



ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

ET D'ANALYSE DES SYSTÈMES

- ENSIAS -



FILIÈRE GÉNIE LOGICIEL (GL)

Rapport du Projet Plateformes de Développement Plateforme de réservation d'automobiles

RÉALISÉ PAR :

CHQIR SAMIA

MAFTAH EL KASSIMY HAJAR

OUZZINE AYA

ENCADRÉ PAR :

Pr. MAHMOUD

EL HAMLAOUI

Remerciements

Il apparaît opportun de commencer ce rapport par des remerciements à notre professeur Mahmoud EL HAMLAOUI qui nous a donné l'opportunité de réaliser ce projet. Nous lui témoignons toute notre reconnaissance pour ses conseils et également pour nous avoir apporté, au travers de ce projet, un apport bénéfique.

Nous lui témoignons également notre respect le plus profond pour avoir accepté de juger ce présent travail et de l'enrichir par ses remarques et ses critiques.

Nous espérons que ces mots puissent être le gage de toute notre reconnaissance à toutes les personnes qui ont contribué, de près ou de loin, au bon déroulement de la réalisation de ce projet.

Table des matières

Résumé	5
1 Présentation du projet	6
1.1 Analyse de l'existant	6
1.2 Problématique	6
1.3 Solution proposée	7
2 Analyse et conception	8
2.1 Description des cas d'utilisation	8
2.1.1 Identification des acteurs	8
2.1.2 Identification des cas d'utilisation du Client	10
2.1.3 Identification des cas d'utilisation de l'Admin	10
2.2 Besoins non fonctionnels	11
2.3 Diagramme de cas d'utilisation	11
2.4 Diagramme de classes	13
3 Réalisation	14
3.1 Architecture Spring Boot	14
3.2 Tests	15
3.2.1 Test d'insertion d'un utilisateur	15
3.2.2 Test de recherche d'un utilisateur	15
3.2.3 Test d'insertion d'un véhicule	16
3.2.4 Test de création d'une demande	16
3.3 Interfaces de <i>MyRide</i>	17
3.3.1 Page d'accueil	17
3.3.2 Page de connexion	18
3.3.3 Page d'inscription	20
3.3.4 Espace Admin	21
3.3.4.1 Section Accueil admin	21
3.3.4.2 Section Admins	22
3.3.4.3 Section Voitures	24
3.3.4.4 Section Demandes	24
3.3.5 Espace Client	26
3.3.5.1 Section Informations Personnelles	26
3.3.5.2 Section Nouvelle Demande	26
3.3.5.3 Section Demandes	27
3.3.6 Section Nos Offres	27
Conclusion et perspectives	28

Table des figures

2.1	Définition de BCryptPasswordEncoder	9
2.2	Stockage des mots de passe chiffrés dans la base de données	9
2.3	Hachage du mot de passe avant stockage	9
2.4	Hachage du mot de passe lors de l'authentification	10
2.5	Diagramme de cas d'utilisation	12
2.6	Diagramme de classes	13
3.1	Architecture Spring Boot	14
3.2	Création de la table Users à travers le test de création d'utilisateur	15
3.3	Insertion réussie dans la table users	15
3.4	Test de recherche d'un utilisateur par email	15
3.5	Test d'insertion d'un véhicule	16
3.6	Test de création d'une demande	16
3.7	Succès des tests	16
3.8	Slide Page d'accueil 1/3	17
3.9	Slide Page d'accueil 2/3	17
3.10	Slide Page d'accueil 3/3	18
3.11	Bas de page d'accueil	18
3.12	Page de connexion	19
3.13	Champs de connexion requis	19
3.14	Message d'erreur d'authentification	19
3.15	Page d'inscription	20
3.16	Validation du mot de passe	20
3.17	Validation du nom et prénom	21
3.18	Validation du format d'email et de téléphone	21
3.19	Validation de l'acceptation des conditions d'utilisation	21
3.20	Section accueil de l'espace admin	22
3.21	Section des admins de l'espace admin	22
3.22	Ajout d'un nouveau admin à la liste des admins	23
3.23	Modification d'un admin de la liste des admins	23
3.24	Section des voitures de l'espace admin	24
3.25	Ajout et modification d'un véhicule	24
3.26	Section des demandes de l'espace admin	25
3.27	Validation d'une demande par l'admin	25
3.28	Section informations de l'espace client	26
3.29	Section nouvelle demande de l'espace client	26
3.30	Section demandes de l'espace client	27
3.31	Section des offres	27

Résumé

Ce présent travail est effectué dans le cadre de notre formation en tant qu'élèves ingénieurs en 2ème année à l'ENSIAS en filière Génie Logiciel. Nous sommes appelés à concevoir et à réaliser un projet au cours duquel nous seront amenés à exploiter nos connaissances et compétences acquises durant notre formation en Génie Logiciel Objet Avancé et notamment en plates-formes de développement. Le but est de réaliser une application Web fonctionnelle basée sur Spring comme framework pour construire les applications Java.

Après l'accord des membres du groupe, nous avons convenu à choisir de traiter comme sujet du projet **la plateforme de réservation d'automobiles**. En effet, se déplacer en ville peut imposer des contraintes pour un grand nombre de personnes que les professionnels du domaine peuvent régler avec soin. Ces derniers vont également pouvoir répondre aux exigences des clients concernant le type de véhicule demandé, les sièges pour enfants, la disponibilité du WiFi, etc.

Le but de notre plateforme est d'établir le lien entre les clients et les conducteurs professionnels. Le choix d'une formule adaptée au préalable va faire gagner un temps inestimable au client. Les responsables de cette agence de réservation sont supposés avoir de l'expérience, la maîtrise de leur métier et qu'ils sauront mettre en place des chauffeurs compétents.

Chapitre 1

Présentation du projet

1.1 Analyse de l'existant

De nos jours, le nombre de déplacements locaux augmente de plus en plus, où les marocains sont amenés à conduire à moins de 80 km de leur domicile. Il est donc compréhensible que le secteur puisse attirer de nouveaux profils.

Au Maroc, le marché du transport urbain est en croissance mais non encore saturé. On s'est aperçu que des solutions sont déjà existantes sur le marché mais n'offre pas assez de flexibilité pour le choix du véhicule et également pour la personnalisation des critères. Nous avons par la suite décidé de traiter dans ce qui suit du projet **une solution de réservation d'automobiles** offrant les fonctionnalités minimales mais essentielles pour le lancement d'une plateforme pour une agence privée du transport urbain.

Bien qu'il y ait un nombre estimable de déplacements locaux par jour au Maroc, plus de trois quarts des marocains réalisent leur déplacement eux-mêmes, sans passer par une agence spécialisée. Cette tâche est avant tout une affaire de transports publics ou de voiture privée. La raison principale pour ce choix est le prix proposé par les agences de transport qui reste relativement élevé. La clé de compétitivité qui peut nous offrir un **avantage concurrentiel significatif** est de garantir que les frais restent tout de même à la portée de la majorité des individus.

