RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITÉ D'ABOMEY-CALAVI**

**INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE**

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.net Courriel : contact@ifri.uac.bj

# MÉMOIRE

pour l'obtention du

**Diplôme de Licence en Informatique**

**Option :** Génie logiciel

**Présenté par** :
Oluwatoyin Elie Isaac **VITODEGNI**

# Conception et réalisation d'une application mobile de recherche de garage automobile

**Sous la supervision** :
Dr Maurice **COMLAN**

**Membres du jury** :

| | | | |
|---|---|---|---|
| Pr Arnaud AHOUANDJINOU | Docteur | IFRI-UAC | Président |
| Mr Nelson SAHO | Docteur | IFRI-UAC | Examinateur |
| Mr Maurice COMLAN | Docteur | ENEAM-UAC | Rapporteur |

Année Académique : **2022-2023**

# Sommaire


Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Liste des figures vi

Liste des tableaux vii

Liste des acronymes viii

Introduction 1

1 Revue de littérature 3

2 Méthode de conception et choix techniques 7

3 Résultats et discussion 19

Conclusion 35

Bibliographie 36

Table des matières 37

# Dédicace

À ma famille, pour leur amour, leur soutien inconditionnel et leur encouragement constant tout au long de ce parcours.

# Remerciements

Je tiens à exprimer ma sincère gratitude envers les personnes et les organisations qui ont contribué à la réalisation de ce mémoire. Le chemin jusqu'à ce point a été marqué par le soutien, l'encouragement et la collaboration de nombreuses personnes, et je tiens à les remercier chaleureusement.

- Professeur Eugène C. EZIN, Directeur de l'Institut de Formation et de Recherche en Informatique (IFRI) ;
- Gaston EDAH Directeur Adjoint de l'Institut de Formation et de Recherche en Informatique (IFRI)
- Docteur Maurice COMLAN, mon encadreur pour sa disponibilité Constante, sa patience inlassable, son suivi attentif, et la confiance qu'il a eu en moi tout au long de ce parcours de recherche ;
- L'ensemble des enseignants de mon institut pour leur dévouement à partager une partie précieuse de leurs connaissances avec moi ;
- Mes Amis de classe, pour les discussions stimulantes, l'entraide et le partage de connaissances ;
- À mes amis proches, pour leur compréhension, leurs encouragements et les moments de détente qui ont équilibré cette aventure académique ;
- Toute ma famille, ainsi que tous mes amis qui m'ont toujours soutenu et encouragé dans mes choix d'étude ;

# Résumé

Dans un contexte où la mobilité revêt une importance primordiale, l'accès à des garages automobiles fiables s'avère crucial pour les conducteurs. Toutefois, avec l'avènement des smartphones et des applications mobiles, il est désormais possible de mettre à leur disposition un outil non seulement pour rechercher rapidement un garage, mais également pour trouver un garage de qualité en cas de besoin, évitant ainsi de se retrouver face à des établissements non qualifiés. L'objectif de notre travail est de fournir une plateforme conviviale et efficace permettant aux automobilistes de trouver rapidement des garages répondant à leurs besoins spécifiques en cas de panne ou d'entretien, tout en offrant aux garages des outils pour gérer efficacement leurs rendez-vous avec les clients. Pour atteindre ces objectifs, nous avons réalisé une étude approfondie sur les différents logiciels de recherche de garages existants, ce qui nous a permis de mettre en place notre solution intitulée "BonGarage". L'analyse des besoins de notre solution a été cruciale, comprenant l'énumération des fonctionnalités requises et l'utilisation de diagrammes UML (diagrammes de classe, de séquence et cas d'utilisation) pour les modéliser de manière précise. Nous avons sélectionné des technologies spécifiques, telles que Figma pour la conception des maquettes UI/UX, Flutter pour le développement de l'interface utilisateur et la logique applicative, et Firebase pour la gestion des données, l'authentification et la sécurité, le tout étant intégré et développé sous Android Studio pour garantir une architecture robuste et une fluidité dans l'exécution. Des simulations et des tests ont été réalisés pour évaluer la solution mise en place. Ce travail nous a ainsi permis de mettre à la disposition des automobilistes une solution conviviale pour leur satisfaction.

**Mots clés** :Application mobile, Recherche,Language de programmation, Automobiliste, garage

# Abstract

In a context where mobility is of paramount importance, access to reliable auto repair shops is crucial for drivers. With the advent of smartphones and mobile applications, it is now possible to provide them with a tool not only for quickly searching for a garage but also for finding a quality garage when needed, thus avoiding unqualified establishments. The objective of our work is to provide a user-friendly and efficient platform that allows motorists to quickly find garages that meet their specific needs in case of breakdown or maintenance, while offering garages tools to effectively manage their appointments with customers.To achieve these goals, we conducted a thorough study of existing garage search software, which allowed us to develop our solution titled "BonGarage." The analysis of our solution's requirements was crucial, involving the enumeration of necessary features and the use of UML diagrams (class, sequence, and use case diagrams) to model them accurately. We selected specific technologies, such as Figma for designing UI/UX mockups, Flutter for developing the user interface and application logic, and Firebase for data management, authentication, and security, all integrated and developed in Android Studio to ensure a robust architecture and smooth execution. Simulations and tests were conducted to evaluate the implemented solution. This work has enabled us to provide motorists with a user-friendly solution for their satisfaction.

**Key words**: Mobile Application, Search, Programming language ,Motorist,garage

# Liste des figures


1.1 SITE Go Africa online . . . . . 4
1.2 SITE VROOMLY . . . . . 5
1.3 Application OOVOOM . . . . . 5

2.1 Method Scrum . . . . . 8
2.2 Diagramme cas d'utilisation . . . . . 10
2.3 Diagramme de classe . . . . . 11
2.4 Diagramme de séquence du cas d'utilisation : ajout garage . . . . . 12
2.5 Diagramme de séquence du cas d'utilisation : Enregistrement garage . . . . . 13
2.6 Interaction entre le modèle vue et Contrôleur . . . . . 14

3.1 logo application . . . . . 20
3.2 page connexion . . . . . 21
3.3 Page choix du type de compte . . . . . 21
3.4 page d'inscription . . . . . 22
3.5 page de réinitialisation mots de passe . . . . . 23
3.6 page localisation . . . . . 24
3.7 Menu . . . . . 25
3.8 Page profile utilisateur . . . . . 25
3.9 page détails garage . . . . . 26
3.10 page détail garage . . . . . 26
3.11 page prise de rendez-vous . . . . . 27
3.12 Page FAQ . . . . . 28
3.13 Page contact admin . . . . . 29
3.14 page enregistrement garage . . . . . 30
3.15 page enregistrement garage . . . . . 30
3.16 page modification information . . . . . 31
3.17 page comfirmation rendez-vous . . . . . 31
3.18 Page connexion administrateur . . . . . 32
3.19 Page interface administrateur . . . . . 32
3.20 Bouton d'ajout et supression garage . . . . . 33
3.21 Page liste message utilisateur . . . . . 33

# Liste des tableaux

1.1 insuffisance des solutions éxistantes. . . . . 6

2.1 Comparaison entre Flutter, React Native et NativeScript . . . . . 16
2.2 Comparaison entre Firebase, AWS Amplify et Backendless . . . . . 16
2.3 Comparaison entre Android Studio, IntelliJ IDEA et Eclipse . . . . . 17
2.4 Comparaison entre Figma, Sketch et Adobe XD . . . . . 17
2.5 Comparaison entre UML et Merise . . . . . 18

# Liste des acronymes

**AAA :**

Authentication, Authorization, Accounting viii,

**FAQ :**

Frequently Asked Questions viii,

**IDE :**

Environnement de devellopement intégré viii,

**IOS :**

Iphone Operating System viii,

**MVC :**

Modèle Vue Contrôleur viii,

**NOSQL :**

Not Only SQL database viii,

**SQL :**

Structured Query Language viii,

**SSL :**

Secure Sockets Layer viii,

**TLS :**

Transport Layer Security viii,

**UML :**

Unified Modeling language viii,

# Introduction générale

Les technologies de l'information et de la communication, notamment le numérique, ont révolutionné les pratiques dans de nombreux domaines. Aucun secteur d'activité n'échappe à cette réalité. Les secteurs vitaux connaissent un essor grâce à l'intégration du numérique. L'industrie automobile ne fait pas exception à cette règle. Aujourd'hui, les véhicules mis à la disposition des consommateurs ne sont plus ceux d'il y a 20 ans, car ils intègrent assez de technologie, ce qui demande un savoir-faire spécifique pour intervenir en cas de panne.

## Contexte

Avec la croissance continue de l'industrie automobile et la fabrication des véhicules à la pointe de la technologie, la nécessité de maintenir et de réparer ces machines complexes est devenue une préoccupation majeure pour de nombreux propriétaires. Trouver un garage automobile fiable et compétent peut souvent être un défi de taille, obligeant les propriétaires à naviguer à travers un océan d'options souvent confuses et peu fiables.

L'idée de créer une application mobile dédiée à la recherche de garages vise à faciliter ce processus pour les propriétaires de voitures . L'application permettra aux utilisateurs de trouver rapidement et facilement des garages qui possèdent l'expertise et l'expérience nécessaires pour entretenir et réparer leurs Véhicules en cas de panne.

## Problématique

Aujourd'hui, les véhicules que nous utilisons sont de plus en plus électroniques et demandent un savoir-faire dans la réparation en cas de panne. Le constat est fait qu'en cas de panne, non seulement les automobilistes ont de difficulté à trouver rapidement le bon garage quand ils sont en dehors de leur zone de résidence habituelle ou non, mais aussi à trouver un garage de référence pouvant intervenir valablement sur le véhicule en panne. Ainsi, Comment donc concevoir et développer une application mobile qui répertorie et facilite la recherche des garages tout en garantissant une expérience utilisateur et répondant aux besoins spécifiques des propriétaires de véhicules ?

## Objectifs

**Objectif général**
l'objectif principal de ce travail est de permettre aux usagers de véhicules d'avoir recours aux garages qualifiés pour la résolution de leurs problèmes en cas de panne.
**Objectifs spécifiques**

- Développer une interface utilisateur conviviale;
- Intégrer la géolocalisation pour la recherche de garages enregistrés dans l'application;
- offrir à l'utilisateur la possibilité de prendre rendez-vous chez un garagiste ;
- Permettre à l'utilisateur de noter et de donner son avis sur un garage

## Organisation du mémoire

Ce document est structuré en trois chapitres. Le premier chapitre présente les solutions existantes et met en évidence l'intérêt de notre solution par rapport à celles déjà disponibles. Le deuxième chapitre expose notre solution à travers sa modélisation, son fonctionnement et les choix techniques effectués pour sa réalisation. Enfin, le troisième chapitre présente les résultats obtenus et une analyse critique sur ces derniers.

Chapitre 1

# Revue de littérature

## Introduction

Dans ce chapitre, nous définissons quelques concepts clés de notre thème de recherche et réalisons une étude de l'existant afin de mieux cerner le problème auquel nous voulons apporter une solution. Cela nous permet de mettre en évidence les insuffisances des solutions existantes et d'exposer l'intérêt de notre solution par rapport à celles-ci.

## 1.1 Définition des concepts clés

**Application mobile :** une application mobile est un logiciel informatique conçu pour fonctionner sur des appareils mobiles tels que des smartphones, des tablettes et des montres connectées. Ces applications sont spécifiquement adaptées aux caractéristiques et aux contraintes des dispositifs mobiles, notamment leur petit écran tactile, leur puissance de calcul limitée, et leur connectivité sans fil. Les applications mobiles sont conçues avec des langages de programmation comme le java et des Framework comme flutter[4].
**Garage automobile :** Un garage automobile est un établissement commercial ou un atelier spécialisé dans la réparation, l'entretien, et parfois la vente de véhicules automobiles, y compris des voitures particulières, des camions, des motos et parfois d'autres types de véhicules motorisés. Ces établissements sont essentiels pour assurer la maintenance, la réparation, et le bon fonctionnement des véhicules à moteur[6].
**Réparation de véhicules :** La réparation de véhicules désigne l'ensemble des opérations et des travaux réalisés pour restaurer ou remettre en état un véhicule automobile (voiture, camion,etc.) qui présente des problèmes mécaniques, électriques, électroniques, carrosserie, ou d'autres types de dysfonctionnements. Ces réparations visent à restaurer la fonctionnalité, la sécurité, et l'apparence du véhicule, le rendant ainsi apte à être utilisé sur la route de manière efficace et sécuritaire.
**Service automobile :** Les services automobiles désignent un ensemble de prestations et d'activités liées à la maintenance, à la réparation, à l'entretien et à la gestion des véhicules automobiles[5].

## 1.2 Présentation des solutions existantes

Pour faciliter le processus de recherche et de réservation de services automobile et avoir un service de qualité pour les voitures, plusieurs développeurs ont essayé d'apporter des solutions. Il s'agit entre autres des applications telles que :

**Go Africa Online :** Go Africa online est une plateforme en ligne qui développe et popularise des services en ligne qui permettent de favoriser les interactions économiques entre tous les professionnels Africains y compris le Bénin. Sur Go Africa online, il est possible de sélectionner le pays de résidence, puis de rechercher les entreprises d'un domaine spécifique. Parmi ces entreprises, on retrouve celles dédiées aux garages automobiles existant au Bénin.

