

# Présentation

Application multiplateforme pour optimiser la gestion des garages. Une solution web et mobile tout- en-un pour les indépendants et petites équipes



**Facilys**



# Sommaire

## Table des matières

Description du projet .....	4
Description .....	4
Objectifs et fonctionnalités principales .....	4
Lien avec la formation .....	5
Compétences acquises mises en pratique.....	5
Concepts théoriques appliqués :.....	6
Motivation .....	6
Raisons du choix de ce projet .....	6
Intérêt personnel/professionnel .....	7
Cas d'utilisation .....	7
Exemples de parcours utilisateurs.....	7
Points complexes et techniques .....	9
Technologies innovantes utilisées.....	10
Fonctionnalités de l'écosystème .....	11
Compétences mises en avant.....	13
Savoir-faire techniques démontrés .....	13
Soft skills développés : .....	14
Conclusion .....	14

# ■ Description du projet

## Description

Le projet consiste en une application multiplateforme permettant aux garages de 1 à 10 personnes d'optimiser la gestion de leurs clients, des véhicules à entretenir, ainsi que la génération de factures, d'ordres de réparation, et l'édition de devis. Une fonctionnalité de gestion d'inventaire est également incluse pour suivre un stock de pièces détachées. Cette application offre un gain de temps significatif pour les petits garages tout en assurant l'historisation des factures et devis. De plus, les factures électroniques respecteront les normes obligatoires établies par l'État français.

## Objectifs et fonctionnalités principales

- **Connexion sécurisée** : Protection des données via une authentification sécurisée et des protocoles de chiffrement.
- **Gestion des clients** : Stockage et gestion des informations clients.
- **Suivi des véhicules** : Enregistrement des détails des véhicules et historique complet des réparations.
- **Devis et facturation** : Création de devis détaillés et génération de factures.
- **Gestion des stocks** : Suivi de l'inventaire des pièces détachées et consommables.
- **Ordres de réparation** : Création et suivi des ordres de réparation.
- **Rapports et analyses** : Génération de statistiques et tableaux de bord sur l'activité.
- **Fonctionnalité offline** : Accès et utilisation de l'application même sans connexion internet.

# ■ Lien avec la formation

## Compétences acquises mises en pratique

Ce projet d'application est directement lié à la formation, car il utilise diverses technologies web ainsi que des solutions cloud pour la gestion des bases de données SQL et NoSQL. L'application sera déployée sur une machine virtuelle scalable. Elle comprend également une application mobile utilisant la technologie PWA, ainsi qu'une version native développée pour Android. De plus, ce projet met en avant les compétences d'un développeur full-stack et d'un développeur DevOps.

**Techniques de programmation appliquées :** Utilisation de langages tels que HTML, CSS, JavaScript, avec des frameworks comme Node.js et Blazor App Server, et C# via ASP.NET Core 8.

**Gestion de bases de données :** Intégration de bases de données SQLite, MariaDB, et Firebase (ou MongoDB), avec modélisation des données et optimisation des requêtes pour de meilleures performances.

**Développement côté serveur :** Utilisation des langages backend Node.js et C# avec Blazor, implémentation d'une API et gestion de l'authentification et des autorisations.

**Méthodologies de développement :** Utilisation d'un diagramme de Gantt et de Trello pour la gestion de projet, avec des pratiques de versioning sous Git et un déploiement continu (CI/CD).

**Sécurité web :** Mise en place de mesures pour protéger contre les vulnérabilités courantes et gestion sécurisée des données utilisateurs.

**Conception d'interface utilisateur :** Application des principes UX/UI, avec un design responsive et une accessibilité optimisée.

**Tests et qualité du code :** Implémentation de tests unitaires et d'intégration, avec l'utilisation d'outils d'analyse de code pour garantir la qualité.

**Déploiement et hébergement :** Choix d'une infrastructure cloud, avec optimisation et configuration du serveur pour un environnement évolutif et scalable.

# Concepts théoriques appliqués :

Cette application étant moderne et utilisant des techniques de programmation orientée objet, les principes suivants seront mis en place :

- **Application des principes SOLID** et des bonnes pratiques de développement.
- **Utilisation des concepts de programmation orientée objet** tels que l'encapsulation, l'héritage, et le polymorphisme.
- **Application des principes de développement web**, y compris le design responsive et l'amélioration progressive (progressive enhancement).
- **GitHub et GitHub Actions** pour la gestion du code et l'automatisation des workflows de CI/CD.

