilleure flexibilité et des fonctionnalités plus avancées, ce qui est essentiel pour les besoins de l'application "Al-Baladiya".

## 3.5 INTERFACE GRAPHIQUE ET EXPÉRIENCE UTILISATEUR

Dans cette section, nous allons explorer les différentes interfaces graphiques de l'application "Albaladiya" et expliquer leur utilisation et objectif.

### Page d'accueil

Cette page d'accueil est la première interface que l'utilisateur voit en ouvrant l'application. Elle présente le logo de l'application et deux boutons pour accéder à l'espace client ou à l'espace administrateur.

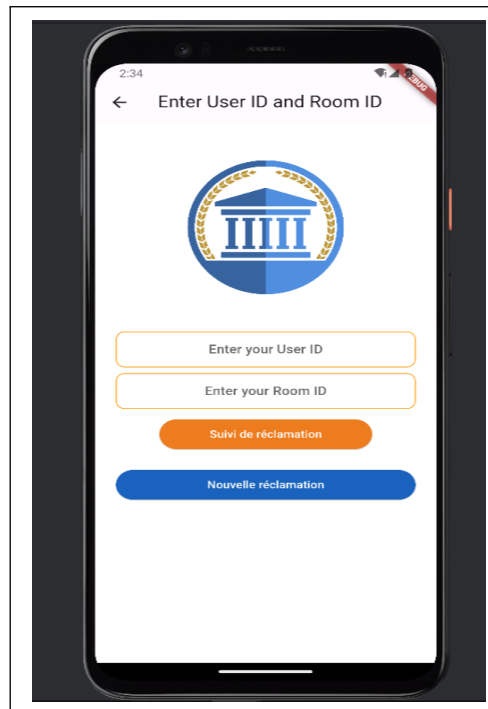


**FIGURE 3.6 – Page d'accueil**

### Sélection de l'action utilisateur

Après avoir sélectionné l'espace client, l'utilisateur est dirigé vers cette interface où il peut entrer son identifiant utilisateur et l'identifiant de sa réclamation. Il a également la possibilité de soumettre une nouvelle réclamation.

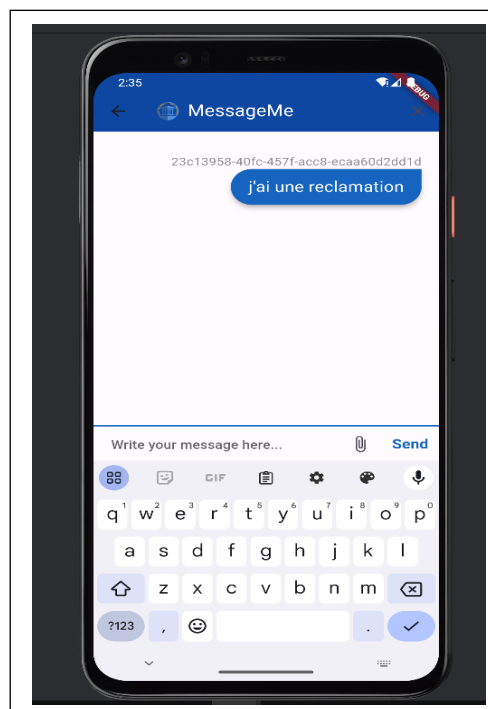




**FIGURE 3.7 – Espace Client**

## Soumission d'une réclamation

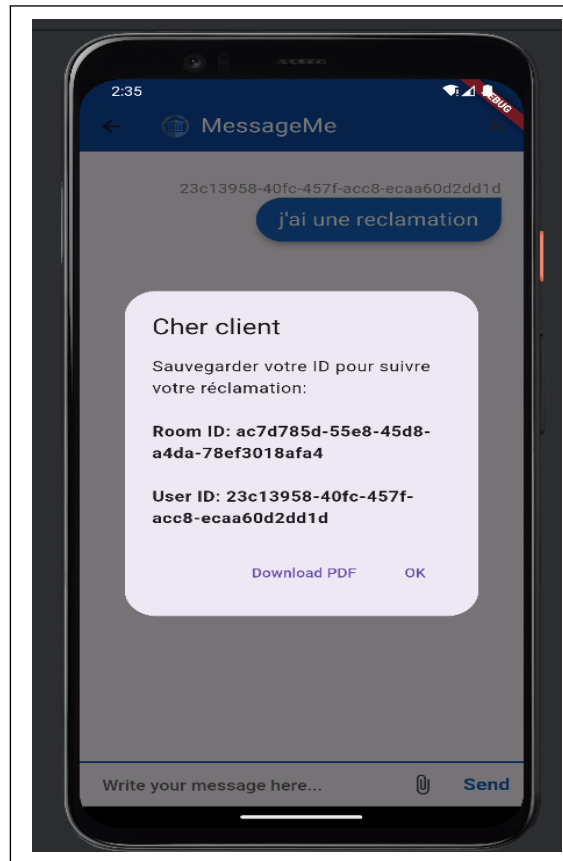
Cette interface permet à l'utilisateur de soumettre une réclamation en envoyant un message texte. Elle est simple et intuitive, facilitant la communication directe avec l'administration.