1.2 Problématique

Lancer une nouvelle plateforme du service de transport urbain au Maroc n'est pas sans mésaventure. D'après l'analyse préalable effectué dans la section précédente, déléguer la tâche de transport à une agence spécialisée reste encore, pour plusieurs au Maroc, un pas qui nécessite vigilance et réflexion préalable : Pour la majorité, le souci est plus orienté vers la sécurité et également vers le budget alloué en comparaison aux transports publics qui offrent des tarifs beaucoup moins chers. Par conséquent, nous nous trouvons face à deux problèmes majeurs pour la réalisation de cette plateforme :

- **L'organisation** : Toute personne souhaitant se déplacer en ville doit pouvoir accéder à la plateforme et saisir ses critères. Si le créneau choisi est disponible elle va devoir recevoir une confirmation. De même, le responsable qui reçoit les demandes a besoin de vérifier l'identité du client comme pour n'importe quel service proposé en ligne.

- **Les tarifs :** Les tarifs proposés pour chaque offre de transport sont à titre de simulation **exclusivement**. Ils peuvent être réadaptés en fonction des ressources de l'agence et de leur chiffre d'affaire provisionnel. Nous en tirons le point essentiel qui est d'offrir des prix raisonnables et abordables.

1.3 Solution proposée

En réponse à la problématique soulevée, nous nous sommes aperçus qu'il est important à saisir que les problèmes cités auparavant méritent une profonde réflexion de la part de l'agence qui adoptera cette plateforme. De notre côté, nous nous intéresserons à l'implémentation des **fonctionnalités de base**. Nous avons associé ce but à notre projet pour aboutir une plateforme Web que nous avons nommée "*MyRide*" dont les objectifs sont les suivants :

- Assurer la demande du service de transport : mettre en relation des clients souhaitant opter pour un déplacement avec des professionnels de transport urbain en proposant un certain nombres de critères.
- Assurer l'organisation : établir une plateforme intuitive et ergonomique avec laquelle le client peut interagir avec simplicité sans avoir à parcourir plusieurs pages, seulement avec quelques clics.
- Assurer la personnalisation des demandes effectuées : le client paiera en fonction de ce qu'il a choisi comme critères. Le paiement n'est pas remboursable une fois la demande est validée.
- Assurer un service complet : les tarifs payés comprennent l'arrivée vers le client en plus du transport. Les demandes seront traitées cas par cas par l'admin lors de la vérification de l'identité des clients. Cette dernière a comme but de renforcer la sécurité et éviter les arnaques qui peuvent succomber pendant chaque service proposé en ligne.

Chapitre 2

Analyse et conception

2.1 Description des cas d'utilisation

Dans ce chapitre, nous menons une analyse des besoins pour détailler ensuite le diagramme des cas d'utilisation et le diagramme de classe. La phase d'analyse et spécification des besoins présente une étape primordiale dans le cycle de développement d'un projet afin de mieux comprendre le travail demandé en dégageant les besoins des différents utilisateurs auxquels nous devons répondre. Nous allons alors rechercher à caractériser les fonctionnalités offertes par la plateforme pour satisfaire les besoins de ses utilisateurs.

2.1.1 Identification des acteurs

Afin de procéder à l'identification de toutes les fonctionnalités de notre plateforme, recenser les besoins fonctionnels et appréhender la liste des exigences traduites par les besoins non fonctionnels, nous devons spécifier d'abord les acteurs principaux. Dans notre cas, ces derniers sont les **clients** qui souhaitent opter pour une formule de transport et l'**admin** qui s'occupe de gérer la plateforme. Ce sont les acteurs principaux qui interagissent avec notre plateforme, c'est-à-dire les utilisateurs basiques qui peuvent se servir du système que nous réalisons.

En accédant à la plateforme, on peut consulter une brève description du service de transport urbain proposé, ou également visualiser les différents tarifs disponibles. Par contre, pour effectuer une nouvelle demande, le client doit avant toute chose s'authentifier en remplissant les champs de nom d'utilisateur et de mot de passe avant d'accéder à son espace. L'admin a la possibilité de s'authentifier à travers la même interface mais sera redirigé vers son l'espace admin.

Afin de permettre cette redirection selon le rôle, on a mis une place une table **users** qui contient un champ dédié à ceci, et dans la quelle on stocke les données lors de l'inscription des clients ou lors de l'ajout d'un nouveau admin. Pour le stockage des mots de passe de manière protégée dans notre base de données, on a choisi parmi les mécanismes d'encodage pris en charge par Spring Security **BCrypt**, car c'est généralement la meilleure solution disponible. En effet, la plupart des autres mécanismes, tels que MD5PasswordEncoder et ShaPasswordEncoder utilisent des algorithmes plus faibles et sont désormais obsolètes.

Nous allons commencer par définir notre BCryptPasswordEncoder en tant que bean dans notre configuration :

```

@Bean
public BCryptPasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}

```

FIGURE 2.1 – Définition de BCryptPasswordEncoder

Les implémentations plus anciennes, telles que SHAPasswordEncoder, exigeraient que le client transmette une valeur de sel lors de l'encodage du mot de passe. BCrypt, cependant, générera en interne un sel aléatoire à la place. Ceci est important à comprendre car cela signifie que chaque appel aura un résultat différent, et nous n'avons donc besoin de coder le mot de passe qu'une seule fois. Pour que cette génération de sel aléatoire fonctionne, BCrypt stockera le sel à l'intérieur de la valeur de hachage elle-même. Par exemple, dans la valeur de hachage suivante, extraite de notre base de données :

password
\$2a\$10\$PwjV6bj... .../V866TXYcP87X/9tcgd.i...

FIGURE 2.2 – Stockage des mots de passe chiffrés dans la base de données

Il y a trois champs séparés par \$:

- Le « 2a » représente la version de l'algorithme BCrypt
- Le « 10 » représente la force de l'algorithme
- Le "PwjV6bj.../V866TXYcP87X/9tcgd.i..." est en fait le sel généré aléatoirement.