L'objectif est de mettre en pratique les connaissances théoriques acquises durant la formation pour concevoir et développer cette application.

## Motivation

### Raisons du choix de ce projet

J'ai choisi de développer ce projet car mon père possède un garage, et ses confrères ayant des structures similaires utilisent régulièrement des fichiers Excel pour enregistrer les données clients et les factures. Une solution logicielle est particulièrement utile pour mieux gérer les clients, maintenir un historique des réparations des véhicules et des factures, ainsi que se conformer aux normes en matière de facturation dématérialisée. Il y a quatre ans, j'avais développé une application similaire en WinForms avec le langage C# sous le framework .NET 4.7. Aujourd'hui, je souhaite créer une application beaucoup plus moderne en utilisant des techniques et technologies web récentes, afin de mettre en pratique les connaissances acquises et de répondre concrètement aux besoins des garages et à leurs problématiques, étant donné qu'il s'agit d'un cas concret.

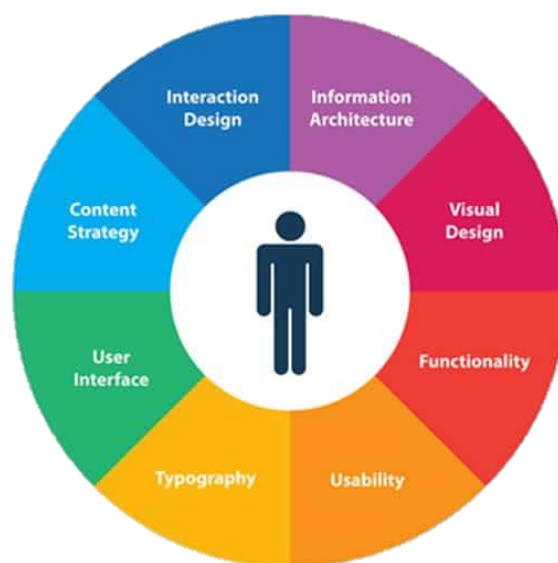
## Intérêt personnel/professionnel

Mon intérêt réside dans l'utilisation de technologies que je n'emploie pas actuellement en entreprise, ainsi que dans la création d'une application fiable destinée aux garages de 1 à 10 personnes. Je souhaite également explorer la possibilité de commercialiser cette application, offrant ainsi une solution efficace et moderne aux professionnels du secteur.

De plus, je suis particulièrement motivé par l'opportunité de proposer des technologies multiplateformes et de démontrer les avantages du cloud computing. Cela est particulièrement pertinent dans le cadre de mon alternance, car cela me permettra de mettre en œuvre ces technologies dans le service informatique où je suis actuellement employé. Mon objectif est d'introduire et d'utiliser ces solutions innovantes au sein de mon équipe pour améliorer les processus existants.

## ■ Cas d'utilisation

### Exemples de parcours utilisateurs



## **Acquisition et Installation du Logiciel**

- Acteur : Gérant du garage
- Objectif : Acheter et installer la solution logicielle
- Contexte : Le gérant souhaite acquérir la solution pour améliorer la gestion de son garage

### **Étapes :**

Le gérant achète la solution via la plateforme de vente en ligne. Après l'achat, il a la possibilité de télécharger les installateurs pour les différents OS tels que : Windows, macOS, et Linux, ainsi que les versions PWA et Android native.

- Il installe les applications sur les différents systèmes d'exploitation.
- Il configure l'application pour l'utiliser dans son garage.

## **Connexion et Utilisation des Modules**

- Acteur : Utilisateur et/ou administrateur du garage
- Objectif : Accéder aux différents modules pour ajouter, consulter des données à la base
- Contexte : L'utilisateur utilise l'application sur un ordinateur ou un téléphone

### **Étapes :**

L'utilisateur se connecte à l'application avec ses identifiants.

- Il accède aux différents modules disponibles (Gestion des clients, Gestion des véhicules, Gestion des factures, etc.).
- Il peut consulter, saisir, supprimer des données.
- Il saisit les informations requises dans les formulaires prévus à cet effet.
- Il enregistre les informations, et celles-ci sont intégrées à la base de données.