**FIGURE 3.8 – Espace reclamation**

### Confirmation de soumission

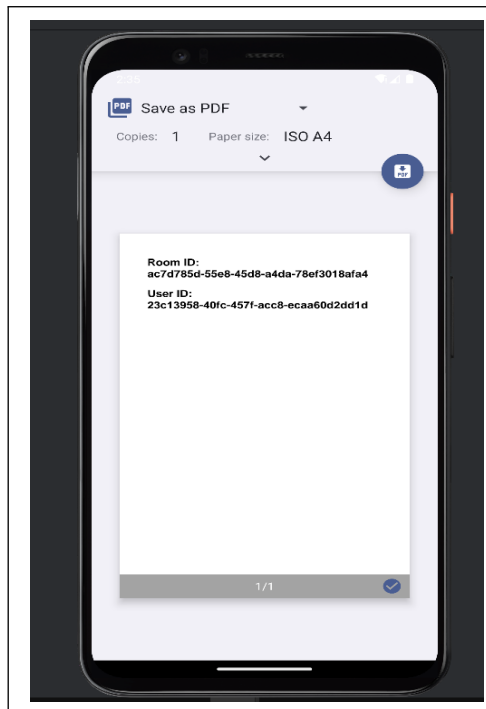
Les utilisateurs peuvent remplir et soumettre un formulaire de réclamation pour signaler un problème ou une demande spécifique.



**FIGURE 3.9** – *Message de Confirmation*

### Sauvegarde de l'ID de réclamation

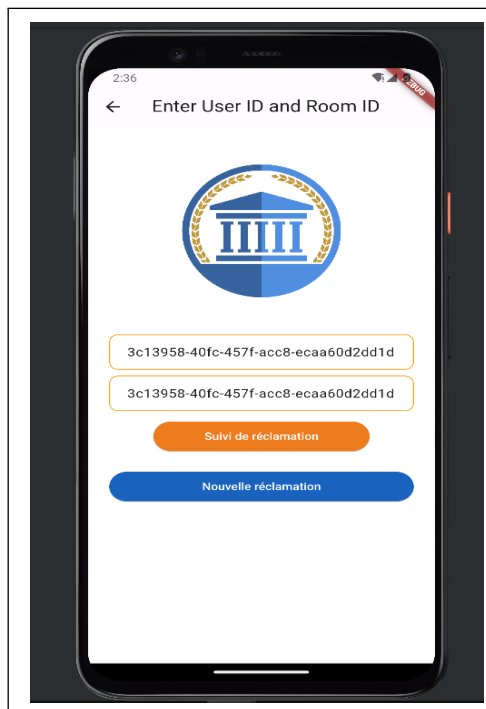
Cette interface permet à l'utilisateur de sauvegarder l'ID de sa réclamation sous forme de fichier PDF. Cette fonctionnalité est utile pour conserver une trace de la réclamation soumise.



**FIGURE 3.10** – *Téléchargement de PDF*

### Suivi de réclamation

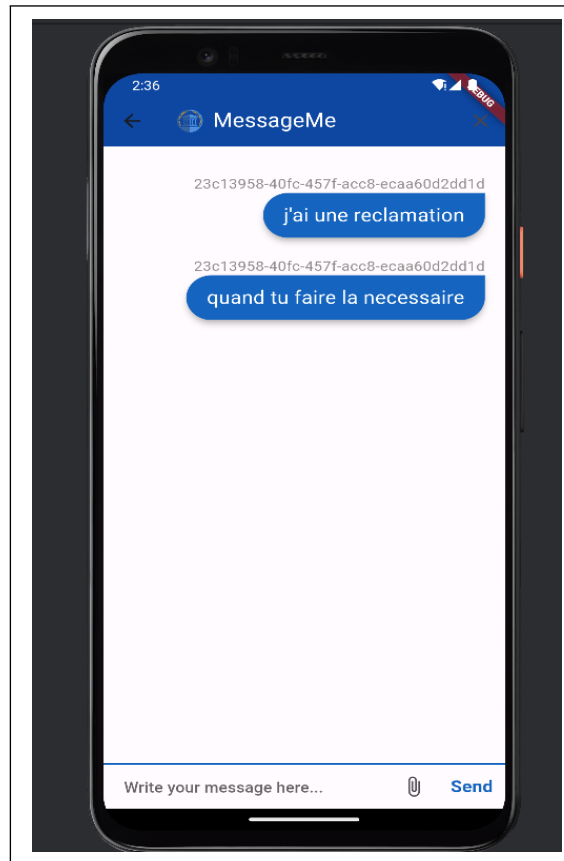
Cette interface permet à l'utilisateur de sauvegarder l'ID de sa réclamation sous forme de fichier PDF. Cette fonctionnalité est utile pour conserver une trace de la réclamation soumise.



**FIGURE 3.11** – *Remplissage des IDs*

### Suivi des messages

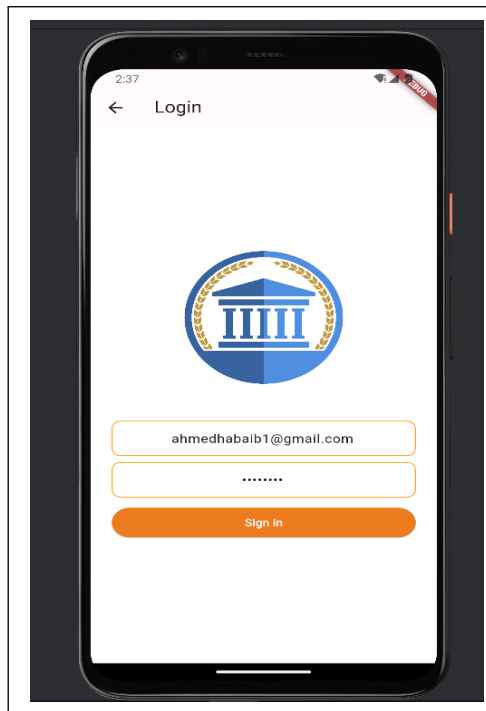
L'interface affiche les messages échangés entre l'utilisateur et l'administration concernant la réclamation. Cela permet une communication transparente et un suivi efficace des demandes.