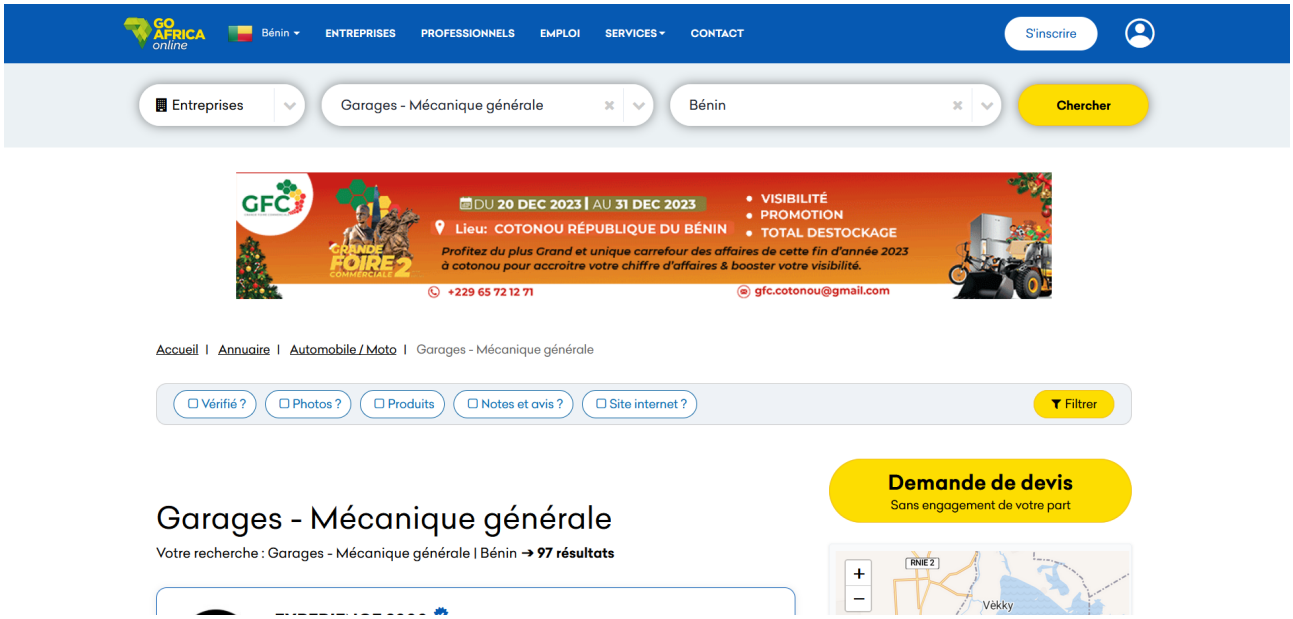



FIGURE 1.1 : SITE Go Africa online

**Vroomly :** Vroomly est une plateforme en ligne qui propose des services liés à l'entretien automobile. Elle permet aux utilisateurs de trouver des garages près de chez eux, de comparer les tarifs des réparations et de lire les avis des clients. Le site Vroomly vise à faciliter le processus de recherche et de réservation de services automobiles, en offrant aux conducteurs une vue d'ensemble des options disponibles pour l'entretien de leur véhicule.

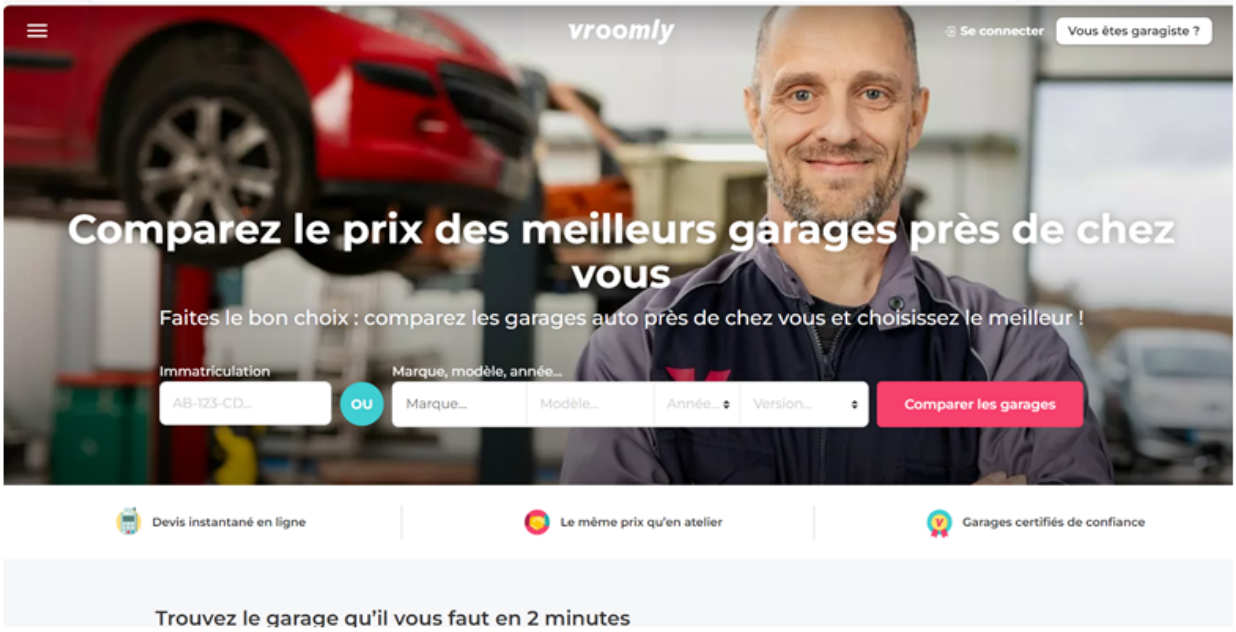



FIGURE 1.2 : SITE VROOMLY

**OOVOOM :** est une application mobile qui permet de trouver facilement le meilleur garagiste près de son lieu de résidence et de centraliser toutes les informations sur le véhicule. L'application a pour vocation de mettre le digital au service de la relation client-garage. Elle permet d'enregistrer toutes les informations nécessaires pour gérer les véhicules (date du dernier contrôle technique, anciennes factures de réparation pour mettre à jour le carnet d'entretien...) et de prendre directement rendez-vous chez un garagiste en ligne sur l'application. En cas de panne, pour un entretien ou une révision, il est possible de rechercher le garage le plus proche pour effectuer la réparation et choisir celui qui convient le mieux. Il y est possible également de demander gratuitement un devis en ligne et recevoir le tarif et détail de la prestation sur l'application.

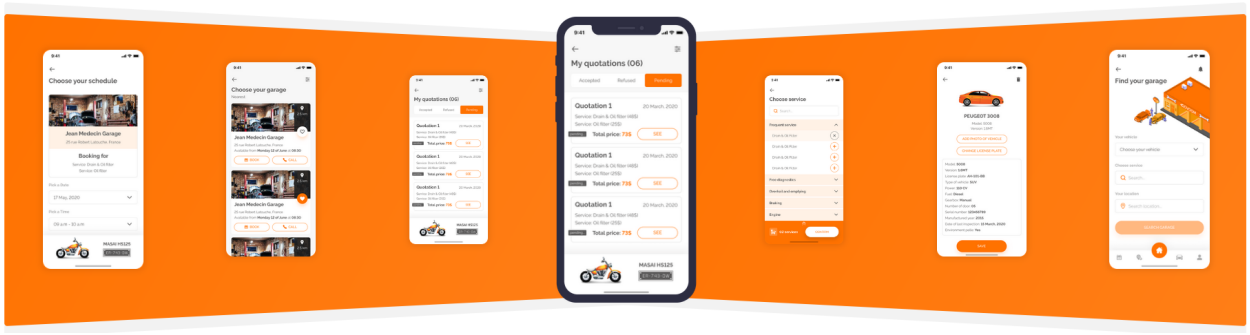


FIGURE 1.3 : Application OOVOOM

## 1.3 Insuffisances des solutions existantes

Les solutions que nous venons de présenter comportent des avantages, mais également des lacunes. Lors de notre évaluation des solutions existantes, nous avons constaté plusieurs insuffisances, à savoir.

**Go Africa Online :**
version mobile inexistante. La recherche de garage n'est pas assez fournie et les informations sur les garages ne sont pas exhaustives.
**Vroomly :**
Version mobile inexistante.
Informations et évaluations limitées : Les avis et les informations sur les garages peuvent ne pas toujours être exhaustifs ou à jour, ce qui peut rendre difficile la prise d'une décision éclairée lors du choix d'un garage.
**OOVOOM :**
Couverture géographique limitée, car on n'y trouve pas les garages partenaires potentiels pouvant se retrouver au Bénin ou ailleurs.

| Solution | Version mobile | Géolocalisation | complexité d'utilisation |
|---|---|---|---|
| Vroomly | non | oui | oui |
| OOVOOM | oui | non | non |
| Go Africa online | non | oui | oui |

TABLE 1.1 : insuffisance des solutions éxistantes.

## 1.4 Intérêt de la solution par rapport aux existantes

L'application mobile à mettre en ligne est une première dans notre pays. Elle se différencie aux applications évoquées précédemment de par son adaptation aux réalités nationales et aux besoins des clients internes. Aussi, certaines applications en dehors de **go africa online** n'ont pas une couverture vers notre pays ni ne référencient pas des garages partenaires nationaux. Elles ne font pas référence de facon spécifique dans les recherches aux garages spécialisés comme l'auraient voulu certains automobilistes.

## Conclusion

Ce chapitre nous a permis d'examiner plusieurs solutions en vue d'apporter une amélioration à notre application pour une bonne expérience utilisateur. La finalité de notre travail est de proposer une solution plus optimale et adaptée aux réalités des automobilistes du Bénin.

Chapitre 2

# Méthode de conception et choix techniques

## Introduction

Dans ce chapitre, nous analysons notre projet, ce qui permet d'exposer ses besoins fonctionnels et non fonctionnels, ainsi que de détailler le processus métier associé. Ensuite, nous présentons les différentes méthodes utilisées pour la conception (modélisation) d'un système, y compris celle utilisée pour notre application. Enfin, nous abordons les choix techniques pour la réalisation du projet.

## 2.1 Analyse

L'analyse est naturellement la première étape d'un projet informatique parce qu'elle nous permet de rassembler toutes les informations permettant d'évaluer la faisabilité et le potentiel de notre produit technologique sur le marché.

### 2.1.1 Methodologie de développement

Dans la réalisation de ce projet, l'ensemble de méthodes et pratiques basées sur les valeurs et les principes du Manifeste Agile (Méthode Scrum) fut la méthodologie appropriée et utilisée. C'est une méthode de développement agile orientée projet informatique dont les ressources sont régulièrement actualisées pour faire face aux imprévus pouvant subvenir et concevoir un produit de qualité.

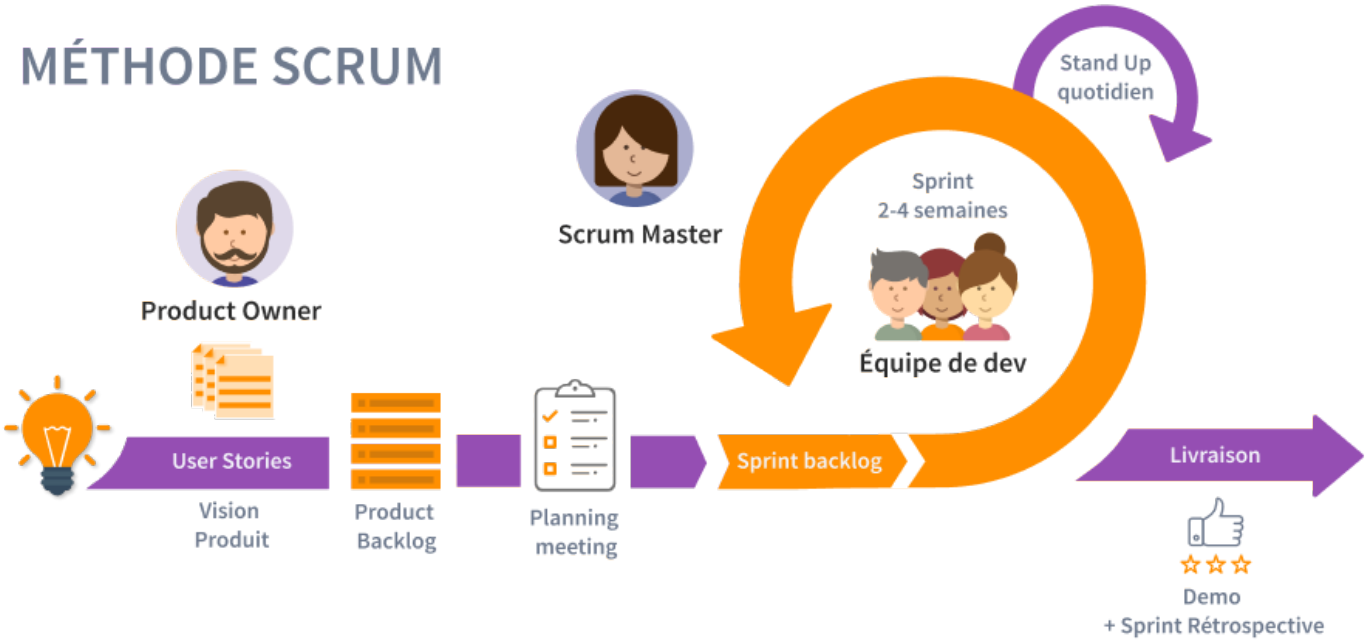



FIGURE 2.1 : Method Scrum

### 2.1.2 Les besoins fonctionnels

Les besoins fonctionnels sont ceux qui précisent ce que le système doit faire. Ils expriment les fonctionnalités concrètes de notre projet. Notre application mobile nous permettra de :

- inscrire chaque utilisateur ;
- inscrire les garagistes ;
- effectuer une demande d'enregistrement de garages ;
- permettre aux utilisateurs de s'authentifier ;
- Trouver les garages en utilisant la géolocalisation ;
- contacter un garage ;
- contacter l'administrateur ;
- consulter une FAQ ;
- prendre rendez-vous chez un garagiste ;
- modifier les informations d'un garage par le garagiste ;
- noter un garage ;
- laisser des commentaires sur un garage ;
- ajouter, supprimer un garage par l'administrateur

### 2.1.3 Les besoins non fonctionnels

Les besoins non fonctionnels caractérisent le système en matière de performance, de matériels, de contraintes d'implémentation et du système d'exploitation. Ils traitent également, des besoins de disponibilité, de sécurité et de déploiement de la solution. En matière de système d'exploitation notre système sera déployé à la fois sous les systèmes Android et iOS.
S'agissant de la sécurité, les données des utilisateurs seront protégées par un système de chiffrement des données et une authentification basée sur le protocole AAA ( Authentification, Authorization, Accounting) inclus dans Firebase.