Fondamentalement, les 22 premiers caractères sont du sel. La partie restante du dernier champ est la version hachée réelle du texte brut

Nous allons maintenant utiliser le PasswordEncoder dans notre processus d'inscription pour hacher le mot de passe avant de le stocker dans la base de données :

```

@PostMapping("/process_signup")
public String processSignUp(User user) {
    User existing = repo.findByEmail(user.getEmail());
    if (existing != null){
        return "redirect:/inscription?error";
    }
    else {
        BCryptPasswordEncoder encoder= new BCryptPasswordEncoder();
        String encodedPassword=encoder.encode(user.getPassword());
        user.setPassword(encodedPassword);
        repo.save(user);
        return "redirect:/connexion?success";
    }
}

```

FIGURE 2.3 – Hachage du mot de passe avant stockage

Traitons maintenant l'autre moitié de ce processus et codons le mot de passe lorsque l'utilisateur s'authentifie. Tout d'abord, nous devons injecter le bean d'encodeur de mot de passe que nous avons défini précédemment dans notre fournisseur d'authentification. Et enfin, nous devons référencer ce fournisseur d'authentification dans notre configuration Java de sécurité :

```

@Bean
public DaoAuthenticationProvider authenticationProvider() {
    DaoAuthenticationProvider authProvider = new DaoAuthenticationProvider();
    authProvider.setUserDetailsService(userDetailsService());
    authProvider.setPasswordEncoder(passwordEncoder());

    return authProvider;
}

@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.authenticationProvider(authenticationProvider());
}

```

FIGURE 2.4 – Hachage du mot de passe lors de l’authentification

2.1.2 Identification des cas d’utilisation du Client

Après avoir identifié le **client** acteur principal, il faut procéder maintenant aux différents cas d’utilisation. Ainsi, pour identifier les cas d’utilisation, il faut se placer du point de vue de notre utilisateur et déterminer comment et surtout pourquoi il se sert de la plateforme. Pour un client, il peut se servir de la plateforme pour :

- Consulter les offres de transport disponibles : Chaque offre propose un certain type de véhicules.
- Contacter les responsables de l’agence : Ces détails sont présents dans le bas de chaque page de la plateforme et ne nécessitent pas une authentification préalable du client.
- Effectuer une estimation de prix : C’est jugé nécessaire que le client puisse avoir une idée sur le prix avant de valider sa demande. Ce cas d’utilisation nécessite que le client soit déjà authentifié par son compte qu’il peut créer sur l’interface dédiée à l’inscription.
- Effectuer une nouvelle demande : Pour cela, il suffira de renseigner les champs indiqués pour le type de véhicule, la date, les services personnalisés et être authentifié sur l’espace des clients.
- Consulter son historique : Afin de pouvoir suivre les demandes effectuées.

2.1.3 Identification des cas d’utilisation de l’Admin

On va faire de même pour l’**admin**. Chaque cas d’utilisation correspond à une fonction métier du système, selon le point de vue de son acteur. Dans tout ce qui suit, l’admin de la plateforme doit être authentifié avec son email et son mot de passe. Ce cas d’utilisation est **commun** entre les 2 utilisateurs du système. En plus de cela, l’admin doit pouvoir :

- Traiter les demandes : Chaque demande effectuée par le client est affichée dans la section dédiée de l’espace admin avec les détails du client qui vont permettre à l’admin de vérifier l’identité de la personne ayant effectué la demande.
- Contacter les clients : En utilisant ces mêmes informations des clients.

- Ajouter des offres : Pour chaque offre, il suffit de décider le type de véhicule, le prix associé et les autres champs nécessaires.
- Modifier la plateforme : En ajoutant à titre d'exemple un nouveau admin.

Comme cité auparavant, l'**authentification** est un cas d'utilisation commun au client et à l'admin. Tout utilisateur est invité à remplir les champs d'authentification pour vérifier son identité avant de pouvoir effectuer des traitements. Une erreur est levée si les champs renseignés lors de l'authentification ne sont pas corrects, c'est-à-dire ne correspondent pas à celles enregistrées dans la base de données. Une erreur intervient également en cas d'informations manquantes lors de l'authentification, c'est-à-dire lors du non-renseignement d'un champ.

2.2 Besoins non fonctionnels

Avant de présenter le diagramme de cas d'utilisation, il reste important de présenter également les besoins non fonctionnels, comme contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement, et que nous devons respecter pour garantir la performance de notre plateforme. Donc pour fournir un produit performant qui respecte les exigences de l'utilisateur et afin de garantir sa satisfaction, des contraintes doivent être prises en compte tout au long du développement du projet :

- **Ergonomie et souplesse** : La plateforme doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles avec affichage des messages de succès ou d'erreur pour faciliter la navigation.
- **Rapidité** : En termes d'optimisation des traitements pour avoir un temps d'exécution raisonnable tout en répondant aux besoins exprimés.
- **Efficacité** : L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.
- **Maintenabilité et scalabilité** : Le code doit être lisible, compréhensible et structuré afin d'assurer son état évolutif et extensible par rapport aux besoins qui peuvent submerger.

2.3 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobés par la limite du système et des relations entre les acteurs et les cas d'utilisation. Dans notre cas, l'ensemble des cas d'utilisation vont être relatifs à nos deux acteurs principaux, déjà mentionnés.

Pour la modélisation objet, nous avons choisi le langage commun **UML** (Unified Modeling Language) avec l'outil **PlantUML** qui permet de générer des diagrammes à partir de fichiers sources. UML est un langage de modélisation formel et normalisé, qui permet de modéliser informatiquement un ensemble d'éléments d'une partie du monde réel en un ensemble d'entités informatiques, appelées objets. Ce langage de modélisation graphique et textuel est destiné

à comprendre et décrire les besoins pour concevoir des solutions.

Notre plateforme assure pour ses utilisateurs diverses fonctionnalités mises en valeur à travers le diagramme de cas d'utilisation illustré par la figure suivante :