**FIGURE 3.12** – *Espace reclamation pour la suivi*

### Page de connexion administrateur

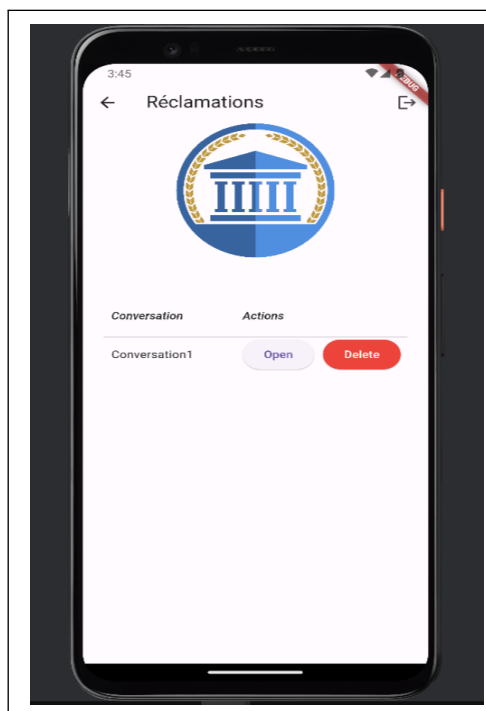
Cette page est destinée aux administrateurs pour se connecter à l'application en utilisant leurs identifiants. Elle assure que seules les personnes autorisées ont accès à l'espace d'administration.



**FIGURE 3.13** – *Identification d'administrateur*

### Liste des réclamations

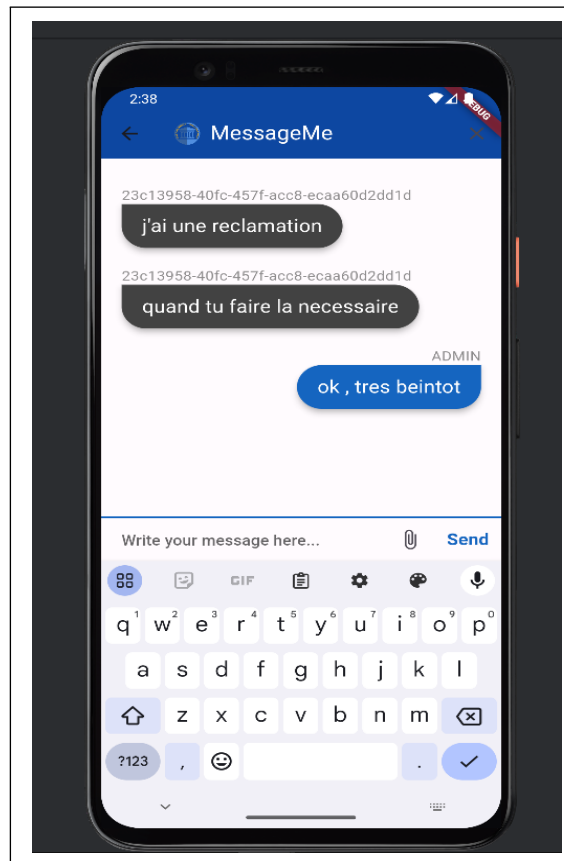
Après la connexion, l'administrateur voit une liste de toutes les réclamations soumises. Chaque réclamation peut être ouverte pour en voir les détails et y répondre.



**FIGURE 3.14** – *Liste des réclamations*

### Réponse aux réclamations

L'administrateur peut répondre aux réclamations directement depuis cette interface. Les réponses sont envoyées instantanément à l'utilisateur, facilitant ainsi la gestion des réclamations.



**FIGURE 3.15** – *Réponse d'admin*

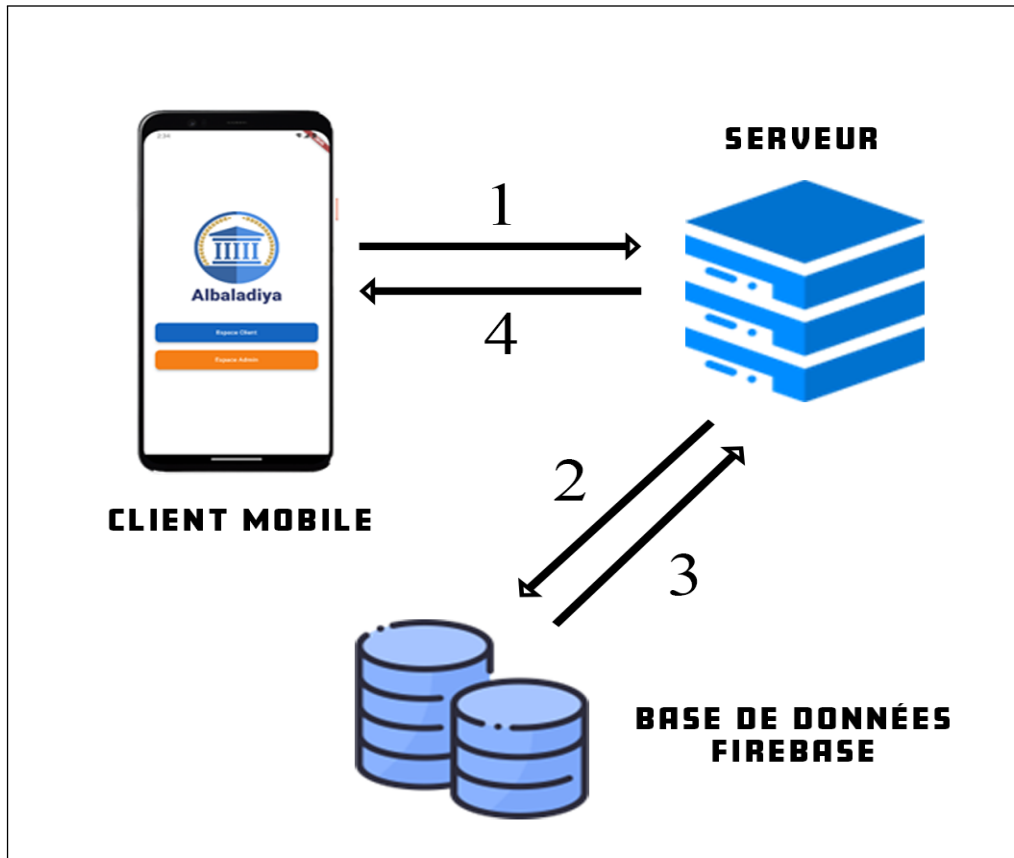
Ces interfaces ont été conçues pour offrir une expérience utilisateur fluide et intuitive, permettant une interaction efficace entre les citoyens et l'administration municipale.