### 2.1.4 Processus métier

Le processus métier d'un garage automobile, incluant la gestion des pannes sur la route, se déroule en plusieurs étapes clés pour garantir un service optimal.

Tout d'abord, lorsqu'un client se présente au garage ou signale une panne, il est pris en charge par un conseiller qui recueille les informations nécessaires. Si la panne survient en dehors du garage, le client contacte l'établissement pour signaler l'incident. Le garage envoie alors un technicien pour un dépannage rapide ou organise le remorquage du véhicule si une réparation sur place est impossible.

Une fois le véhicule au garage, un diagnostic approfondi est réalisé pour identifier les problèmes. Un devis est ensuite établi et soumis au client pour validation. Lorsque ce devis est accepté, l'intervention est planifiée en fonction de la disponibilité des pièces et des mécaniciens.

Les réparations sont effectuées selon un ordre de réparation précis, et si des problèmes supplémentaires sont découverts, le garage contacte le client pour obtenir son approbation avant de continuer. Une fois les réparations terminées, des tests de contrôle qualité sont réalisés pour s'assurer du bon fonctionnement du véhicule.

Le véhicule est ensuite restitué au client, avec une explication détaillée des travaux effectués et la facture correspondante. Si le véhicule a été réparé après une panne sur la route et que le client ne peut pas se déplacer, le garage peut également proposer de livrer le véhicule.

Parallèlement, le garage gère ses stocks de pièces détachées pour garantir une disponibilité rapide des composants nécessaires. La facturation et la comptabilité sont également suivies de près pour garantir une gestion efficace des finances. Enfin, la gestion des plannings du personnel assure que chaque tâche est exécutée dans les délais avec les ressources adéquates.

## 2.2 Méthodes de conception

La conception tient une place primordiale dans la réalisation d'un projet. Elle nous permet de structurer et d'ordonnancer les différentes étapes de faisabilité du projet. Pour y parvenir, il est essentiel d'utiliser un langage de modélisation. Nous avons opté pour notre projet, le langage UML (Unified Modeling Langage) qui nous permet de représenter sous forme de diagrammes notre application

mobile à développer. L'UML permet de faire l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que par leur comportement[7]. Avec ce langage nous avons modélisé les diagrammes suivants :

### 2.2.1 Diagramme de cas d'utilisation

Le diagramme que voici rend plus compréhensible les caractéristiques, les objectifs, les fonctionnalités et les besoins de notre application. Il nous montre les différentes interactions entre elle et les acteurs(Administateur, Utilisateur, Garagiste).

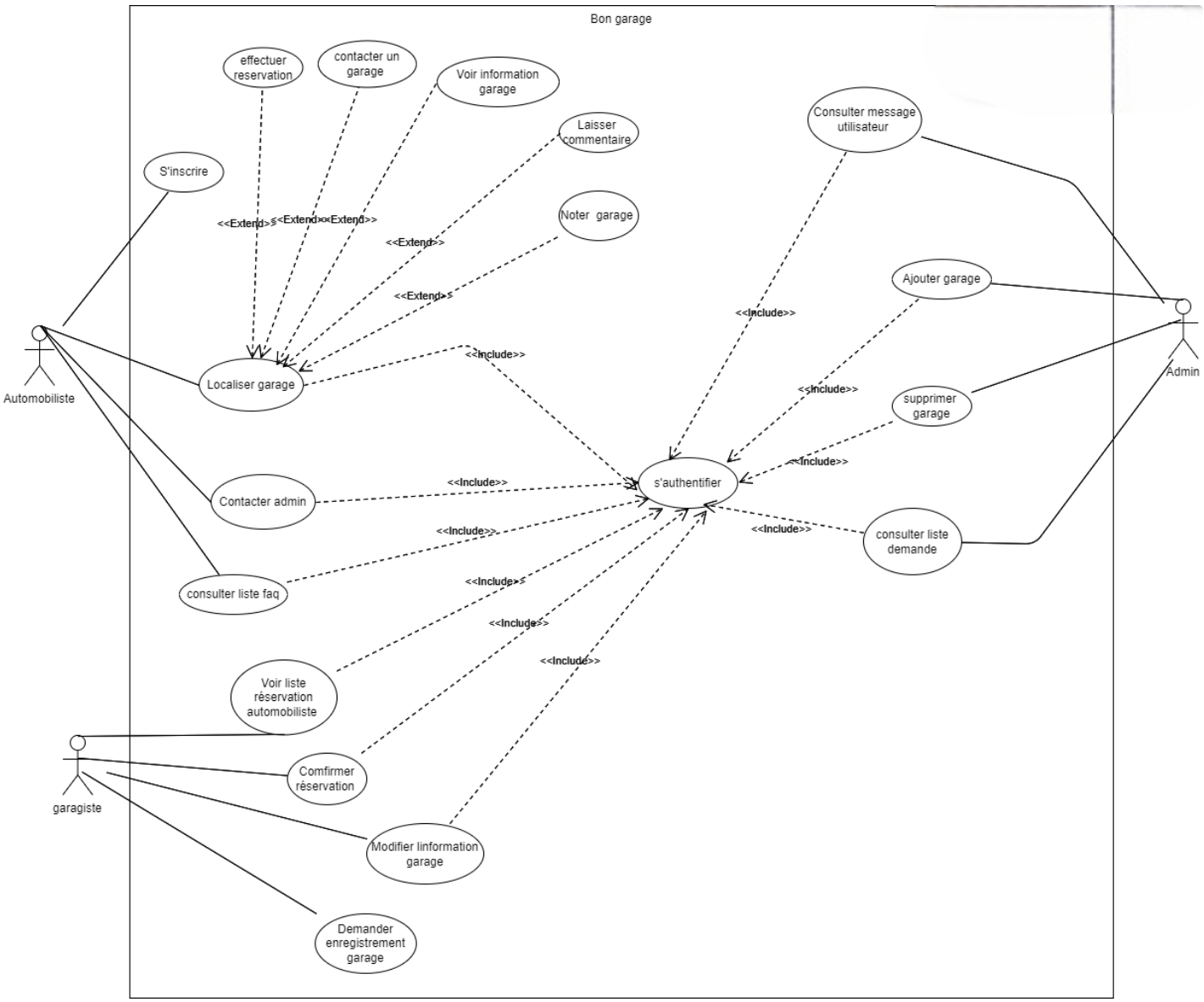



FIGURE 2.2 : Diagramme cas d'utilisation

### 2.2.2 Diagramme de classes

Le diagramme de classe représente les classes et les différentes relations et interactions entre ces classes.

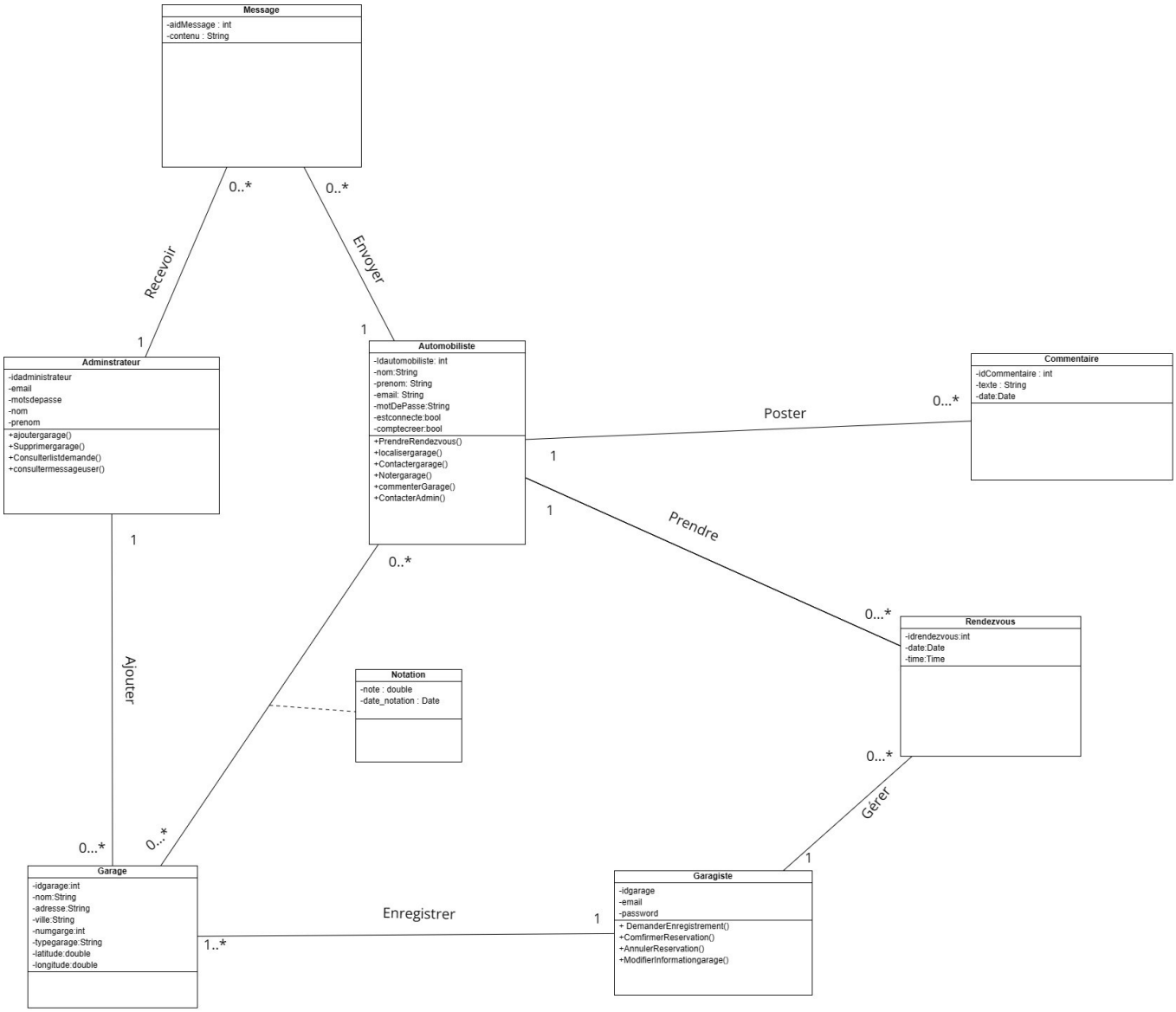



FIGURE 2.3 : Diagramme de classe

### 2.2.3 Diagramme de séquence

Le diagramme de séquence est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.