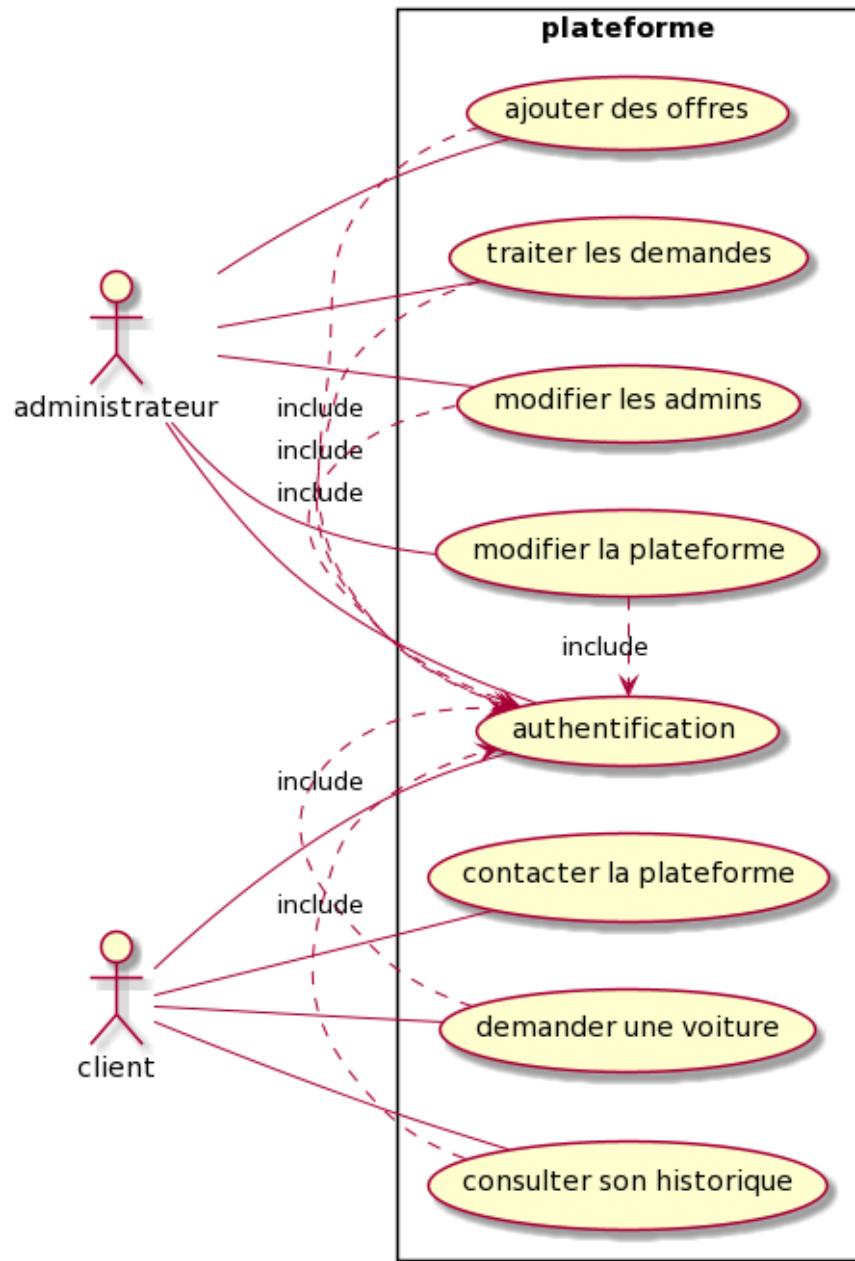


FIGURE 2.5 – Diagramme de cas d'utilisation

2.4 Diagramme de classes

Contrairement au diagramme de cas d'utilisation qui montre le système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système, mais qui va servir de base à l'implémentation, et c'est pour cette raison qu'il faut respecter au maximum une convention de nommage claire, intuitive et compréhensible.

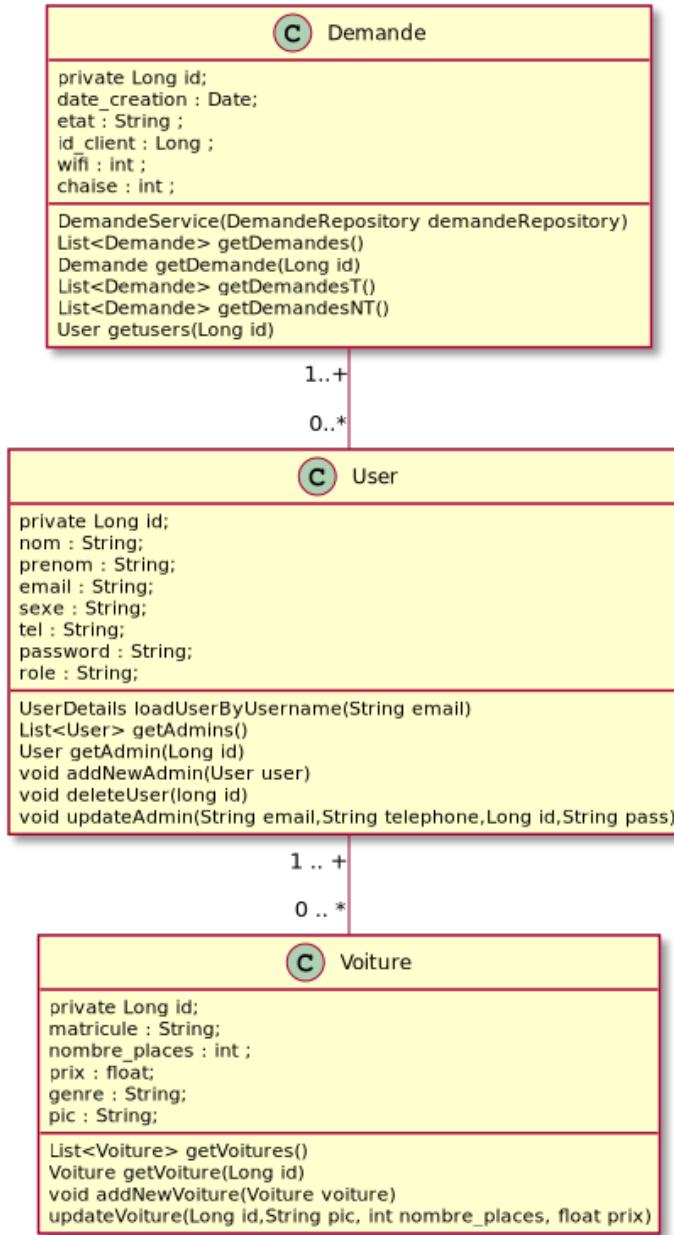


FIGURE 2.6 – Diagramme de classes

Chapitre 3

Réalisation

3.1 Architecture Spring Boot

Spring Boot est un module du Spring Framework. Il est utilisé pour créer des applications Spring autonomes de qualité production avec un minimum d'efforts. Il est développé au-dessus du framework Spring de base. Spring Boot suit une architecture en couches dans laquelle chaque couche communique avec la couche directement en dessous ou au-dessus (structure hiérarchique). Avant de comprendre l'architecture Spring Boot, nous devons connaître les différentes couches et classes qui y sont présentes. Il y a quatre couches dans Spring Boot :

- Couche de présentation : la couche de présentation gère les requêtes HTTP, traduit le paramètre JSON en objet, authentifie la requête et la transfère à la couche métier. En bref, il se compose de vues, c'est-à-dire d'une partie frontale.
- Couche métier : La couche métier gère toute la logique métier. Il se compose de classes de services et utilise des services fournis par des couches d'accès aux données. Il effectue également l'autorisation et la validation.
- Couche de persistance : la couche de persistance contient toute la logique de stockage et traduit les objets métier depuis et vers les lignes de la base de données.
- Couche de base de données : dans la couche de base de données, les opérations CRUD (créer, récupérer, mettre à jour, supprimer) sont effectuées.