## 3.6 ARCHITECTURE

L'architecture globale de l'application "Al-Baladiya" illustre l'ensemble des composants et comment ils interagissent les uns avec les autres.

### 3.6.1 ARCHITECTURE EN 3 TIERS

L'architecture de l'application est basée sur un modèle en 3 tiers, qui comprend le client mobile, le serveur et la base de données Firebase. Le diagramme ci-dessous illustre cette architecture.



**FIGURE 3.16** – Architecture en 3 Tiers de l'Application "Al-Baladiya"

Les interactions entre ces composants se déroulent comme suit :

1. Le client mobile envoie des requêtes au serveur.
2. Le serveur interagit avec Firebase pour stocker ou récupérer des données.
3. Firebase renvoie les données demandées au serveur.
4. Le serveur transmet les réponses appropriées au client mobile.

Grâce à cette architecture, l'application "Al-Baladiya" bénéficie d'une structure claire et évolutive, facilitant la gestion des différentes couches de l'application et assurant une communication efficace entre elles.

### 3.6.2 CLIENT MOBILE (FRONTEND)

Le client mobile, développé avec Flutter, est responsable de l'interface utilisateur et de la gestion des interactions avec les utilisateurs. Il communique avec Firebase pour les opérations backend.

### 3.6.3 FIREBASE

Pour le stockage et la gestion des données de l'application "Al-Baladiya", nous avons opté pour Firebase Firestore en raison de sa flexibilité, de sa scalabilité et de ses capacités de requêtes avancées. Cette solution permet de synchroniser les données en temps réel et de les stocker de manière sécurisée, tout en offrant une excellente performance même pour de grands volumes de données.

- **Firebase Authentication** : Gère l'authentification des utilisateurs.
- **Firestore Database** : Stocke les données des réclamations, des utilisateurs et des messages de chat.

## 3.7 DÉPLOIEMENT

Avant de procéder au déploiement de l'application "Al-Baladiya", il est important de noter que le projet est déjà importé sur GitHub, ce qui facilite la gestion du code source et la collaboration. Le déploiement se fera sur AWS EC2 après avoir comparé différentes solutions cloud disponibles.



### 3.7.1 GIT ET GITHUB

#### Définition de Git :

Git est un système de contrôle de version distribué conçu pour gérer tout projet, du plus petit au plus grand, avec rapidité et efficacité. Git permet à plusieurs développeurs de travailler simultanément sur le même projet, de suivre les modifications apportées au code source et de revenir à une version précédente si nécessaire.



FIGURE 3.17 – *Logo de Git*

#### Définition de GitHub :

GitHub est une plateforme de développement collaboratif qui utilise Git pour le contrôle de version. Il permet de stocker des dépôts de code source, de collaborer avec d'autres développeurs, de suivre les problèmes et les demandes de tirage, et bien plus encore.



FIGURE 3.18 – *Logo de GitHub*

#### Utilité de GitHub :

- **Collaboration** : GitHub permet à plusieurs développeurs de travailler sur le même projet en même temps, facilitant ainsi la collaboration et la contribution.

- **Gestion de version** : Avec GitHub, il est facile de suivre les modifications apportées au code source, de revenir à une version précédente si nécessaire et de gérer les différentes branches de développement.
- **Documentation** : GitHub offre des fonctionnalités de wiki et de pages GitHub pour créer une documentation complète et détaillée pour les projets.
- **Intégration continue** : GitHub peut être intégré avec des outils d'intégration continue pour automatiser les tests et les déploiements.

### GitHub Desktop :

GitHub Desktop est une application qui simplifie l'utilisation de GitHub en offrant une interface graphique conviviale. Elle permet de cloner des dépôts, de créer des branches, de faire des commits et de gérer des pull requests facilement.