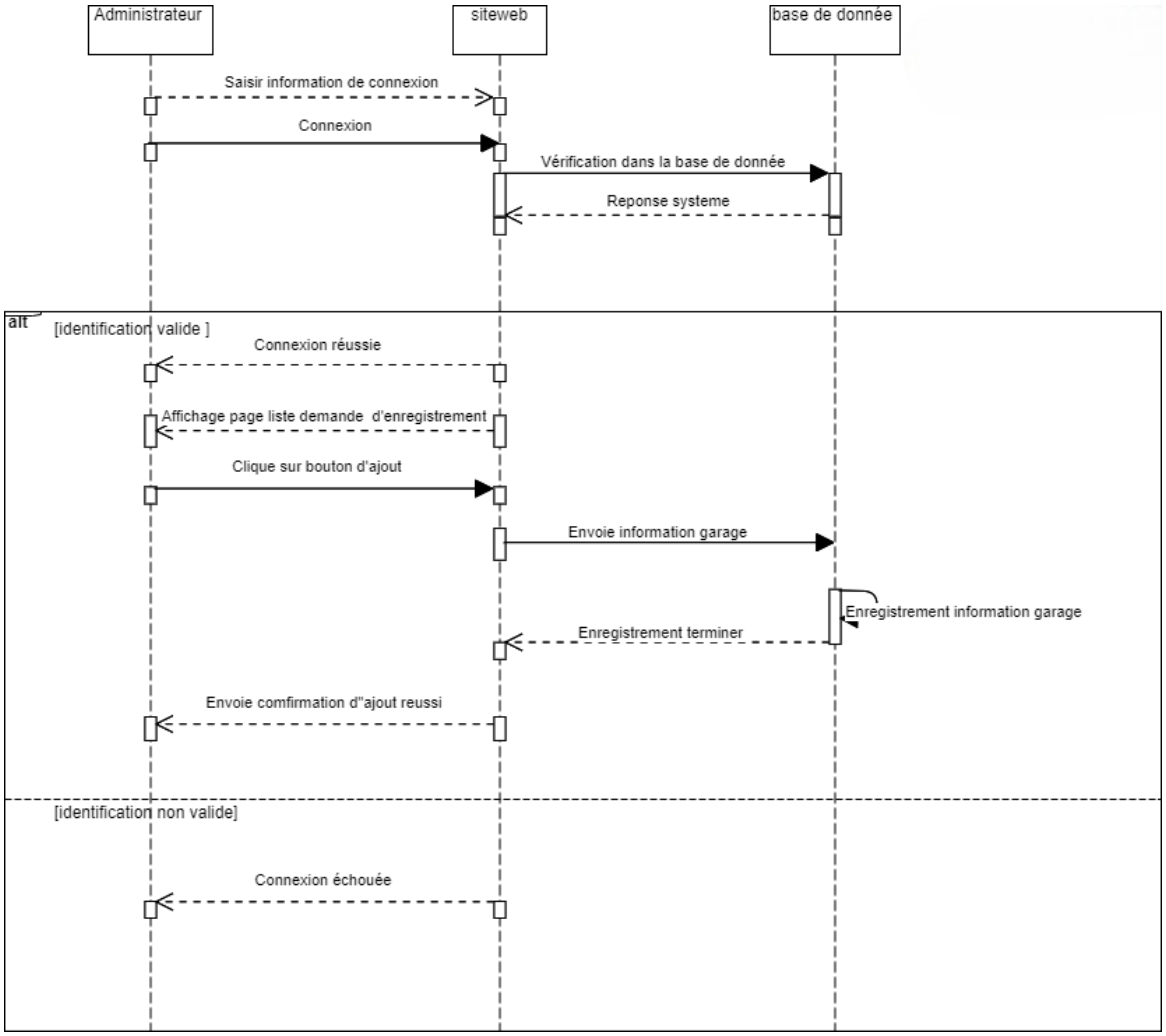



FIGURE 2.4 : Diagramme de séquence du cas d'utilisation : ajout garage

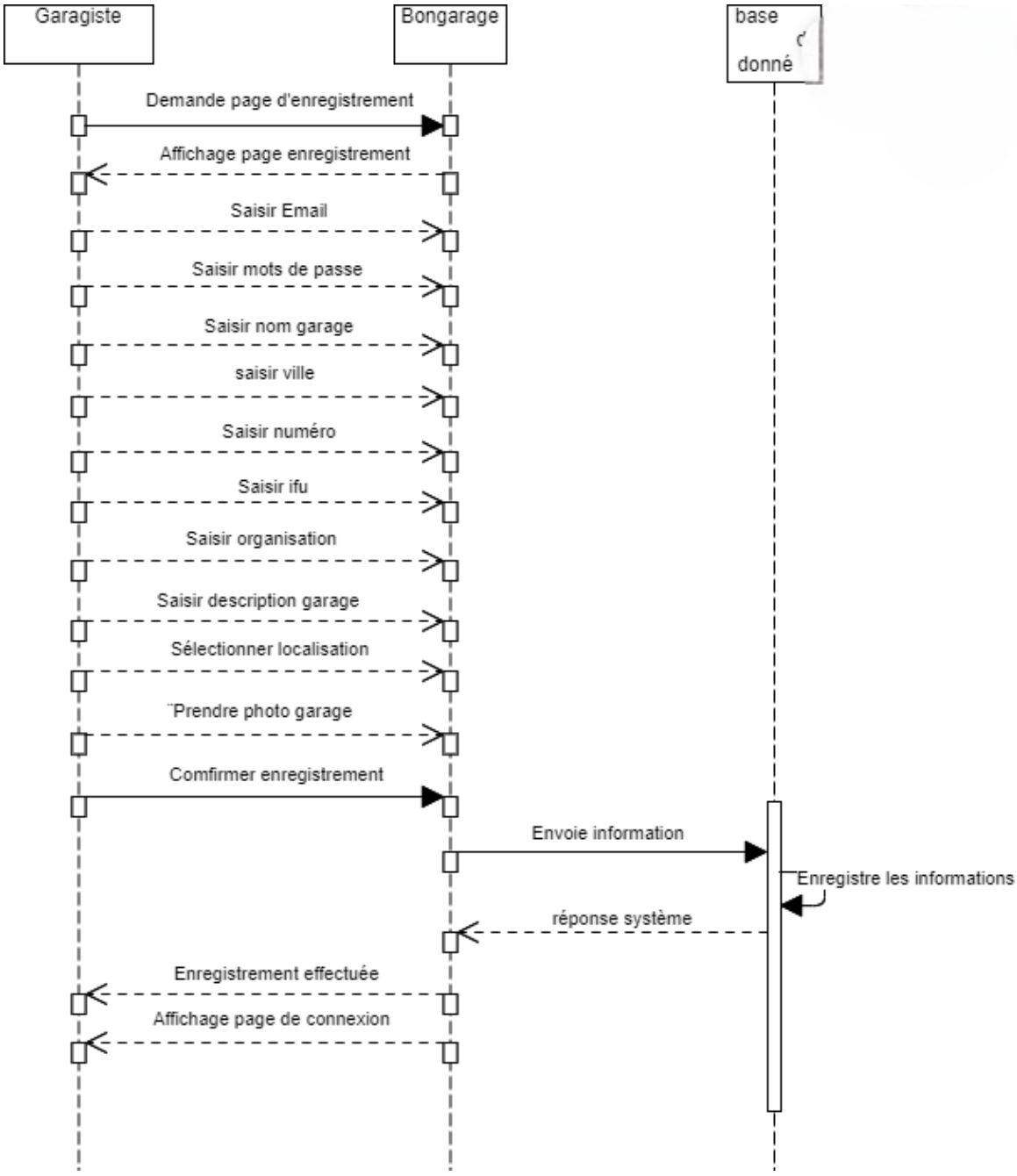



FIGURE 2.5 : Diagramme de séquence du cas d'utilisation : Enregistrement garage

## 2.3 Choix Technique

Pour la réalisation du projet nous avons opté pour les outils techniques tels que :

### 2.3.1 L'achitechture MVC

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont : le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

- Modèle

Le modèle contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité.Le modèle offre des méthodes pour mettre à jour ces données (insertion suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Dans le cas de données importantes, le modèle peut autoriser plusieurs vues partielles des données.

- Vue
  La vue, est l'interface graphique de l'application. C'est via cet élément que vont se faire les interactions entre l'utilisateur et le code métier. Elle affiche aussi les données qu'elle a récupérées auprès du modèle.

- Contrôleur
  Le contrôleur, est l'élément qui contient la logique métier. C'est ici que sont la plupart des algorithmes, calculs, etc. C'est aussi l'intermédiaire principal entre la vue et le modèle. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour.

#### 2.3.1.1 Interaction

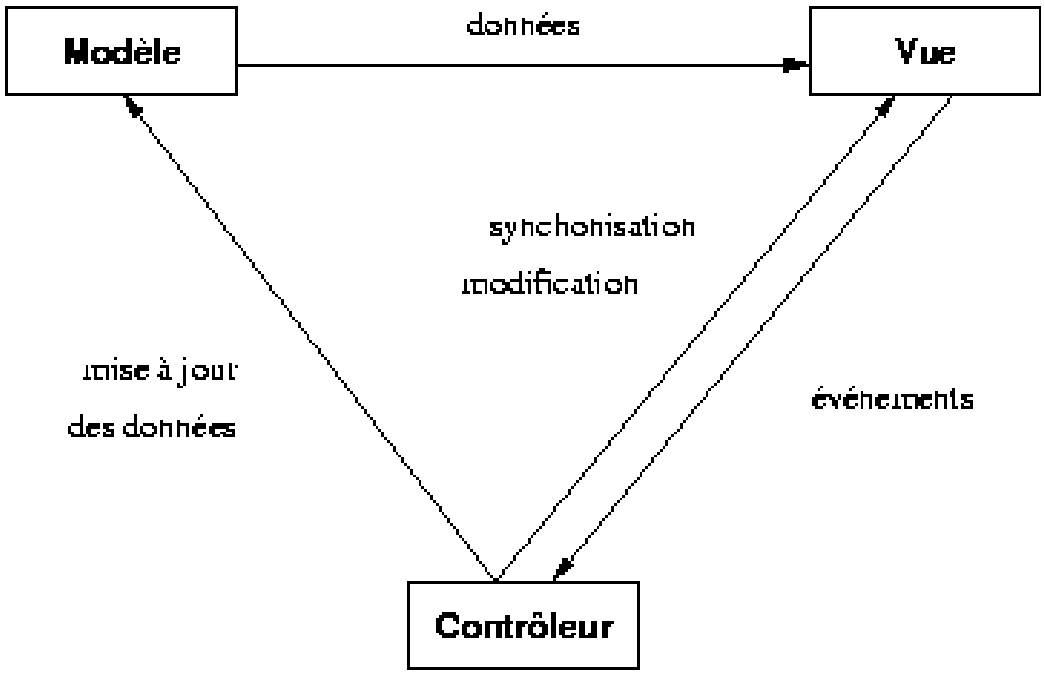



FIGURE 2.6 : Interaction entre le modèle vue et Contrôleur

[3] Source :https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html&ved=2ahUKEwj_hvnn2cCGAxWuD1kFHQcaCGMQFnoECC8QAQ&usg=AOvVaw3QAjwktNLUad976ahFnJJL

Nous avons opté pour cette architecture parcequ'elle offre un cadre pour une bonne structuration de notre application.

### 2.3.2 Front-end

Le Front-end est la partie visible de l'application mobile, celle avec laquelle les utilisateurs interagissent.
**Outils utilisés Pour la réalisation du front-end de l'application :**

#### 2.3.2.1 Figma

Figma est une plateforme collaborative pour éditer des graphiques vectoriels et faire du prototypage. Elle permet de concevoir des designs systèmes pour faciliter la création de sites web et d'applications mobiles. C'est une solution à destination des designers et des développeurs.

**Language de programmation :**

#### 2.3.2.2 Flutter

Fluter est un kit de développement logiciel d'interface utilisateur open-source crée par Google. Il est utilisé pour développer des applications pour Android, IOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code[9, 2]. Nous

### 2.3.3 Back-end

Le Back-end contrairement au front-end est la partie du site ou de l'application que les utilisateurs ne peuvent pas voir.
**Firebase**
Firebase est un outil d'aide au développement d'application web, Android, IOS et unity. Il propose un hébergement en nuage et utilise NOSQL pour héberger des bases de données. Il propose des logiciels utilisés pour le développement d'application mobile pour enregistrer des données, envoyer des notifications et des publicités, remonter les erreurs et les clics effectués sur l'application et est invisible de l'utilisateur de l'application. Elle offre un ensemble de fonctionnalités de sécurité pour aider à protéger les données des applications développées avec sa plateforme notament l'authentification et autorisation, le chiffrement des données en transit et au repos (Firebase utilise le chiffrement SSL/TLS), gestion des utilisateurs et des sessions.[8, 1].

### 2.3.4 Outils et IDE

Au nombre des outils que nous avons utiliser on peut énumérer :
**Android Studio** : Android Studio est l'environnement de développement IDE officiel des applications Android Basé sur le puissant outil de développement et d'édition de code intellij IDEA, Android studio offre encore plus de fonctionnalités qui améliorent la productivité lorsqu'on crée des applications Android.

### 2.3.5 Motivation de choix des technologies utilisées

Ces outils ont été utilisés à cause des différents Avantages qu'ils nous apportent lors du dévelloppement de notre solution

#### Comparaison entre Flutter, React Native et NativeScript

| **Critère** | **Flutter** | **React Native** | **NativeScript** |
|---|---|---|---|
| Langage | Dart | JavaScript | JavaScript/TypeScript |
| Performance | Très bonne (compilation native) | Bonne (bridge natif) | Bonne (bridge natif) |
| UI | Widgets personnalisés | Utilise des composants natifs | Utilise des composants natifs |
| Hot Reload | Oui | Oui | Oui |
| Communauté | En croissance rapide | Très large | Modérée |
| Documentation | Très bonne | Très bonne | Bonne |
| Plugins | Large variété | Large variété | Moins nombreuse que les autres |

TABLE 2.1 : Comparaison entre Flutter, React Native et NativeScript

Flutter offre une performance quasi-native grâce à la compilation en code natif et permet de créer des interfaces utilisateur attrayantes et cohérentes. Sa communauté croissante et son support solide en font un choix robuste pour des applications multiplateformes.

#### Comparaison entre Firebase, AWS Amplify et Backendless

| **Critère** | **Firebase** | **AWS Amplify** | **Backendless** |
|---|---|---|---|
| Base de Données | Realtime Database, Firestore | DynamoDB | Realtime Database, SQL |
| Authentification | Intégré | Intégré | Intégré |
| Fonctions Cloud | Cloud Functions | Lambda | Cloud Code |
| Scalabilité | Très bonne | Très bonne | Bonne |
| Prix | Pay-as-you-go | Pay-as-you-go | Pay-as-you-go |
| Facilité d'Intégration | Très facile | Modérément facile | Facile |
| Outils Analytiques | Firebase Analytics | AWS Pinpoint | Intégré |

TABLE 2.2 : Comparaison entre Firebase, AWS Amplify et Backendless

Firebase propose une suite complète et intégrée de services backend avec une excellente documentation et un support communautaire. Son intégration facile avec Android Studio en fait un choix logique pour le développement mobile.

### Comparaison entre Android Studio, IntelliJ IDEA et Eclipse

| Critère | Android Studio | IntelliJ IDEA | Eclipse |
|---|---|---|---|
| Langage Supporté | Java, Kotlin | Java, Kotlin, autres | Java, C++, PHP, autres |
| Performance | Optimisé pour Android | Très bonne | Modérée |
| Outils de Debugging | Avancés | Avancés | Bonne |
| Interface | Intuitive | Intuitive | Moins intuitive |
| Plugins | Nombreux | Nombreux | Nombreux |
| Intégration Firebase | Très bonne | Bonne | Possible mais complexe |

TABLE 2.3 : Comparaison entre Android Studio, IntelliJ IDEA et Eclipse

En tant qu'IDE officiel pour Android, Android Studio est optimisé pour le développement Android avec des outils puissants de debugging et une intégration native avec Firebase, ce qui le rend idéal pour le développement mobile.

### Comparaison entre Figma, Sketch et Adobe XD

| Critère | Figma | Sketch | Adobe XD |
|---|---|---|---|
| Collaboration | En temps réel | Par fichiers | En temps réel |
| Prototypage | Oui | Oui | Oui |
| Plateforme | Web, Windows, macOS | macOS uniquement | Windows, macOS |
| Coût | Gratuit (basic), payant (pro) | Payant | Gratuit (basic), payant (pro) |
| Plugins | Oui | Oui | Oui |

TABLE 2.4 : Comparaison entre Figma, Sketch et Adobe XD

Figma permet une collaboration en temps réel sur différentes plateformes, ce qui le rend idéal pour les équipes distribuées. Ses fonctionnalités de prototypage et de design sont robustes et bien intégrées.