Voici une figure qui résume l'architecture utilisée :

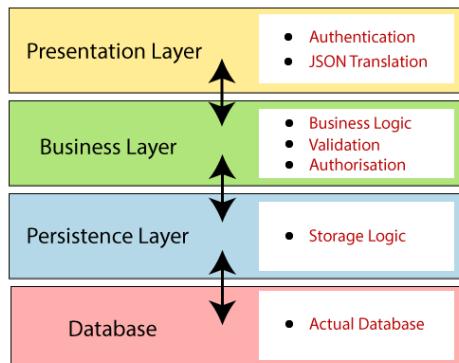


FIGURE 3.1 – Architecture Spring Boot

3.2 Tests

3.2.1 Test d'insertion d'un utilisateur

Après avoir défini l'ensemble des champs nécessaires pour la table "users", on effectue un test d'insertion avec **spring.jpa.hibernate.ddl-auto=create** dans le fichier **application.properties** afin de créer dans un premier temps la table dans notre base de données.

```
1 package com.ride;
2 import static org.assertj.core.api.Assertions.assertThat;
3 import org.junit.jupiter.api.Test;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.boot.test.autoconfigure.jdbc.AutoConfigureTestDatabase;
6 import org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTest;
7 import org.springframework.test.annotation.Rollback;
8 import org.springframework.boot.test.autoconfigure.orm.jpa.TestEntityManager;
9 import org.springframework.boot.test.autoconfigure.orm.jpa.AutoConfigureTestDatabase.Replace;
10 @DataJpaTest
11 @AutoConfigureTestDatabase(replace = Replace.NONE)
12 @Rollback(false)
13 public class UserRepositoryTests {
14     @Autowired
15     private UserRepository repo;
16     @Autowired
17     private TestEntityManager entityManager;
18     @Test
19     public void testCreateUser() {
20         User user=new User();
21         user.setNom("Saidi");
22         user.setPrenom("Said");
23         user.setEmail("said@saidi.com");
24         user.setSexe("H");
25         user.setTel("0675467328");
26         user.setPassword("said");
27         user.setRole("C");
28
29         User savedUser = repo.save(user);
30
31         User existUser = entityManager.find(User.class, savedUser.getId());
32
33         assertThat(existUser.getEmail()).isEqualTo(user.getEmail());
34     }
35 }
36 }
```

Hibernate:

```
drop table if exists users
Hibernate:

create table users (
    id bigint not null auto_increment,
    email varchar(100) not null,
    nom varchar(50) not null,
    password varchar(100) not null,
    prenom varchar(50) not null,
    role varchar(2) not null,
    sexe varchar(10) not null,
    tel varchar(10) not null,
    primary key (id)
) engine=InnoDB
Hibernate:

alter table users
add constraint UK_6dotkott2kjsp8vw4d0m25fb7 unique (email)
```

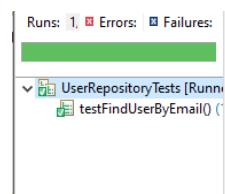
FIGURE 3.2 – Crédation de la table Users à travers le test de création d'utilisateur

	<input type="checkbox"/>	Éditer		Copier		Supprimer	1	said@saidi.com	Saidi	said	
	<input type="checkbox"/>	Tout cocher	Avec la sélection :		Éditer		Copier		Supprimer		Exporter

FIGURE 3.3 – Insertion réussie dans la table users

Une fois cette étape dépassée, on peut rendre **spring.jpa.hibernate.ddl-auto=None** dans le fichier **application.properties** et poursuivre le reste de notre implémentation

3.2.2 Test de recherche d'un utilisateur



```
Runs: 1, Errors: 0, Failures: 0
34     }
35
36     @Test
37     public void testFindUserByEmail() {
38         String email="ayaouzine1@gmail.com";
39         User user=repo.findByEmail(email);
40         assertThat(user).isNotNull();
41     }
42
43 }
```

FIGURE 3.4 – Test de recherche d'un utilisateur par email

3.2.3 Test d'insertion d'un véhicule

```
@DataJpaTest
@AutoConfigureTestDatabase(replace = Replace.NONE)
@Rollback(false)
public class VoitureRepositoryTests {

    @Autowired
    private VoitureRepository repo;
    @Autowired
    private TestEntityManager entityManager;
    @Test
    public void testCreateVoiture() {
        Voiture voiture=new Voiture();
        voiture.setMatricule("P5642");
        voiture.setNombre_places(4);
        voiture.setGenre("mercedes");
        voiture.setPic("https://media.istockphoto.com/photos/red-bus-picture-id171297525?s=612x612");
        voiture.setPrix(1500);
        Voiture savedVoiture = repo.save(voiture);

        Voiture existVoiture = entityManager.find(Voiture.class, savedVoiture.getId());
        assertEquals(existVoiture.getMatricule(),voiture.getMatricule());
    }
}
```

FIGURE 3.5 – Test d'insertion d'un véhicule

3.2.4 Test de création d'une demande

```
@Autowired
private DemandeRepository repo;
@Autowired
private UserRepositoryTests repoTest;
@Autowired
private TestEntityManager entityManager;
@Test
public Long testCreateDemande() {
    Demande demande=new Demande();
    demande.setChaise(0);
    demande.setEtat("NF");
    demande.setId_client(repoTest.testFindUserByEmail().getId());
    demande.setEmail(repoTest.testFindUserByEmail().getEmail());
    demande.setNom(repoTest.testFindUserByEmail().getNom());
    demande.setPrenom(repoTest.testFindUserByEmail().getPrenom());
    demande.setPrix(200);
    demande.setTel(repoTest.testFindUserByEmail().getTel());
    demande.setWifi(0);
    Demande savedDemande = repo.save(demande);

    Demande existDemande = entityManager.find(Demande.class, savedDemande.getId());
    assertEquals(existDemande.getNom(),demande.getNom());
    return demande.getId();
}
```

FIGURE 3.6 – Test de création d'une demande

Le succès des deux tests indiqués auparavant est représenté par la figure suivante :

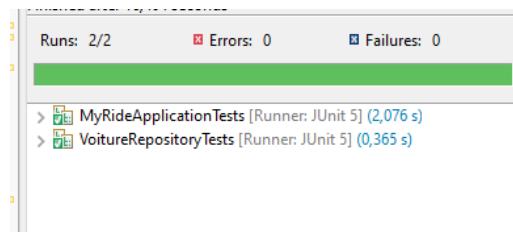


FIGURE 3.7 – Succès des tests

3.3 Interfaces de *MyRide*

3.3.1 Page d'accueil

C'est la première page qui s'affiche à un utilisateur de la plateforme. Elle propose sous forme de slide la présentation générale de la plate-forme. Elle contient ensuite une section réservée à des informations générales à propos du service proposé et une autre section dans le bas de page pour les détails du contact. Ce même bas de page est disponible dans les deux sections d'authentification et d'inscription.