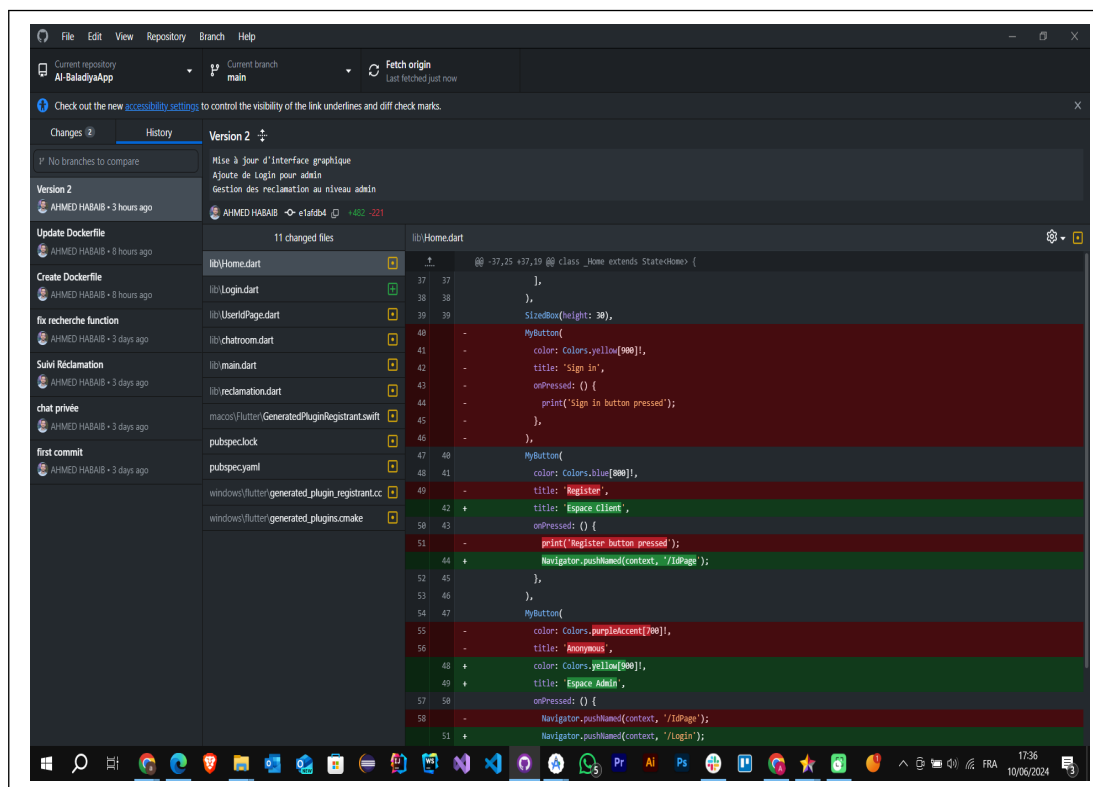


FIGURE 3.19 – Interface de GitHub Desktop

### 3.7.2 LES SOLUTIONS CLOUD

Avant de choisir Amazon AWS, nous avons examiné plusieurs options de cloud computing pour trouver la solution la plus adaptée à nos besoins. Voici une comparaison des principales caractéristiques des solutions cloud disponibles :

Critère	Amazon AWS	Google Cloud Platform (GCP)	Microsoft Azure
Échelle Mondiale	✓ Très étendue	✓ Très étendue	✓ Très étendue
Services Offerts	✓ Très diversifiés	✓ Très diversifiés	✓ Très diversifiés
Tarification	✓ Flexible, paiement à l'utilisation	✓ Flexible, paiement à l'utilisation	✓ Flexible, paiement à l'utilisation
Facilité d'Utilisation	✓ Documentation complète et interface intuitive	✓ Documentation complète et interface intuitive	✓ Documentation complète et interface intuitive
Support et Communauté	✓ Large communauté et support 24/7	✓ Large communauté et support 24/7	✓ Large communauté et support 24/7
Intégration avec Firebase	✓ Bonnes intégrations	✓ Excellentes intégrations	✓ Bonnes intégrations
Services d'IA et ML	✓ Avancés	✓ Très avancés	✓ Avancés
Sécurité	✓ Haute sécurité, certifications multiples	✓ Haute sécurité, certifications multiples	✓ Haute sécurité, certifications multiples

TABLE 3.2 – Comparaison entre Amazon AWS, Google Cloud Platform et Microsoft Azure

### 3.7.3 CHOIX DE AWS



FIGURE 3.20 – Logo AWS

Amazon Web Services (AWS) a été choisi pour le déploiement de l'application "Al-Baladiya" pour les raisons suivantes :

- **Échelle Mondiale** : AWS dispose d'une infrastructure mondiale avec de nombreuses régions et zones de disponibilité, offrant une haute disponibilité et une faible latence pour les utilisateurs.
- **Services Diversifiés** : AWS offre une large gamme de services qui peuvent être intégrés pour répondre aux besoins spécifiques du projet, y compris des services de calcul, de stockage, de base de données et de machine learning.
- **Tarification Flexible** : AWS propose des options de tarification pay-as-you-go, permettant de ne payer que pour les ressources utilisées, ce qui est économique pour des projets de toutes tailles.
- **Sécurité** : AWS fournit des mesures de sécurité robustes et des certifications conformes aux standards de l'industrie, assurant la protection des données et des applications.