### Comparaison entre UML et Merise

| Critère | UML (Unified Modeling Language) | Merise |
|---|---|---|
| **Objectif principal** | Modélisation orientée objet | Modélisation des systèmes d'information |
| **Approche** | Orientée objet | Orientée entité-relation |
| **Niveaux de Modélisation** | Conception, spécification, mise en œuvre | Conceptuel, organisationnel, logique, physique |
| **Langage de modélisation** | Utilise des diagrammes standardisés (classes, séquences, etc.) | Utilise des modèles conceptuels, logiques, physiques |
| **Popularité** | Très populaire | Moins populaire |
| **Documentation** | Abondante, internationale | Majoritairement en français |
| **Flexibilité** | Très flexible, utilisé pour divers types de projets | Moins flexible, principalement pour les SI |
| **Complexité** | Peut être complexe à maîtriser | Relativement simple, mais moins puissant |

TABLE 2.5 : Comparaison entre UML et Merise

UML est largement reconnu et adopté au niveau international, ce qui permet une standardisation et une compréhension commune parmi les équipes de développement. Sa flexibilité et sa capacité à modéliser différents aspects du logiciel, depuis la conception jusqu'à la mise en œuvre, en font un outil très puissant pour la modélisation de systèmes complexes. De plus, l'abondance de ressources et de documentations disponibles facilite l'apprentissage et l'utilisation de UML dans divers projets.

## Conclusion

On retient de ce chapitre qu'il est la clé de voûte de notre travail. Il nous a permis de poser les balises de ce que doit être notre application mobile. Il fait la revue de l'arsenal technique et technologique sur lesquelles est basé le développement de notre application et donne une image claire de sa consistance technique.

Chapitre 3

# Résultats et discussion

## Introduction

Après les étapes précédentes, nous avons développé une solution intitulée "Bon garage " qui a pour objectif de répertorier les garages qualifiés pour permettre aux automobilistes de localiser ceux qui sont à proximité d'eux. Dans ce chapitre, nous présentons en détail cette solution dans son fonctionnement tout en exposant ses caractéristiques et les perspectives d'amélioration.

## 3.1 PRESENTATION DE L'APPLICATION

L'application mobile qui permettra la recherche et la localisation des garages automobiles se présente comme suit :

### 3.1.1 Interface d'accueil

Au lancement de l'application, nous avons un splash screen avec le logo de notre application (voir figure 3.1). Après quelques secondes, l'application redirige l'utilisateur vers la page de connexion (voir figure 3.2).

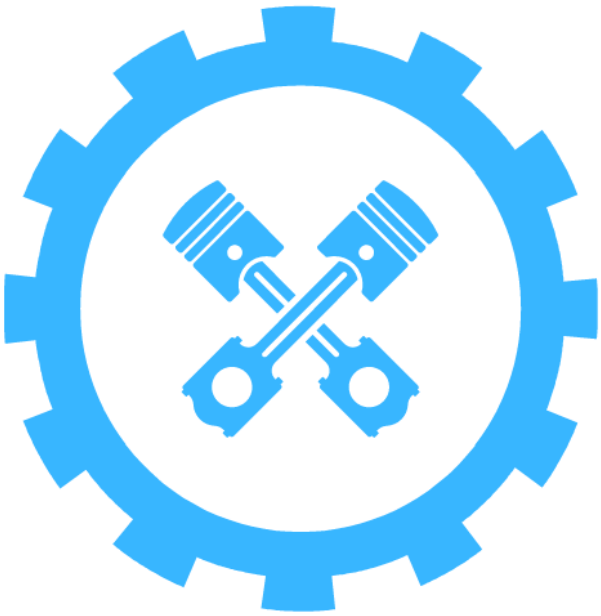


FIGURE 3.1 : logo application

- **AUTOMOBILISTE**

### 3.1.2 Interface de création et de Connexion

#### 3.1.2.1 Page de connexion

Aprés avoir accédé à la page de connexion , l'automobiliste est invité à se connecter en fournissant son email et son mot de passe s'il a déjà un compte. S'il n'a pas de compte l'utilisateur peut créer un compte en appuyant sur le lien "Inscrivez-vous" qui est au niveau de la page de connexion pour accéder au formulaire d'inscription. Pour accéder au formulaire d'inscription, une page de choix du type de compte lui sera présentée, que ce soit un compte automobiliste ou un compte garagiste. Dans ce cas, l'utilisateur devra cliquer sur l'option correspondante.(voir figure 3.2 et 3.3)

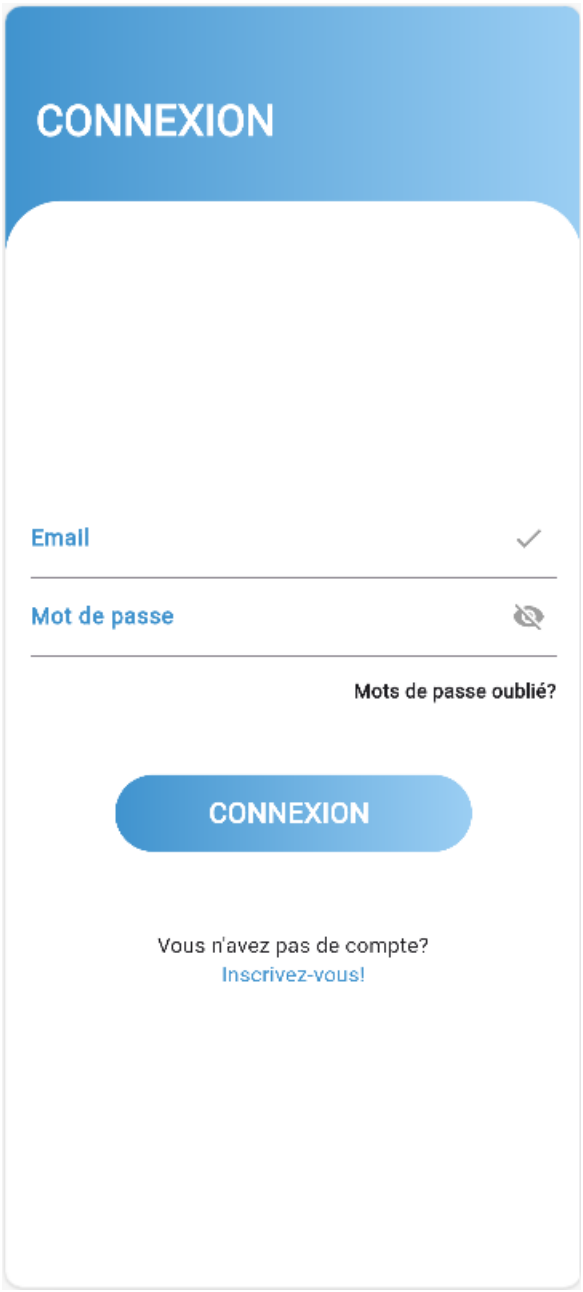



FIGURE 3.2 : page connexion




FIGURE 3.3 : Page choix du type de compte

#### 3.1.2.2 Page d'inscription

Si l'automobiliste n'a pas de compte, cette page lui permet de créer un nouveau compte en fournissant les informations nécessaires telles que son nom, son adresse e-mail et un mot de passe sécurisé qui doit contenir au minimum 6 caractères. Après avoir rempli les champs requis, l'automobiliste peut appuyer sur le bouton "Inscription" alors sont compte sera créé et il sera redirigé vers la page de connexion pour se connecter et accéder à la page de recherche de garages (voir figure 3.4)

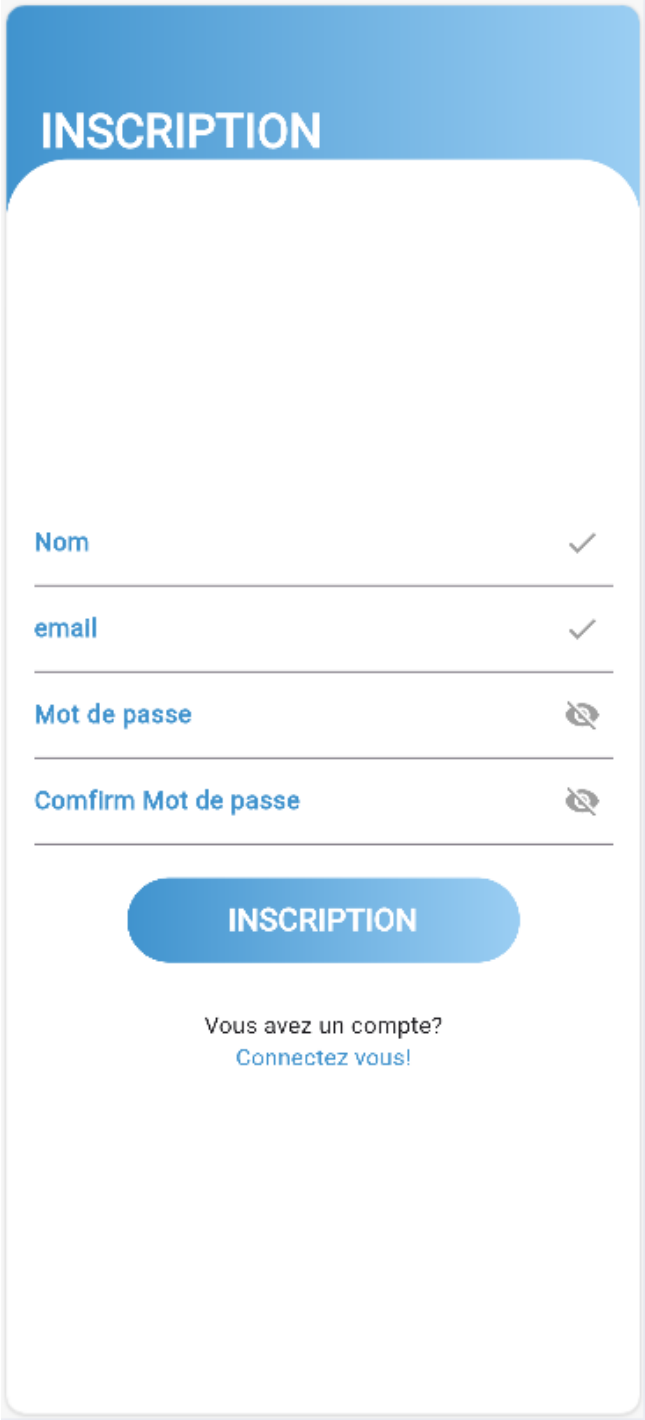



FIGURE 3.4 : page d'inscription

#### 3.1.2.3 Page de Réinitialisation de Mot de Passe

Lorsque l'automobiliste oublie son mot de passe, il pourra le réinitialiser en cliquant sur le lien "Mot de passe oublié" situé en bas du formulaire qui est sur la page de connexion (voir figure 3.2). Une fois sur la page de changement de mot de passe, l'automobiliste devra renseigner son adresse e-mail, puis appuyer sur le bouton "Changer". Un lien de réinitialisation de mot de passe lui sera envoyé par e-mail(voir figure 3.5).

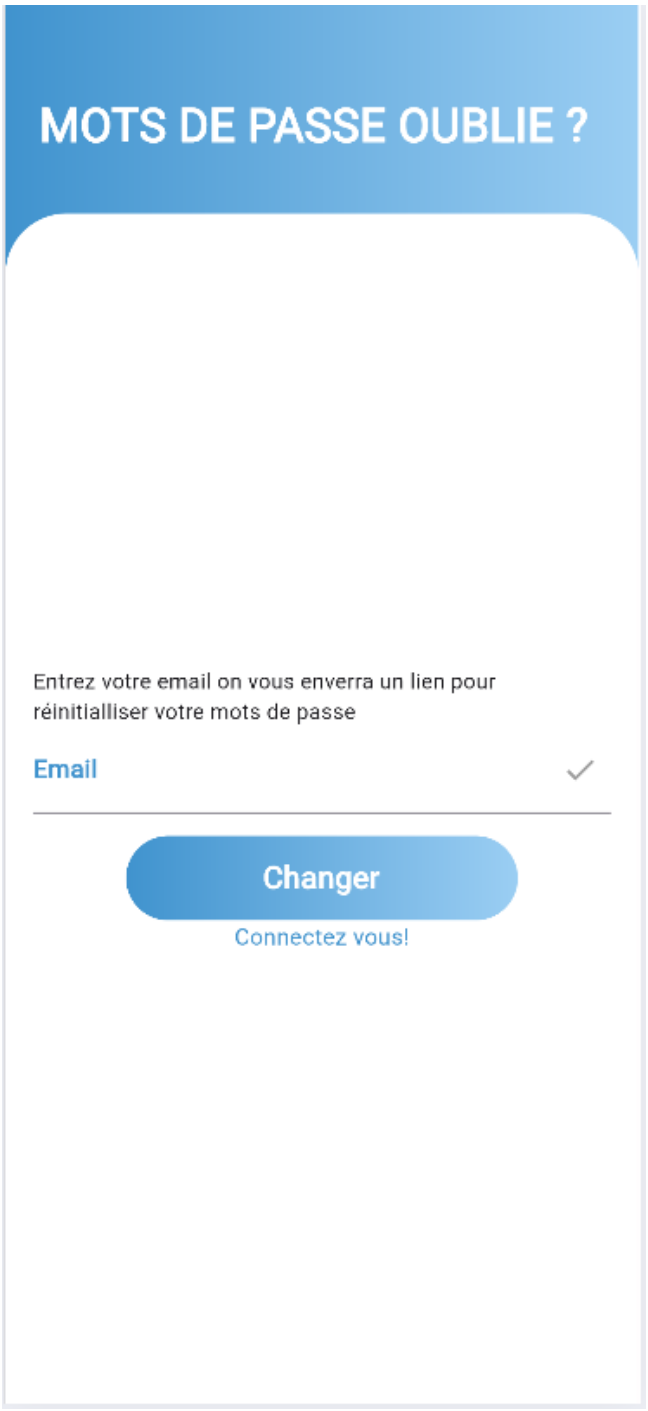



FIGURE 3.5 : page de réinitialisation mots de passe

### 3.1.3 Interface de recherche des garages

Après que l'automobiliste se soit connecté avec succès, il sera automatiquement redirigé vers la page de localisation de garage. Lorsqu'il accède à la page de localisation pour la première fois, l'application lui demandera l'autorisation d'accéder à son emplacement. Il devra cliquer sur "Autoriser" ou "OK" pour permettre à l'outil d'utiliser son emplacement actuel. Sur cette page, l'automobiliste verra une carte interactive.Il pourra naviguer sur la carte en la faisant glisser avec son doigt ou en utilisant les boutons de zoom pour se rapprocher ou s'éloigner. Une fois que l'emplacement de l'automobiliste est pris en compte et affiché sur la carte, il verra automatiquement les garages qui sont à proximité de son emplacement ou non (Les garages sont représentés par le logo en forme de voiture et l'automobiliste

par le logo en forme de personne (voir figure 3.6).