# ■ Points complexes et techniques

## Défis techniques à relever

Le développement de cette application présente plusieurs défis techniques, principalement en raison de la diversité des technologies employées. Le projet utilisera plusieurs Framework, notamment Node.js et Blazor :

- **Node.js** sera utilisé pour développer le site web de vente de l'application, avec une intégration de Firebase pour la gestion de l'authentification et des bases de données en temps réel.
- **Blazor** sera employé pour la création de l'application principale, qui devra gérer des bases de données relationnelles comme SQLite pour le mode hors ligne et MariaDB pour le mode en ligne.

Un autre défi important est l'utilisation de **Electron.NET** pour simuler une application de bureau "lourde", permettant une exécution sur plusieurs systèmes d'exploitation tels que Windows, Linux, et MacOS.

L'intégration de **PWA (Progressive Web App)** est également un enjeu crucial pour garantir la compatibilité mobile avec Android et iOS, tout en utilisant Kotlin pour développer une application native Android.

Le principal défi technique réside dans l'harmonisation de toutes ces technologies afin de créer une application multiplateforme stable, performante et fiable. L'objectif est de garantir une expérience utilisateur fluide, que ce soit sur ordinateur, mobile ou tablette, tout en assurant une synchronisation des données entre les modes hors ligne et en ligne.