### 3.7.4 PRÉPARATION POUR LE DÉPLOIEMENT

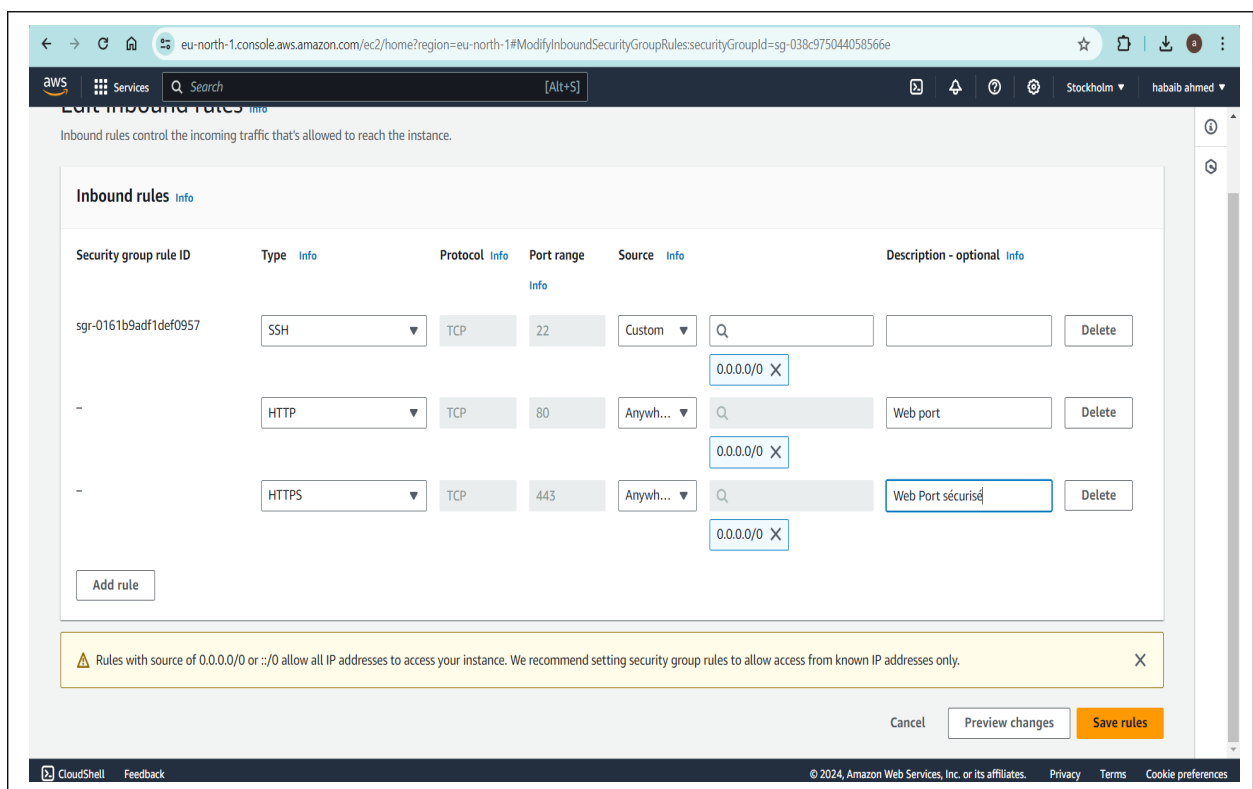
Avant le déploiement, il est crucial de s'assurer que toutes les configurations sont correctes et que l'application est prête pour la production. Voici les étapes à suivre pour préparer le déploiement sur AWS EC2 :

- **Configuration** : Vérifiez que toutes les configurations Firebase sont correctement intégrées dans l'application.
- **Build de Production** : Créez les builds de production pour Android et iOS en utilisant Android Studio.
- **Tests** : Assurez-vous que l'application a été minutieusement testée, y compris les tests unitaires, d'intégration et de performance.

### 3.7.5 DÉPLOIEMENT SUR AWS EC2

Le déploiement de l'application mobile Flutter "Al-Baladiya" sur une instance AWS EC2 exécutant Ubuntu et utilisant Nginx comme serveur web implique plusieurs étapes. Voici les étapes détaillées : **Étape 1 : Configuration de l'Instance EC2**

- Connectons-nous à notre console AWS et allons dans la section EC2.
- Lançons une nouvelle instance EC2 en suivant les étapes ci-dessous :
  - Choisissons une AMI (Amazon Machine Image). Une AMI Ubuntu est souvent une bonne option.
  - Choisissons le type d'instance (t3.micro est gratuit si nous sommes dans la limite de l'offre gratuite).
  - Configurons les détails de l'instance, le stockage et les balises si nécessaire.
  - Configurons un groupe de sécurité pour ouvrir les ports nécessaires (par exemple, HTTP, HTTPS, SSH).
  - Lançons l'instance et téléchargeons le fichier clé PEM pour nous connecter via SSH.
- Configuration du groupe de sécurité pour ouvrir les ports nécessaires :



**FIGURE 3.21** – Groupe de sécurité

- SSH (Port 22)
  - **Type** : SSH
  - **Protocole** : TCP

- **Plage de Ports** : 22
- **Source** : 0.0.0.0/0 (autorise l'accès depuis n'importe quelle adresse IP)
- **Description** : Permet les connexions SSH pour l'administration à distance de l'instance EC2.

Cette règle est essentielle pour pouvoir se connecter à l'instance via SSH à partir de notre terminal local.

- HTTP (Port 80)
  - **Type** : HTTP
  - **Protocole** : TCP
  - **Plage de Ports** : 80
  - **Source** : 0.0.0.0/0 (autorise l'accès depuis n'importe quelle adresse IP)
  - **Description** : Web port

Cette règle permet aux utilisateurs d'accéder à notre application web via HTTP (non sécurisé). Elle est nécessaire pour servir le contenu web aux utilisateurs.