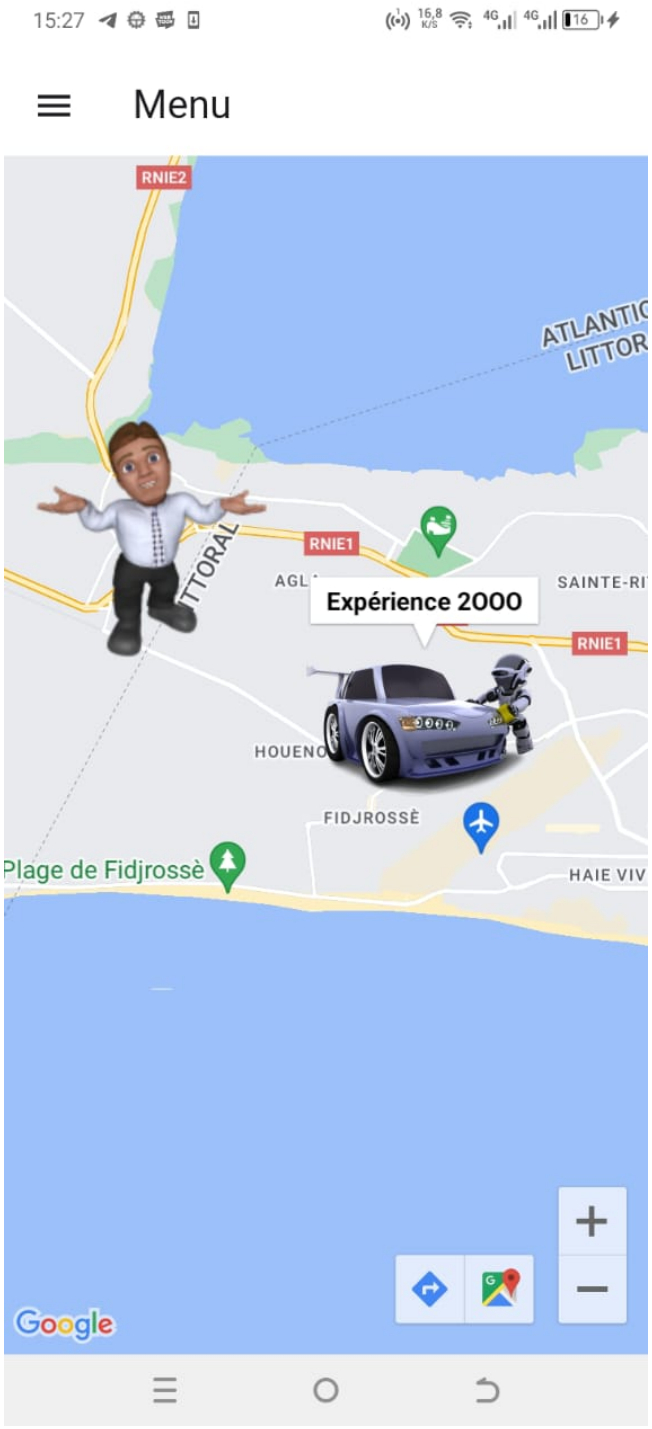



FIGURE 3.6 : page localisation

Sur cette même page, l'automobiliste a la possibilité d'accéder au menu en cliquant sur son icône située en haut à gauche de la page de localisation. Il peut ainsi voir son profil, se déconnecter ou consulter d'autres pages. Pour accéder à la page de profil, il devra cliquer sur le lien "Mon profil". Au niveau de cette page, il pourra consulter ses informations, mettre une photo de profil et modifier ses informations s'il le souhaite, notamment son nom et son adresse e-mail.(voir figure 3.7 et 3.8)

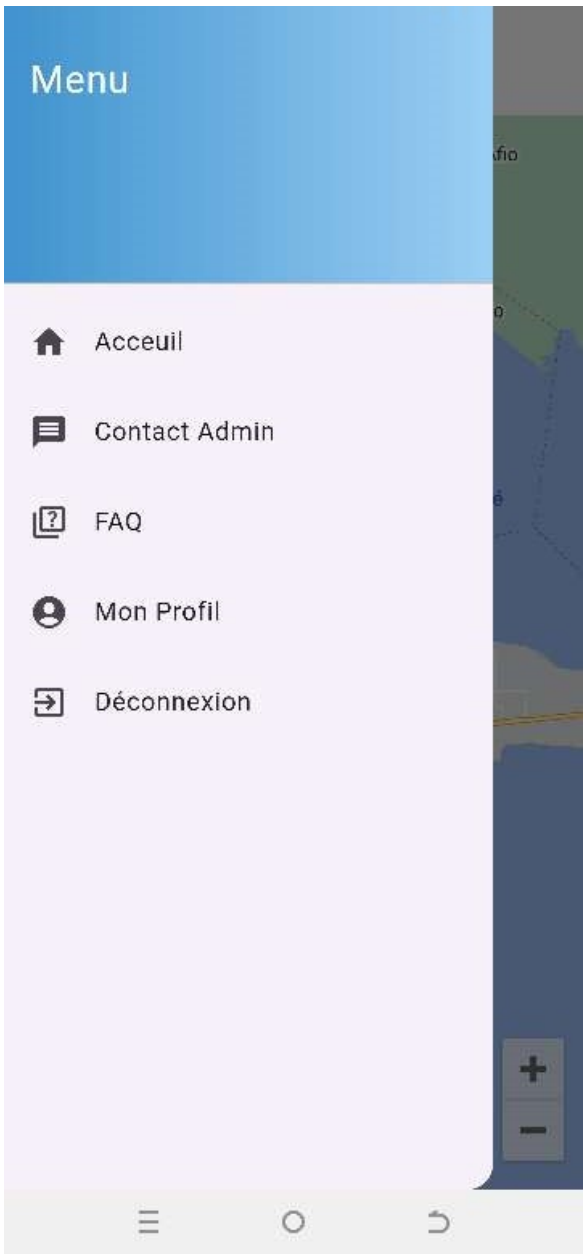



FIGURE 3.7 : Menu

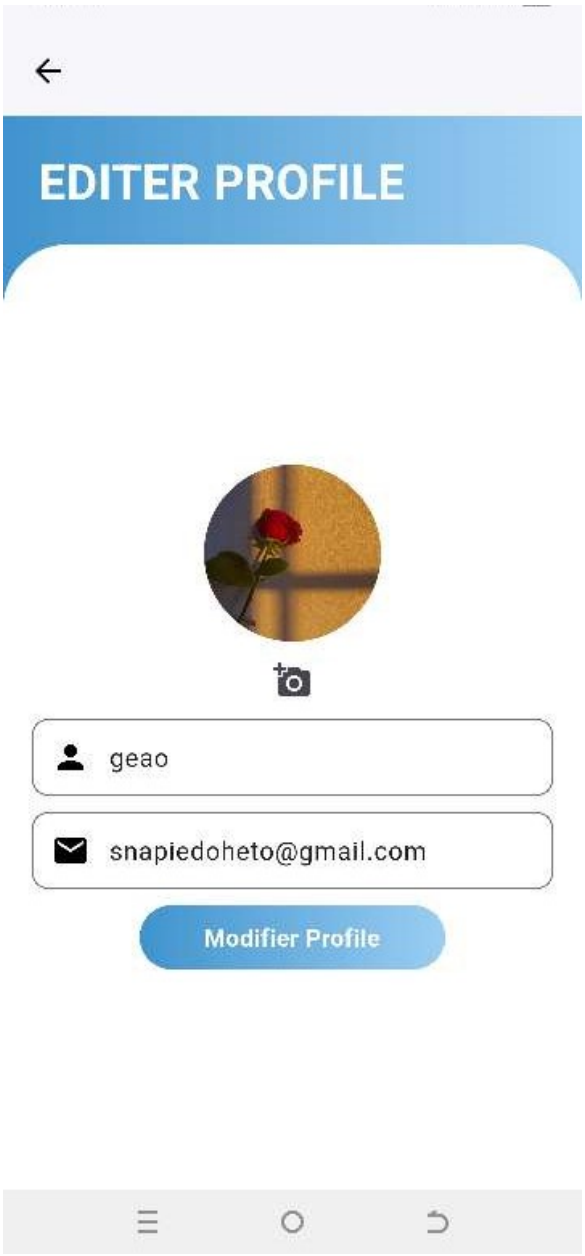



FIGURE 3.8 : Page profile utilisateur

### 3.1.4 Interface de détails sur les garages

Lorsque l'automobiliste a détecté les garages qui sont à proximité après s'être connecté, il a la possibilité de voir les détails de chaque garage,d'obtenir leur numéro de téléphone ou si possible prendre rendez-vous, noter un garage et aussi laisser un commentaire sur le garage. Pour ce faire, il lui suffit de choisir un des garages de son choix et de cliquer dessus. Les informations correspondantes à ce garage lui seront alors renvoyées, telles que le nom du garage, ses services, ses heures d'ouverture ainsi que son numéro de téléphone. Si l'utilisateur désire se rendre dans le garage, le bouton "Itinéraire" lui permettra de s'y rendre en suivant le chemin indiqué par la carte (voir figure 3.9 et 3.10).

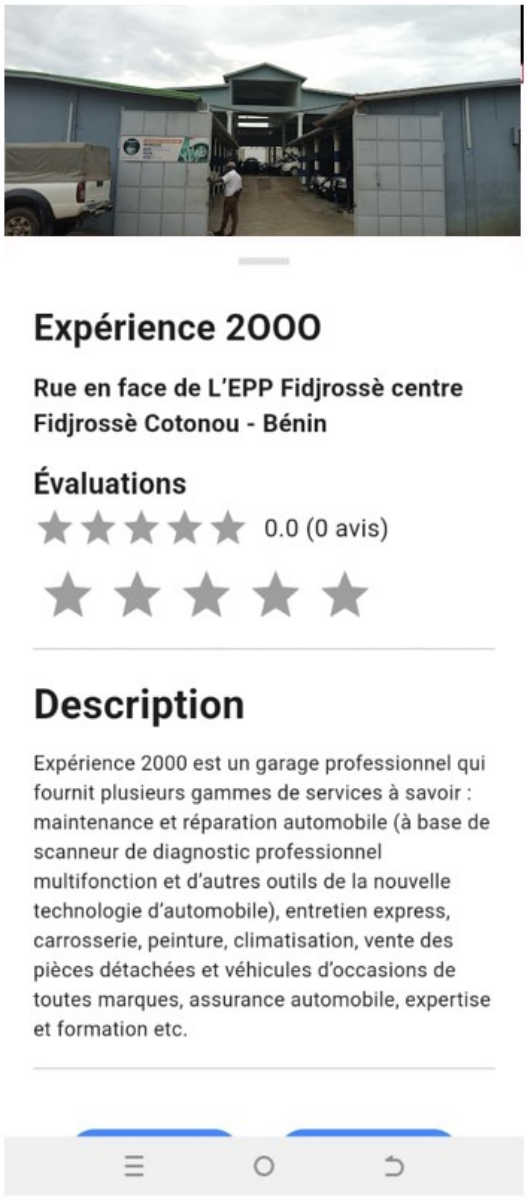



FIGURE 3.9 : page détails garage

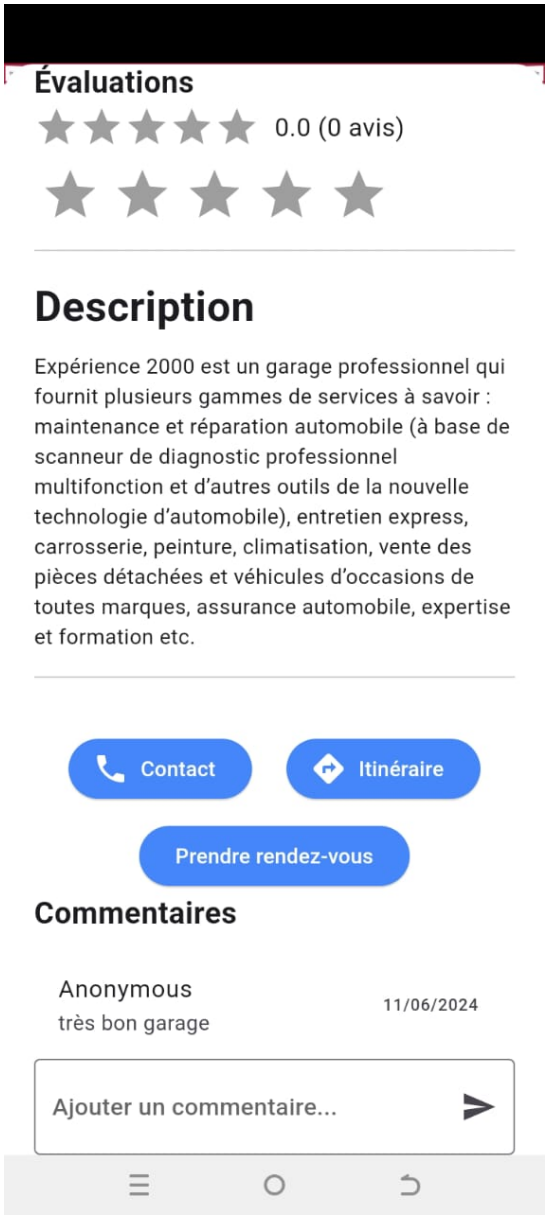



FIGURE 3.10 : page détail garage

### 3.1.5 Interface de prise de rendez-vous

L'automobiliste désirant recourir aux garages répertoriés dans notre application pour effectuer une prestation sur sa voiture, telle qu'une vidange ou toute autre intervention ne nécessitant pas nécessairement l'immobilisation du véhicule, peut prendre rendez-vous chez l'un d'entre eux. Il peut sélectionner la date et l'heure souhaitées, puis cliquer sur le bouton "Confirmer".(voir 3.11)