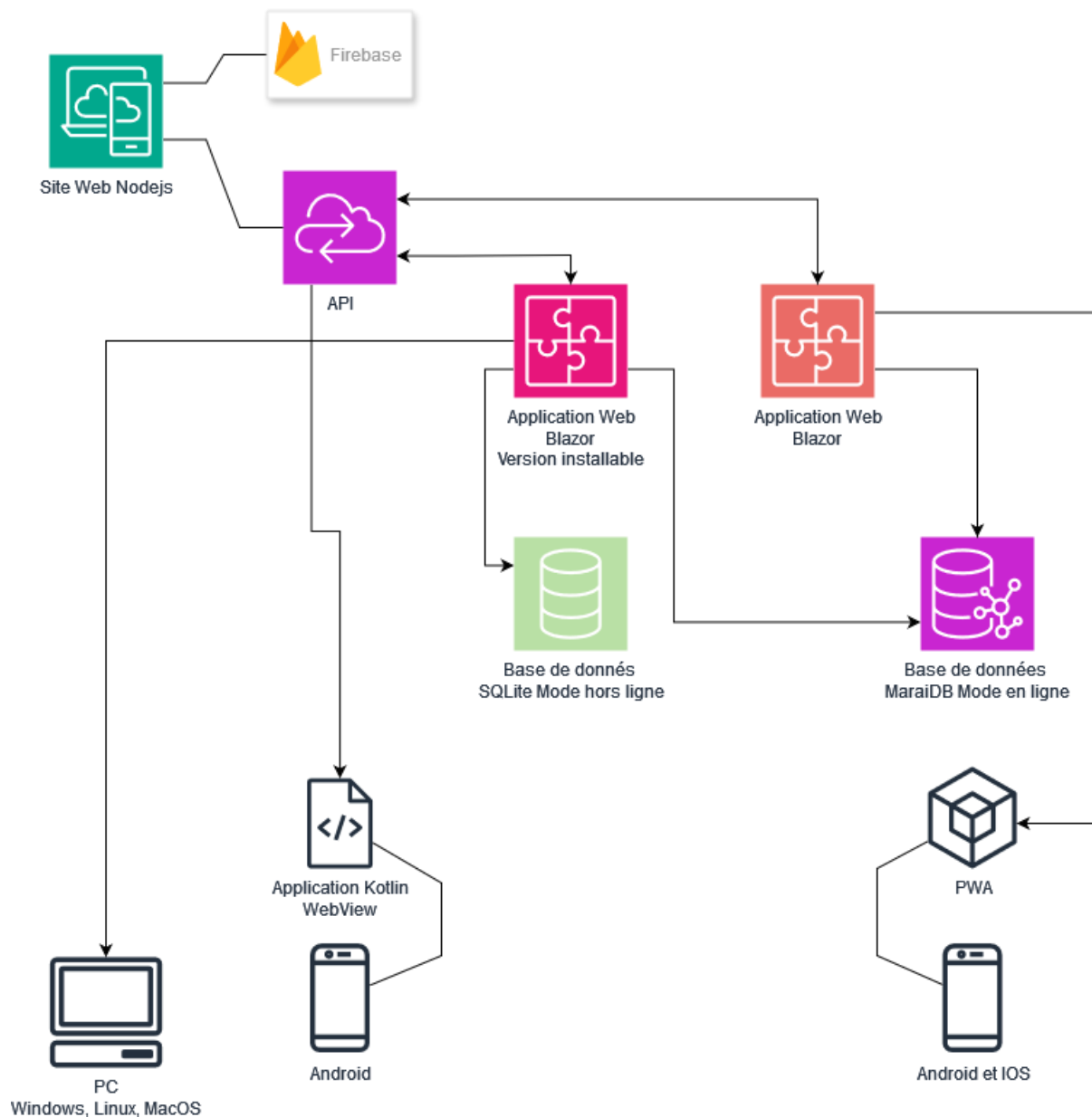


# Technologies innovantes utilisées

Le projet intègre plusieurs technologies innovantes pour assurer une application performante, sécurisée et scalable. Voici les principales technologies utilisées :

- **Node.js (version 18 LTS)** : Utilisé pour la création d'une API backend et la gestion des services web, permettant une communication fluide avec les autres composants.
- **Blazor App Server Net 8** : Employé pour développer l'interface utilisateur de l'application, offrant une expérience web interactive avec des fonctionnalités avancées côté serveur en utilisant le langage C#.
- **Firebase** : Solution cloud utilisée pour l'authentification, le stockage des données en temps réel, et les notifications push, apportant des fonctionnalités modernes et évolutives à l'application.
- **MariaDB et SQLite** : MariaDB sera utilisé pour les fonctionnalités en ligne (base de données relationnelle), tandis que SQLite servira pour le stockage des données en mode hors ligne, permettant à l'application de fonctionner même sans connexion internet.
- **DigitalOcean ou Google Cloud Platform (GCP)** : Une plateformes cloud sera utilisées pour déployer l'application sur une machine virtuelle (VM) scalable ou dans des conteneurs Docker, garantissant une haute disponibilité et une infrastructure performante.
- **PWA (Progressive Web App)** : Technologie utilisée pour rendre l'application accessible sur n'importe quel navigateur, tout en offrant des fonctionnalités d'une application mobile native (hors ligne, notifications).
- **Kotlin** : Utilisé pour le développement de l'application native Android, garantissant une meilleure performance et une compatibilité complète avec les appareils Android.
- **API Node.js** : Une API sera développée en Node.js pour faciliter les échanges de données entre le frontend et le backend de l'application, et permettre l'intégration avec d'autres services externes.
- **Utilisation de GitHub pour CI/CD** : GitHub sera utilisé pour la gestion du code source ainsi que pour la mise en place de pipelines de CI/CD (intégration et déploiement continus), permettant des déploiements automatisés et une meilleure gestion des versions.

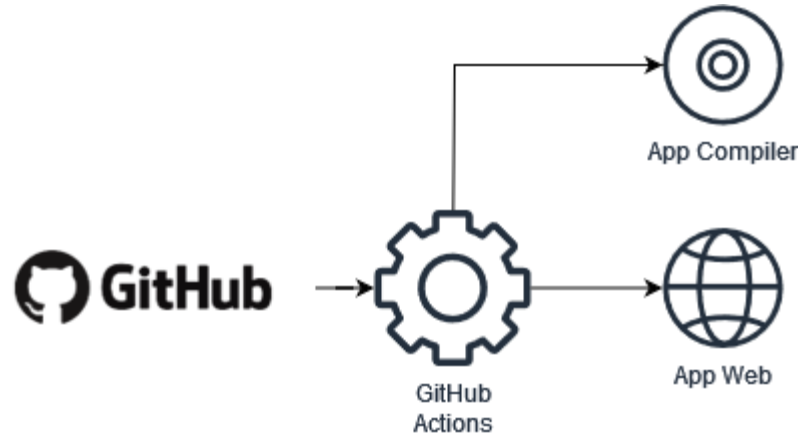
# Fonctionnalités de l'écosystème



L'écosystème du projet repose sur plusieurs outils et services cloud pour garantir une gestion efficace du cycle de développement, du déploiement et de l'hébergement.