- HTTPS (Port 443)
  - **Type** : HTTPS
  - **Protocole** : TCP
  - **Plage de Ports** : 443
  - **Source** : 0.0.0.0/0 (autorise l'accès depuis n'importe quelle adresse IP)
  - **Description** : Web Port sécurisé

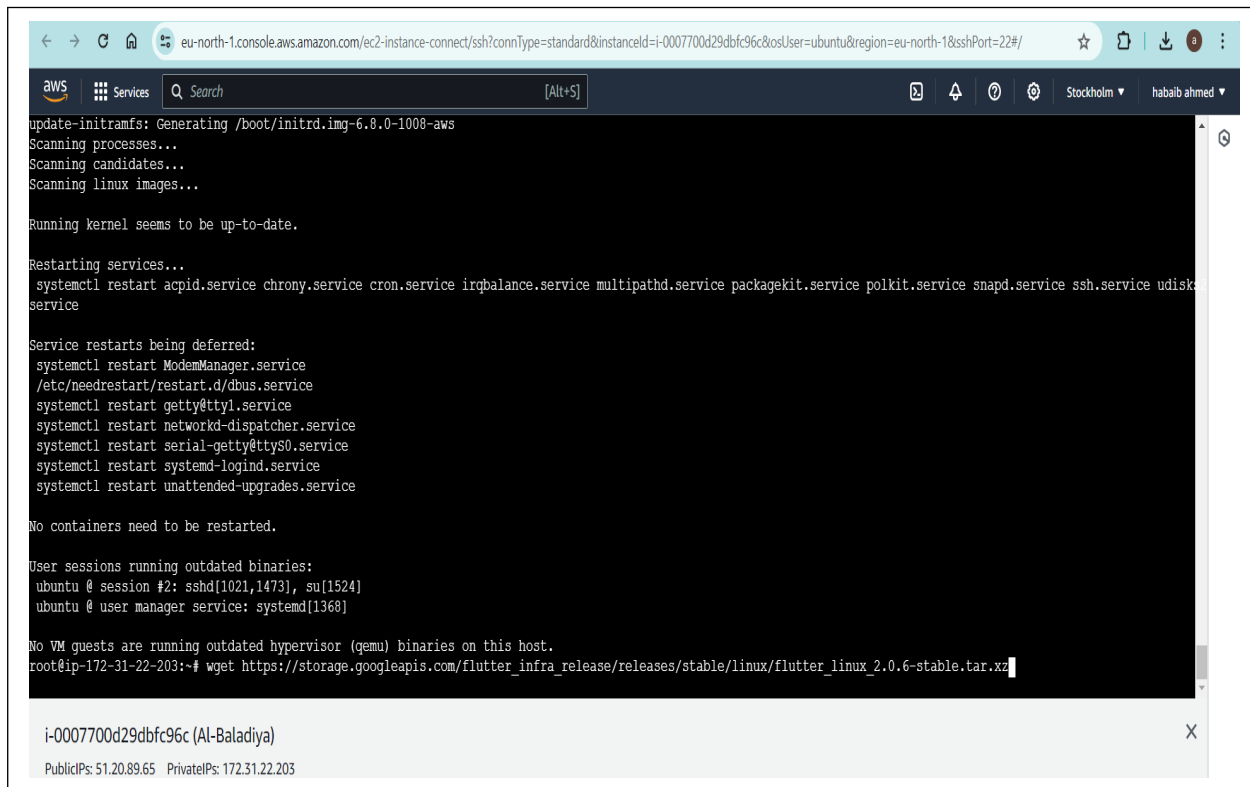
Cette règle permet aux utilisateurs d'accéder à notre application web via HTTPS (sécurisé). HTTPS assure que les données transmises entre le client et le serveur sont chiffrées et sécurisées.

## Étape 2 : Connexion à l'Instance EC2

Ouvrons un terminal sur notre machine locale.

## Étape 3 : Configuration de l'Environnement sur EC2

- Mettons à jour le système :
  - *sudo apt-get update*
  - *sudo apt-get upgrade -y*
- Installons Flutter :
  - Téléchargeons le SDK Flutter :



**FIGURE 3.22** – Téléchargement d'SDK Flutter

- Extrayons le fichier :
  - *tar xf flutter-linux.tar.xz*
- Ajoutons Flutter au PATH :
  - *export PATH="\$PATH : \$(pwd)/flutter/bin"*

- Installons les dépendances Flutter :
  - *sudo apt-get install -y clang cmake ninja-build pkg-config libgtk-3-dev*
- Clonons notre dépôt GitHub :
  - *git clone https://github.com/Ahmedhabaib/Al-BaladiyaApp.git*

### Étape 4 : Déploiement de l'Application

- Construisons notre application :
  - *flutter build web*
- Installons et configurons un serveur web (Nginx) pour servir notre application Flutter.
  - Pour Nginx :



```

eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0007700d29dbfc96c&osUser=ubuntu&region=eu-north-1&sshPort=22#/?
aws Services Search [Alt+S] Stockholm habaib ahmed
ubuntu @ session #2: sshd[1021,1473], su[1524]
ubuntu @ user manager service: systemd[1368]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-22-203:~# git clone https://github.com/Ahmedhabaib/Al-BaladiyaApp.git
Cloning into 'Al-BaladiyaApp'...
remote: Enumerating objects: 214, done.
remote: Counting objects: 100% (214/214), done.
remote: Compressing objects: 100% (148/148), done.
remote: Total 214 (delta 42), reused 205 (delta 33), pack-reused 0
Receiving objects: 100% (214/214), 419.14 KiB | 4.71 MiB/s, done.
Resolving deltas: 100% (42/42), done.
root@ip-172-31-22-203:~# ls
Al-BaladiyaApp flutter flutter linux 2.0.6-stable.tar.xz snap
root@ip-172-31-22-203:~# cd Al-BaladiyaApp
root@ip-172-31-22-203:~/Al-BaladiyaApp# flutter build web
Woah! You appear to be trying to run flutter as root.
We strongly recommend running the flutter tool without superuser privileges.
/
fatal: detected dubious ownership in repository at '/root/flutter'
To add an exception for this directory, call:

    git config --global --add safe.directory /root/flutter
Building flutter tool...
/root/flutter/bin/internal/shared.sh: line 161: 11340 Killed                  "$SDART" --verbosity=error --disable-dart-dev $FLUTTER_TOOL_ARGS --snapshot="$SNAPSHOT_
H" --packages="$FLUTTER_TOOLS_DIR/.packages" --no-enable-mirrors "$SCRIPT_PATH"
root@ip-172-31-22-203:~/Al-BaladiyaApp# sudo apt-get install nginx -y