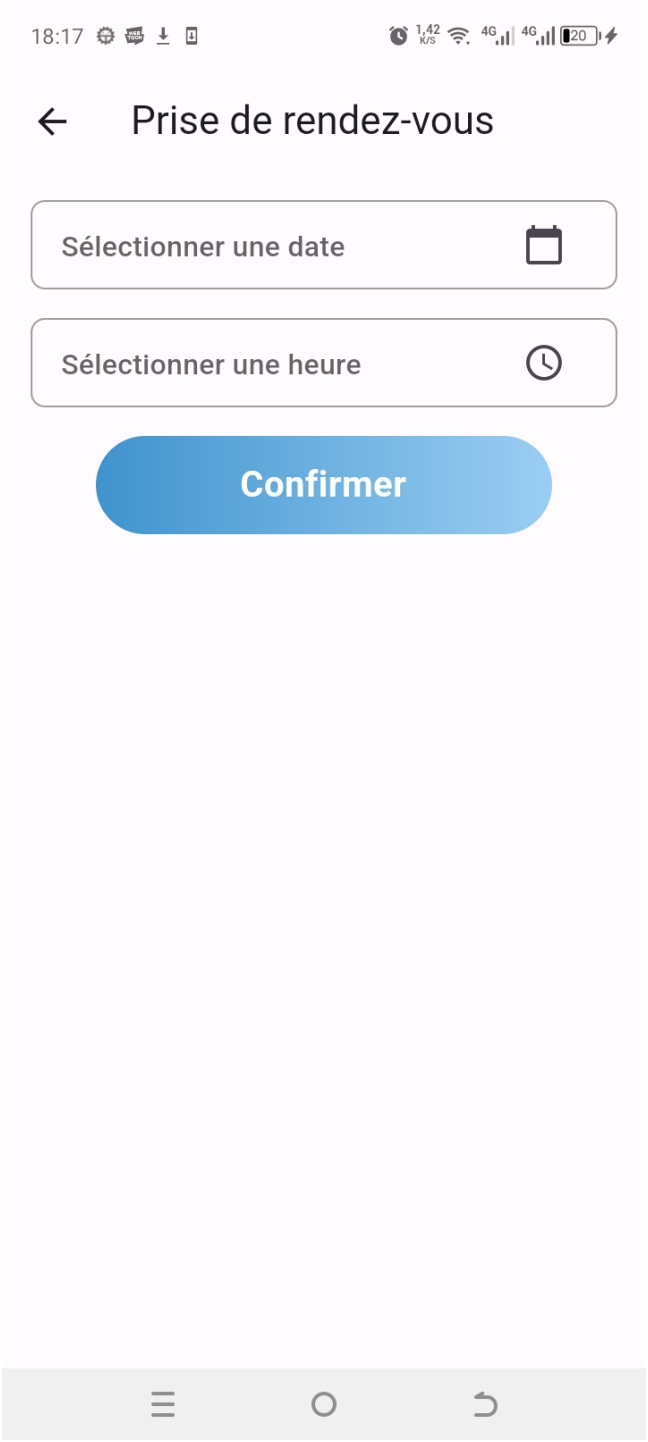



FIGURE 3.11 : page prise de rendez-vous

### 3.1.6 Interface consultation FAQ et contact Admin

#### 3.1.6.1 Page consultation FAQ

cette page a été mise à la disposition des utilisateurs pour permettre de trouver rapidement des réponses à leurs questions courantes sans avoir besoin de contacter le support client. Cela améliore l'expérience utilisateur en réduisant les frictions et en offrant une réponse immédiate à leurs préoccupations. L'utilisateur peut accéder à cette page en cliquant sur l'icône du "Menu". Ensuite, il verra un lien "FAQ" avec une icône de question devant. S'il clique sur le lien, il sera dirigé vers la page de consultation (voir figure 3.12).




FIGURE 3.12 : Page FAQ

#### 3.1.6.2 Page contact Administrateur

Après s'être connecté avec succès, l'automobiliste a la possibilité de contacter l'administrateur pour signaler des problèmes. Pour accéder à la page de contact de l'administrateur, l'utilisateur est invité à cliquer sur l'icône du "Menu". Ensuite, il verra un lien "Contactez l'admin" avec une icône de téléphone devant. S'il clique sur le lien, il sera dirigé vers la page. La figure 3.13 présente la page de contact où l'automobiliste peut saisir son nom, expliquer la raison de son contact avec l'administrateur et fournir un moyen de contact pour recevoir une réponse de l'administrateur.




FIGURE 3.13 : Page contact admin

- **GARAGISTE**

### 3.1.7 Page d'enregistrement garages

Le garagiste doit suivre une procédure d'enregistrement pour que son garage soit répertorié sur l'application. Pour ce faire, il doit cliquer sur le lien "Inscrivez-vous" et choisir l'option "compte garagiste", puis cliquer sur le bouton "S'inscrire" (voir figure 3.3). Ensuite, un formulaire lui sera présenté pour procéder à l'enregistrement de son garage sur l'application (voir figures 3.14 et 3.15).

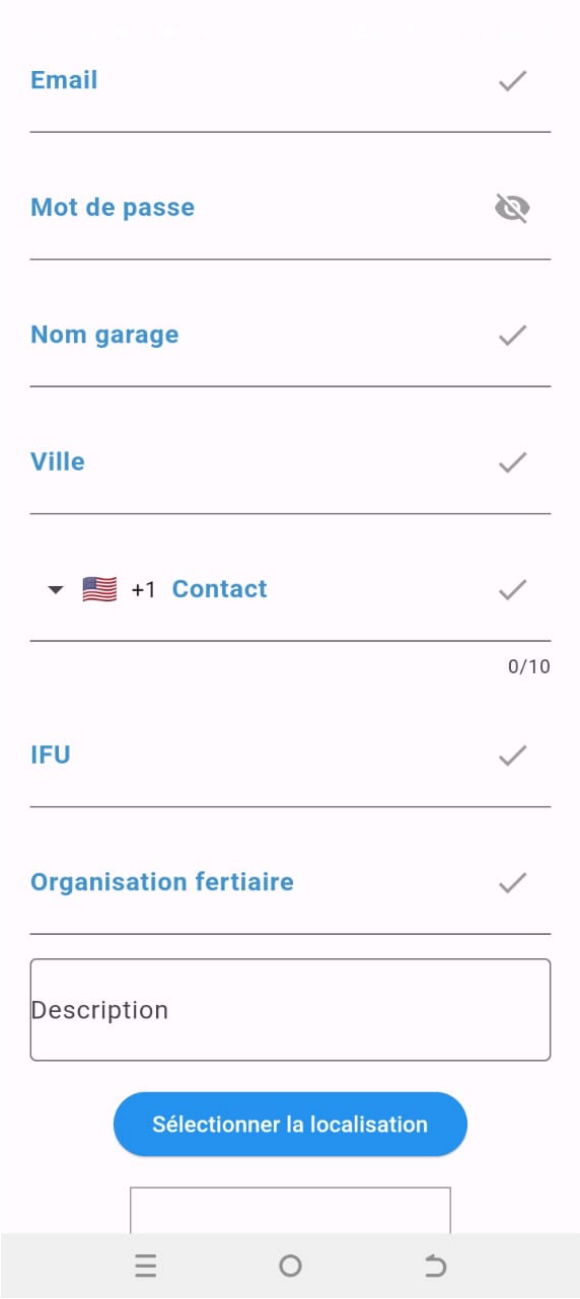



FIGURE 3.14 : page enregistrement garage

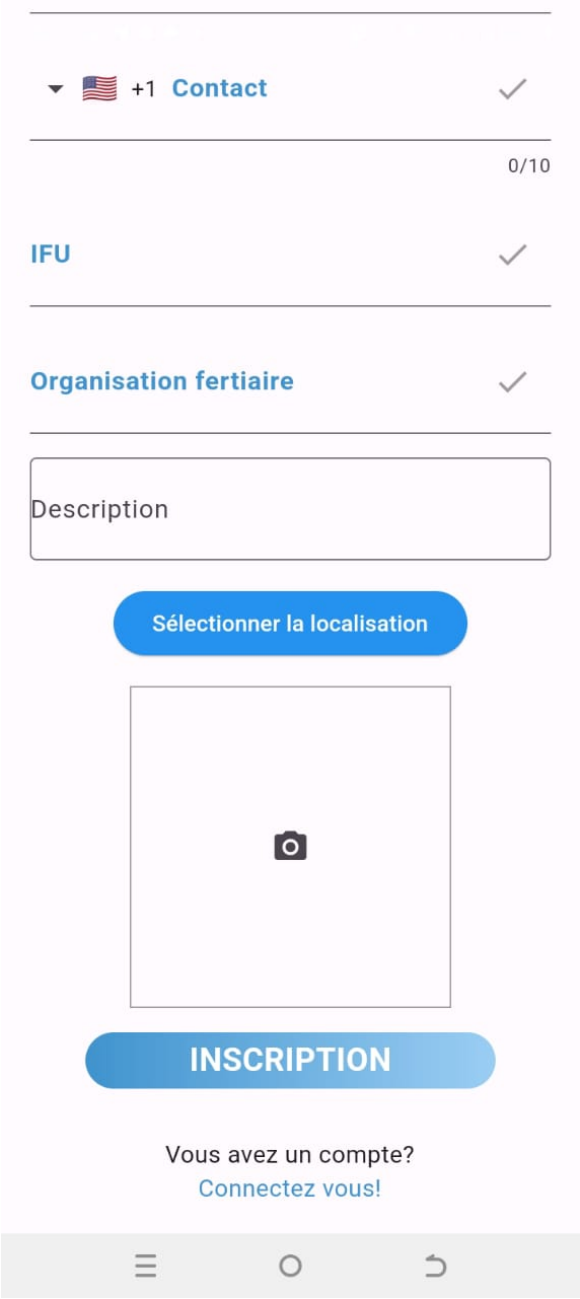



FIGURE 3.15 : page enregistrement garage

### 3.1.8 page de modification information garage et comfirmation de demande de rendez-vous

Le garagiste pourra modifier les informations de son garage, notamment son numéro et sa description (voir figure 3.16). De plus, il pourra consulter la liste des rendez-vous des automobilistes et les confirmer ou non (voir figure 3.17).




FIGURE 3.16 : page modification information

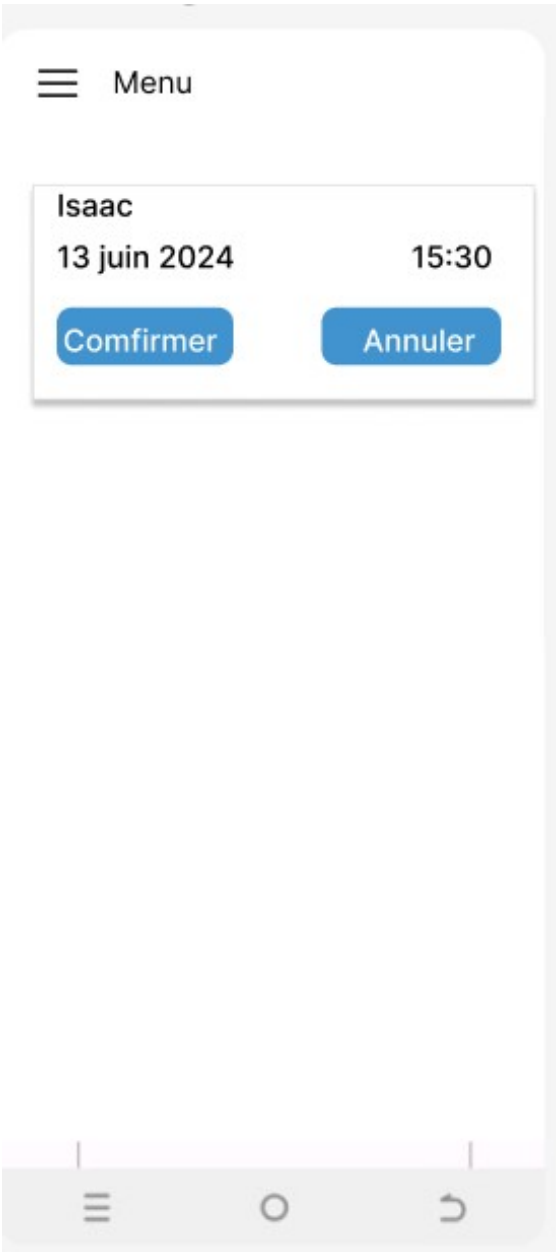



FIGURE 3.17 : page comfirmation rendez-vous

,

- **ADMINISTRATEUR**

Ceci est la partie privé de notre application, là où les opérations concernant les garages s'effectuent. Elle se présente comme suit :
Cette page est la page de connexion de l'administrateur sur celle-ci, il devra renseigner son nom et mot de passe pour accéder a son dashboard pour effectuer les opérations (voir figure 3.18).