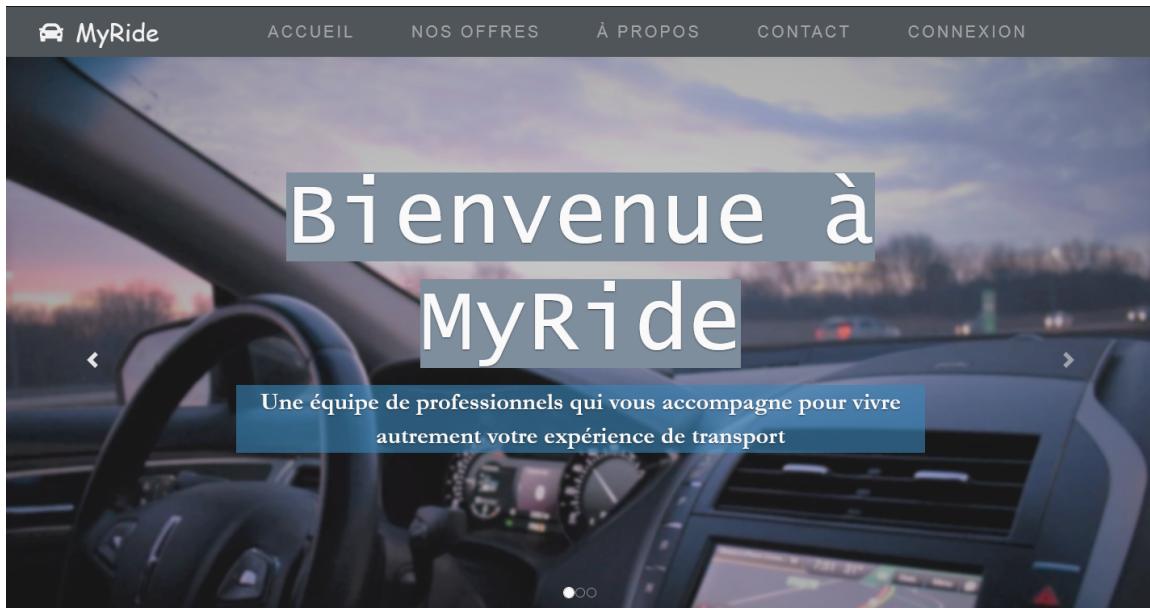


FIGURE 3.8 – Slide Page d'accueil 1/3



FIGURE 3.9 – Slide Page d'accueil 2/3

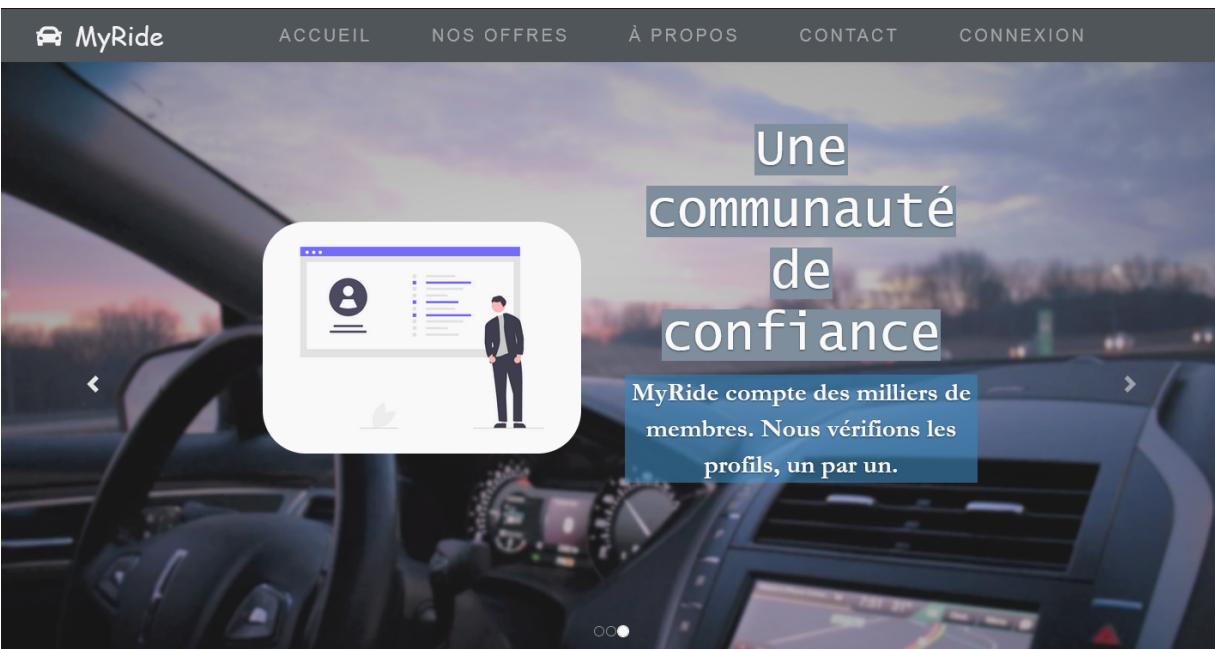


FIGURE 3.10 – Slide Page d'accueil 3/3

The bottom section of the MyRide homepage is divided into several sections. At the top, there are four circular icons with symbols: a checkmark, a dollar sign, a gavel, and a shield. Below each icon is a heading and a brief description. The first section is "Services de qualité" with the subtext "Personnel rigoureux et bien formé. Professionnalisme et sérieux". The second is "Meilleur rapport qualité prix" with the subtext "Offrant ponctualité ainsi que la flexibilité adaptée à vos propres critères". The third is "Tous vos critères sont respectés" with the subtext "Choisissez le transport qui vous convient le mieux. On s'occupe du reste". The fourth is "Sécurité et disponibilité" with the subtext "Tous nos conducteurs ont l'expertise du domaine. Vous traverserez votre trajet en toute sécurité". At the very bottom, there is a "Contact" section with address, phone number, and email information, along with a "Notre but principal est de vous satisfaire" statement.

FIGURE 3.11 – Bas de page d'accueil

3.3.2 Page de connexion

Cette interface permet l'authentification du client et de l'admin où chacun va être redirigé vers sa propre page d'accueil en fonction de son rôle. Évidemment, les deux champs d'authentification sont tous les deux requis.