```

i-0007700d29dbfc96c (Al-Baladiya)

PublicIPs: 51.20.89.65 PrivateIPs: 172.31.22.203

FIGURE 3.23 – Installation NGINX

- Configurons Nginx pour servir notre application Flutter :
- *sudo nano /etc/nginx/sites-available/default*

```

GNU nano 7.2 /etc/nginx/sites-available/default
#server {
#    listen 80;
#    listen [::]:80;
#
#    server_name example.com;
#
#    root /var/www/example.com;
#    index index.html;
#
#    location / {
#        try_files $uri $uri/ =404;
#    }
#}
server {
    listen 80;
    server_name votre-domaine.com;
    root /home/ubuntu/Al-BaladiyaApp/build/web;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-J To Bracket M-Q Previous
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^/_ Go To Line M-E Redo      M-G Copy      ^G Where Was  M-W Next
i-0007700d29dbfc96c (Al-Baladiya)
PublicIPs: 51.20.89.65 PrivateIPs: 172.31.22.203

```

FIGURE 3.24 – Fichier de configuration du serveur

— Redémarrons Nginx :

— *sudo systemctl restart nginx*

## 3.8 CONCLUSION

Dans ce chapitre, nous avons détaillé la réalisation de l'application "Al-Baladiya". Nous avons décrit l'environnement de travail, les langages et frameworks utilisés, ainsi que le choix de Firebase. Une comparaison des technologies a été faite, justifiant notre sélection de Firestore. Nous avons également expliqué le déploiement sur AWS EC2 et la configuration de Nginx. Enfin, une démonstration a illustré le fonctionnement de l'application. Ce chapitre montre que notre application est prête à être utilisée de manière sécurisée et performante.



---

## CONCLUSION GÉNÉRALE

Tout au long de la préparation de notre projet de fin d'année, nous avons appliqué les connaissances acquises durant nos études en ingénierie des systèmes informatiques à l'Institut International de Technologie. Le projet visait à développer une application mobile, "Al-Baladiya", pour permettre aux citoyens de soumettre des réclamations en ligne aux administrations municipales.

Nous avons commencé par une étude préalable pour comprendre les besoins et les solutions existantes. Cette analyse nous a permis de concevoir une application adaptée aux attentes des utilisateurs, en utilisant des technologies modernes telles que Flutter pour le développement de l'interface utilisateur et Firebase pour la gestion des données en temps réel.

La méthodologie suivie a inclus la définition des cas d'utilisation, la modélisation UML, et la conception d'une architecture système robuste. Nous avons ensuite détaillé la réalisation technique, en couvrant les aspects de l'environnement de développement, les langages de programmation et les frameworks utilisés, ainsi que les raisons de nos choix technologiques.

Le déploiement de l'application sur AWS EC2 a été une étape cruciale, assurant une infrastructure scalable et sécurisée. Nous avons configuré et testé l'application, démontrant son fonctionnement et sa capacité à répondre aux besoins des utilisateurs.

Ce projet a permis de mettre en pratique nos compétences techniques et de gestion de projet, et a posé les bases pour des améliorations futures. Les perspectives incluent l'ajout de

fonctionnalités avancées et l'intégration de services supplémentaires pour mieux répondre aux attentes des utilisateurs et des administrations municipales.

En conclusion, ce projet a été une expérience enrichissante, nous permettant de consolider nos acquis et de contribuer à une solution innovante pour améliorer la communication entre les citoyens et les administrations municipales.

# APPLICATION MOBILE FLUTTER ET DÉPLOIEMENT SUR CLOUD

---

---

**HABAIB Ahmed**

---

---

## **Résumé :**

Ce travail fait partie de notre projet de fin d'année pour la validation de la deuxième année de cycle d'ingénierie GLID à l'Institut International de Technologie. L'objectif de ce travail est le développement d'une application mobile nommée "Al-Baladiya", destinée à permettre aux citoyens de soumettre des réclamations en ligne et en temps réel aux administrations municipales. Les citoyens peuvent également joindre des images dans une salle de chat et suivre leurs réclamations à l'aide de l'ID de la salle. L'application a été développée en utilisant le framework Flutter et sera déployée sur l'infrastructure cloud d'Amazon EC2. Ce projet utilise Firebase pour la gestion des données et des notifications en temps réel.

**Mots clés :** Application mobile, Flutter, Cloud, Amazon EC2, Firebase, Réclamations en ligne.

## **Abstract :**

This work is part of the final project for the validation of the second year of the GLID engineering cycle at the International Institute of Technology. The objective of this work is the development of a mobile application named "Al-Baladiya", aimed at allowing citizens to submit complaints online and in real-time to municipal administrations. Citizens can also attach images in a chat room and track their complaints using the room ID. The application was developed using the Flutter framework and will be deployed on Amazon EC2 cloud infrastructure. This project uses Firebase for data management and real-time notifications.

**Key-words :** Mobile application, Flutter, Cloud, Amazon EC2, Firebase, Online complaints.