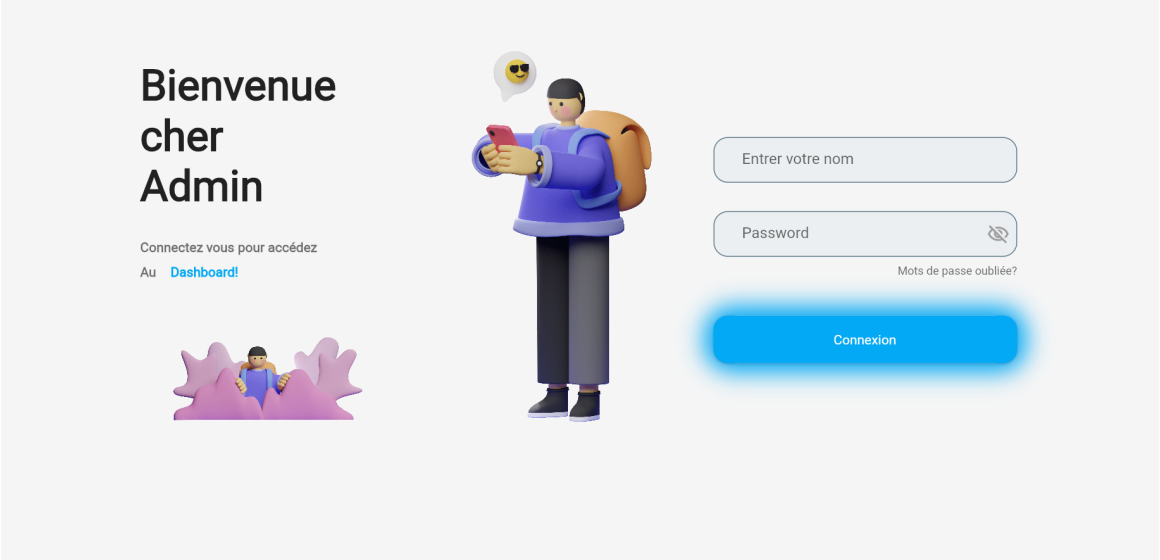



FIGURE 3.18 : Page connexion administrateur

Sur cette interface, il pourra voir la liste des demandes d'ajout des garagistes et le nombre de demande,en plus il pourra ajouter ou supprimer les garages en cliquant sur les boutons correspondants (voir figure 3.19)

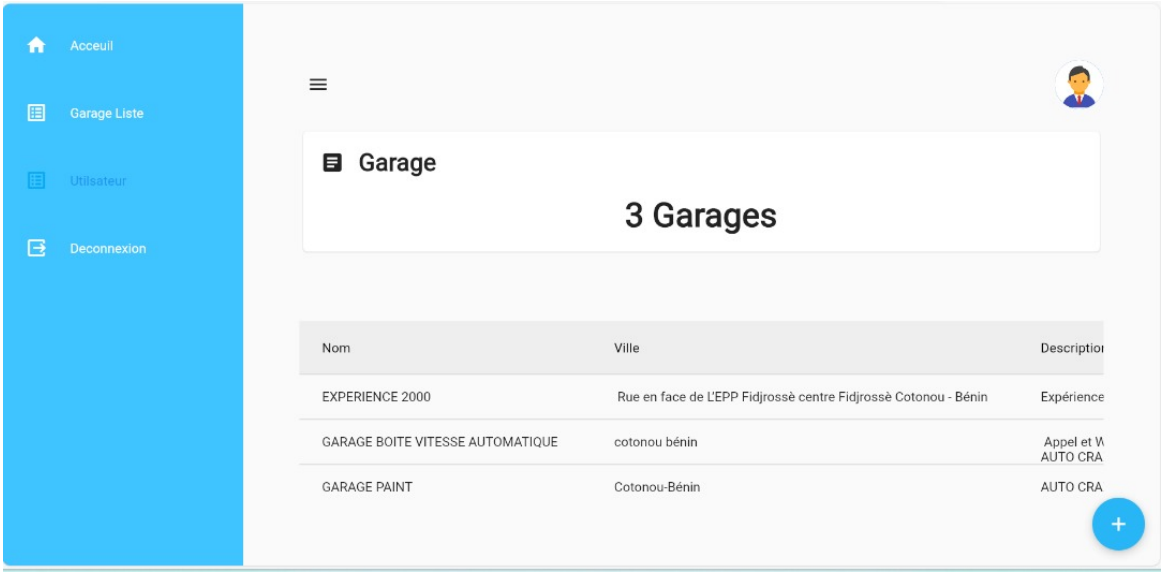



FIGURE 3.19 : Page interface administrateur

Dans cette partie, l'administrateur pourra ajouter un garage en cliquant sur le logo en forme de validation qui est devant le garage, et sur le logo en forme de corbeille pour supprimer le garage.(voir figure 3.20)

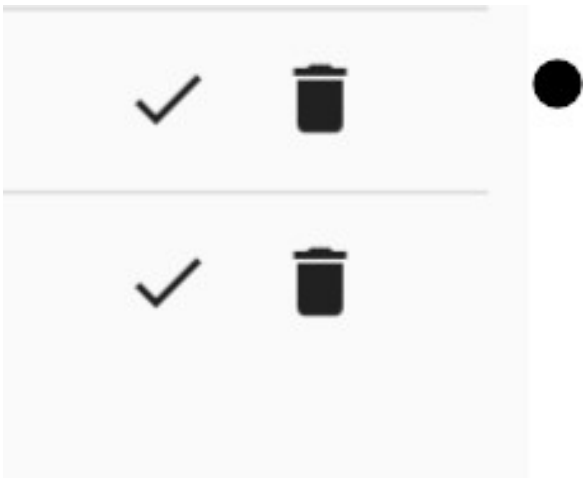


FIGURE 3.20 : Bouton d'ajout et supression garage

sur cette page l'administrateur pourra consulter les messages envoyés par les utilisateurs

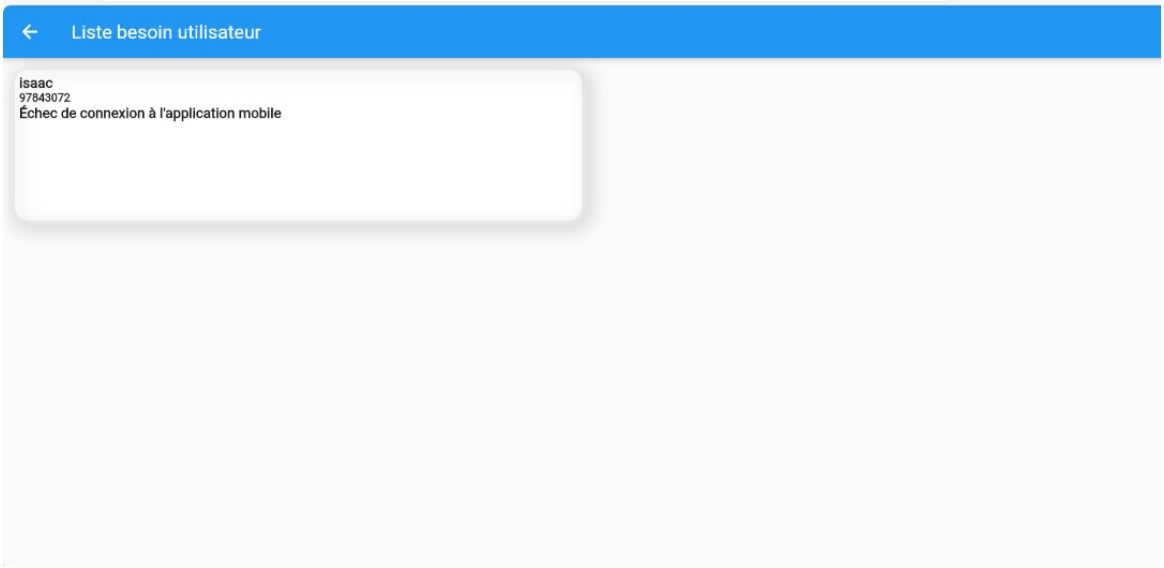



FIGURE 3.21 : Page liste message utilisateur

## 3.2 Discussion

Notre application mobile de recherche de bons garages est un outil mis à la disposition des utilisateurs de véhicules afin de soulager leurs peines dans la recherche et la détection de garages qualifiés pouvant résoudre leurs problèmes. Elle apporte aux automobilistes une solution aux chagrins des pannes mal résolues et sonne le glas des piètres prestations à coûts exorbitants dans certains garages non qualifiés. Aussi, l'application apporte de la visibilité et de la notoriété aux garages sérieux qui gagnent en clientèles et en chiffre d'affaires. Malgré ces avantages, l'application reste à parfaire car assez d'améliorations peuvent y être apportées afin d'upgrader sa maturité notament :

- **Optimisation des performances :** Des efforts supplémentaires peuvent être déployés pour optimiser les performances de l'application, notamment en réduisant les temps de chargement, en optimisant la consommation de mémoire et en améliorant la réactivité de l'interface utilisateur.

- **Expérience utilisateur améliorée :** L'interface utilisateur de l'application pourrait être repensée pour offrir une expérience plus intuitive et conviviale. Cela peut inclure la simplification des flux de travail, l'amélioration de la navigation et la mise en œuvre de meilleures pratiques de conception.

## Conclusion

L'application développée offre un environnement graphique très convivial et simple d'utilisation à n'importe quel utilisateur. Son ergonomie simple et le choix simplifié des différentes options fait d'elle une application facile dans son utilisation pour n'importe quelle couche de la société.

# Conclusion générale

En conclusion, nous avons développé une application mobile destinée aux automobilistes, spécialement conçue pour répertorier et permettre la recherche de garages automobiles. Pour assurer une conception efficace, nous avons réalisé une analyse des besoins, qui nous a conduits à modéliser des diagrammes UML (de séquence, de classe et des cas d'utilisation) afin de représenter visuellement les besoins des utilisateurs et les fonctionnalités de l'application. Nous avons choisi Flutter pour créer une interface intuitive, tandis que Firebase a été intégré pour assurer la sécurité et la gestion des données. Cette application vise à offrir aux propriétaires de véhicules un accès facile à des garages de qualité, tant dans leur région qu'ailleurs. Une fois un garage trouvé, les utilisateurs ont la possibilité de le contacter pour obtenir des informations supplémentaires sur les réparations. Malgré les nombreux avantages qu'offre cette application aux automobilistes, il reste des améliorations à apporter, notamment :

**Automatisation :** automatiser la vérification des informations de chaque garage et l'ajout de nouveaux garages afin de réduire la charge de travail de l'administrateur.

**Améliorations continues :** Poursuivre l'évaluation et l'amélioration de l'application en tenant compte des commentaires et de l'expérience des utilisateurs, afin de la rendre encore plus conviviale et utile dans le secteur des garages automobiles.

# Bibliographie


[1] firebase. https://fr.wikipedia.org/wiki/Firebase.

[2] flutter documentation. https://flutter.dev/development.

[3] interaction entre le modéle,la vue et le contrôleur . https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html&ved=2ahUKEwj_hvnn2cCGAxWuD1kFHQcaCGMQFnoECC8QAQ&usg=AOvVaw3QAjwktNLUad976ahFnJJL.

[4] jobphoning definition application mobile https://jobphoning.com/dictionnaire/application-mobile#.

[5] service d'entretien et de réparation automobile. https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/2069910/service-dentretien-et-de-reparation-automobile.

[6] Techno science. https://www.techno-science.net/glossaire-definition/Atelier-de-reparation-automobile.html.

[7] M. Blaha and J. Rumbaugh. *Modélisation et conception orientées objet avec UML 2*. Pearson Education, 2005.

[8] L. Moroney and L. Moroney. The firebase realtime database. *The Definitive Guide to Firebase : Build Android Apps on Google's Mobile Platform*, pages 51–71, 2017.

[9] M. L. Napoli. *Beginning flutter : a hands on guide to app development*. John Wiley & Sons, 2019.

# Table des matières

**Dédicace** ii

**Remerciements** iii

**Résumé** iv

. . . . . iv

**Abstract** v

. . . . . v

**Liste des figures** vi

**Liste des tableaux** vii

**Liste des acronymes** viii

**Introduction** 1

**1 Revue de littérature** 3

Introduction . . . . . 3

1.1 Définition des concepts clés . . . . . 3

1.2 Présentation des solutions existantes . . . . . 4

1.3 Insuffisances des solutions existantes . . . . . 5

1.4 Intérêt de la solution par rapport aux existantes . . . . . 6

Conclusion . . . . . 6

**2 Méthode de conception et choix techniques** 7

Introduction . . . . . 7

2.1 Analyse . . . . . 7

2.1.1 Methodologie de développement . . . . . 7

2.1.2 Les besoins fonctionnels . . . . . 8

2.1.3 Les besoins non fonctionnels . . . . . 9

2.1.4 Processus métier . . . . . 9

2.2 Méthodes de conception . . . . . 9

2.2.1 Diagramme de cas d'utilisation . . . . . 10

2.2.2 Diagramme de classes . . . . . 10

2.2.3 Diagramme de séquence . . . . . 11

2.3 Choix Technique . . . . . 13

2.3.1 L'achitechture MVC . . . . . 13

2.3.1.1 Interaction . . . 14
2.3.2 Front-end . . . 15
2.3.2.1 Figma . . . 15
2.3.2.2 Flutter . . . 15
2.3.3 Back-end . . . 15
2.3.4 Outils et IDE . . . 15
2.3.5 Motivation de choix des technologies utilisées . . . 16
Conclusion . . . 18

**3 Résultats et discussion** **19**
Introduction . . . 19
3.1 PRESENTATION DE L'APPLICATION . . . 19
3.1.1 Interface d'accueil . . . 19
3.1.2 Interface de création et de Connexion . . . 20
3.1.2.1 Page de connexion . . . 20
3.1.2.2 Page d'inscription . . . 21
3.1.2.3 Page de Réinitialisation de Mot de Passe . . . 22
3.1.3 Interface de recherche des garages . . . 23
3.1.4 Interface de détails sur les garages . . . 25
3.1.5 Interface de prise de rendez-vous . . . 26
3.1.6 Interface consultation FAQ et contact Admin . . . 27
3.1.6.1 Page consultation FAQ . . . 27
3.1.6.2 Page contact Administrateur . . . 28
3.1.7 Page d'enregistrement garages . . . 29
3.1.8 page de modification information garage et comfirmation de demande de rendez-vous . . . 30
3.2 Discussion . . . 33
Conclusion . . . 34

**Conclusion** **35**

**Bibliographie** **36**

**Table des matières** **37**