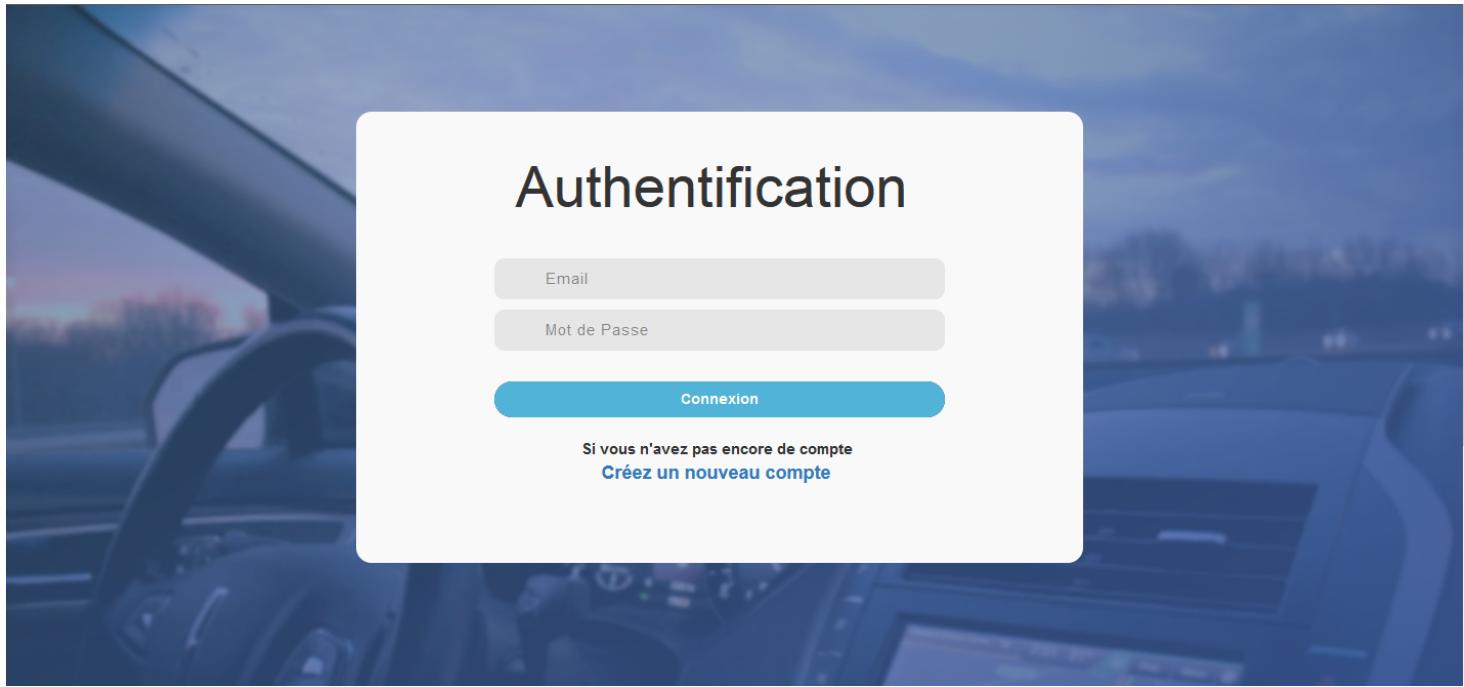


FIGURE 3.12 – Page de connexion

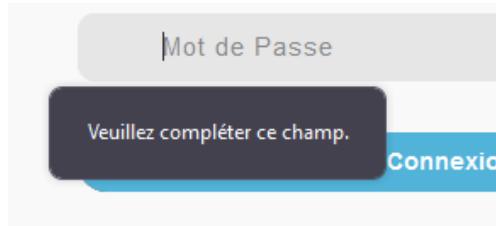


FIGURE 3.13 – Champs de connexion requis

Nous prenons les 2 principaux cas d'erreurs qui peuvent être déclenchés :

- Saisie d'un email qui n'a pas de compte lié dans la plateforme.
- Saisie d'un mot de passe erroné pour un compte déjà existant.

Dans ces deux cas, le message d'erreur suivant s'affiche :

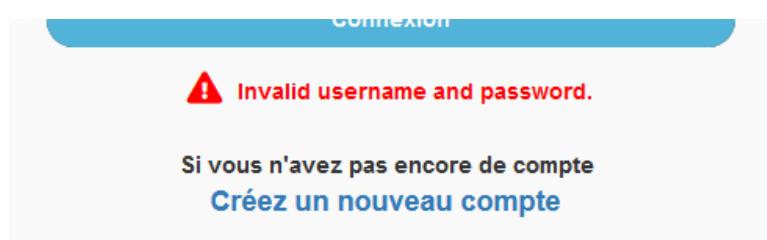


FIGURE 3.14 – Message d'erreur d'authentification

3.3.3 Page d'inscription

Comme dans toutes les plateformes en ligne, le visiteur ne peut profiter des services proposés qu'après la création d'un compte. Pour ce faire, nous proposons à l'utilisateur dans la phase de connexion de créer un compte s'il ne le possède pas déjà. Dans le cas échéant, nous collectons les informations nécessaires, et qui serviront également à l'utilisateur pour ses prochaines connexions. Ici la validation du format de l'email, du téléphone et de la conformité du mot de passe et de sa confirmation se fait instantanément avant clic sur envoyer.

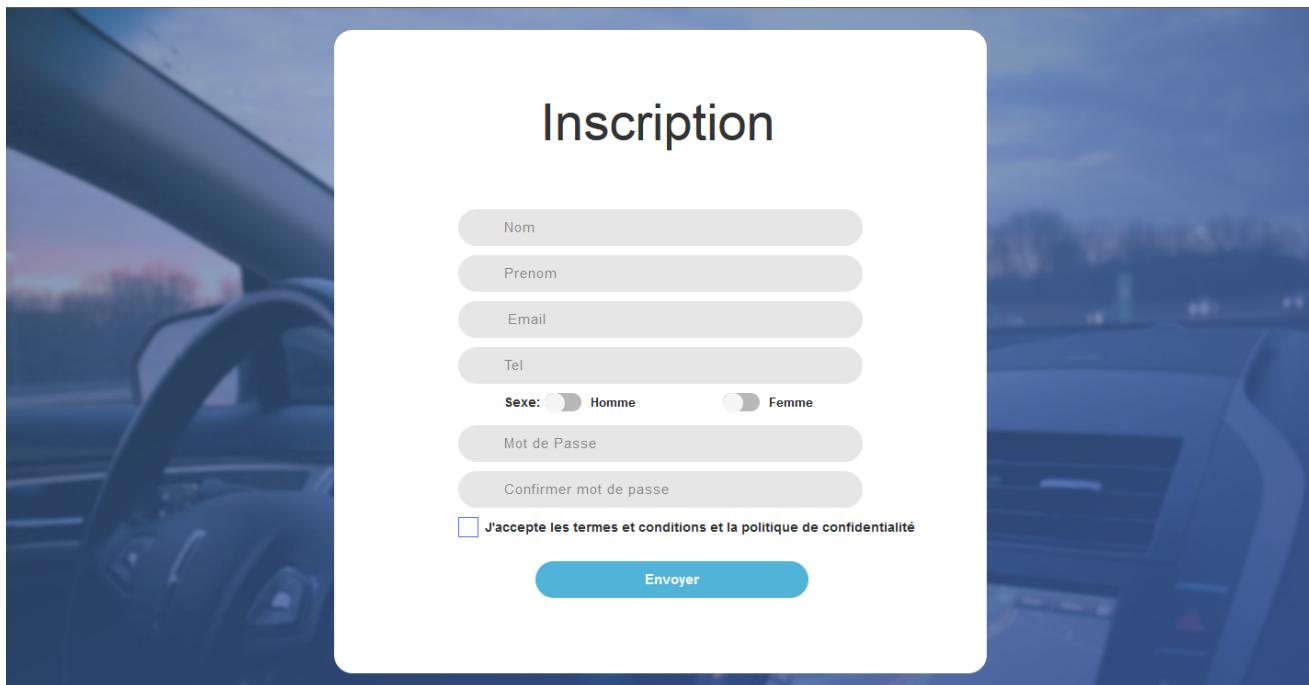


FIGURE 3.15 – Page d'inscription

Les figures ci-dessous présentent la validation de la confirmation de mot de passe. En effet, le bouton d'envoi est disabled jusqu'à ce que les mots de passes soient identiques