**GitHub** : GitHub jouera un rôle central dans le processus de développement. Il sera utilisé pour la gestion du code source, la collaboration en équipe, ainsi que le déploiement des différentes versions de l'application.

**GitHub Actions** : Cette fonctionnalité sera exploitée pour automatiser les pipelines de CI/CD (intégration continue et déploiement continu). Elle permettra de tester, compiler et déployer les différentes applications (web, mobile, PWA) de manière automatisée, garantissant ainsi des mises à jour rapides et fiables.



**Cloud** : Des services cloud seront utilisés pour plusieurs aspects clés de l'application

- **Base de données NoSQL** : Une solution NoSQL sera utilisée, comme Firebase ou MongoDB, pour stocker et synchroniser les données en temps réel.
- **Hébergement de l'application Blazor** : L'application développée avec Blazor sera hébergée sur une infrastructure cloud (DigitalOcean ou Google Cloud Platform), permettant un déploiement scalable et une gestion flexible des ressources.

Cet écosystème technologique vise à optimiser le flux de développement, de déploiement et d'hébergement, tout en assurant la stabilité et la performance de l'application.



# ■ Compétences mises en avant

## Savoir-faire techniques démontrés

Dans le cadre de ce projet, plusieurs compétences techniques clés ont été mises en œuvre.

**Langages de programmation** : Utilisation de langages tels que C#, JavaScript, HTML, et CSS pour le développement de l'application et de ses différentes interfaces.

**Frameworks et bibliothèques** : Intégration de Blazor pour l'application côté serveur et ElectronNet pour créer des versions compilables, ASP.NET Core pour l'architecture backend, ainsi que Entity Framework pour la gestion des bases de données.

**Gestion de bases de données** : Conception et gestion de bases de données relationnelles et non relationnelles, telles que MariaDB, SQLite (pour le mode offline), et Firebase (ou MongoDB) pour les besoins en NoSQL.

**Développement d'API** : Création et consommation d'API RESTful avec Node.js pour permettre la communication entre les différents modules et services.

**Tests et qualité du code** : Mise en place de tests unitaires et d'intégration pour assurer la fiabilité du code, avec une attention particulière sur la qualité et à la maintenabilité.

**Déploiement et hébergement** : Compétences en déploiement sur des infrastructures cloud comme Google Cloud Platform ou DigitalOcean, avec l'utilisation de machines virtuelles scalables et de containers Docker.

**Sécurité** : Intégration de mesures de sécurité, telles que l'authentification sécurisée et la protection des données sensibles, conformément aux normes du RGPD et aux bonnes pratiques du développement sécurisé.

## Soft skills développés :

En tant que développeur unique sur ce projet, plusieurs compétences interpersonnelles et de gestion ont également été sollicitées.

**Gestion de projet** : Gestion complète du cycle de vie du projet, en planifiant et en organisant les différentes phases de développement, en utilisant des outils comme Trello et en appliquant une méthodologie proche de Scrum pour structurer le travail.

**Autonomie et résolution de problèmes** : En travaillant seul, je dois faire preuve de proactivité.

**Communication et documentation** : Bien que ce projet soit personnel, j'ai produit une documentation claire et complète pour faciliter la future maintenance ou commercialisation du produit.

**Créativité** : Je dois faire preuve de créativité en concevant des interfaces utilisateurs intuitives tout en intégrant des fonctionnalités complexes comme la gestion des devis, factures et inventaires de manière fluide et simple d'utilisation pour l'utilisateur final.

## Conclusion

En conclusion, ce projet vise à mettre en pratique une variété de compétences techniques, notamment l'utilisation de technologies modernes comme **Blazor**, **Node.js**, et des solutions **cloud** pour offrir une application multiplateforme. En relevant des défis tels que l'intégration des bases de données **SQL** et **NoSQL**, et le déploiement sur des environnements scalables, ce projet permettra de répondre aux besoins spécifiques des garages indépendants. Il démontrera également ma capacité à gérer l'ensemble du cycle de développement, de l'architecture à la mise en production.