FIGURE 3.16 – Validation du mot de passe

Les figures présentées par la suite récapitulent l'ensemble des erreurs relatives à la validation du formulaire d'inscription avant son envoi.

Inscription

Inscription

The screenshot shows a registration form with two fields highlighted in red. The first field, 'Nom', contains 'aya' and has an error message: 'Veuillez utiliser au moins 2 caractères pour ce champ (vous utilisez actuellement 1 caractères)'. The second field, 'Prénom', contains 'b' and also has the same error message. Both fields have placeholder text 'Nom' and 'Prénom' respectively.

FIGURE 3.17 – Validation du nom et prénom

The screenshot shows a registration form with two fields highlighted in red. The first field, 'email', contains 'email' and has an error message: 'Le format valide: exemple@abc.xyz'. The second field, 'Téléphone', contains '8987676565' and has an error message: 'Le format valide: 06xxxxxxxx / 05xxxxxxxx / 07xxxxxxxx'. Both fields have placeholder text 'email' and 'Téléphone' respectively. A gender selection field 'Femme' is also visible.

FIGURE 3.18 – Validation du format d'email et de téléphone

The screenshot shows a registration form with a checkbox field highlighted in red. The checkbox is labeled 'J'accepte les termes et conditions et la politique de confidentialité' and has an error message: 'Veuillez cocher cette case si vous désirez poursuivre.' Below the checkbox is a 'Next Step' button.

FIGURE 3.19 – Validation de l'acceptation des conditions d'utilisation

3.3.4 Espace Admin

Après authentification réussie avec données correspondantes à un rôle **admin**, l'utilisateur a plus de privilèges et peut accéder à des sections différentes :

3.3.4.1 Section Accueil admin

La première interface à laquelle l'admin est redirigé est sa page d'accueil, qui ne contient de différent par rapport à la première page de notre plateforme que le titre .

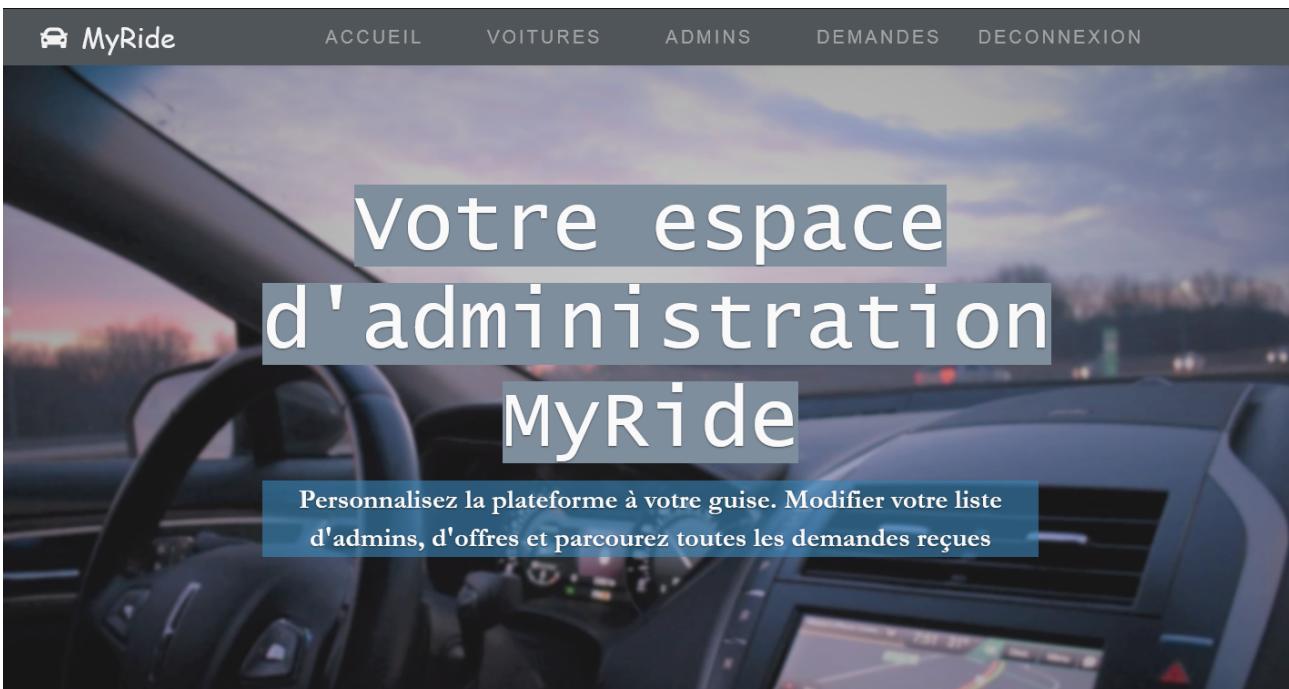


FIGURE 3.20 – Section accueil de l'espace admin

3.3.4.2 Section Admins

Dans cette section, l'admin consulte les détails des admins existants et peut en ajouter un nouveau. Pour chaque admin, il peut modifier les informations correspondantes ou effectuer une action de suppression.

The image shows the "Listes des Admins" (List of Admins) page from the MyRide administration interface. The page has a dark header with the "MyRide" logo and navigation links. Below the header is a large, rounded rectangular container with a white background. Inside this container, the title "Listes des Admins" is centered at the top. Below the title is a small blue button labeled "Ajouter un admin". Underneath the button is a table showing a single row of data. The table has columns for "Nom" (Name), "Prenom" (First Name), "Email", "Telephone" (Phone Number), "Sexe" (Gender), and "Actions". The data in the table is as follows:

Nom	Prenom	Email	Telephone	Sexe	Actions
ouzzine	aya	ayaouzzine1@gmail.com	0687676565	F	Edit Delete

FIGURE 3.21 – Section des admins de l'espace admin

Ajouter Admin

Nom

Prénom

Email

Téléphone

Sexe

Password

Save

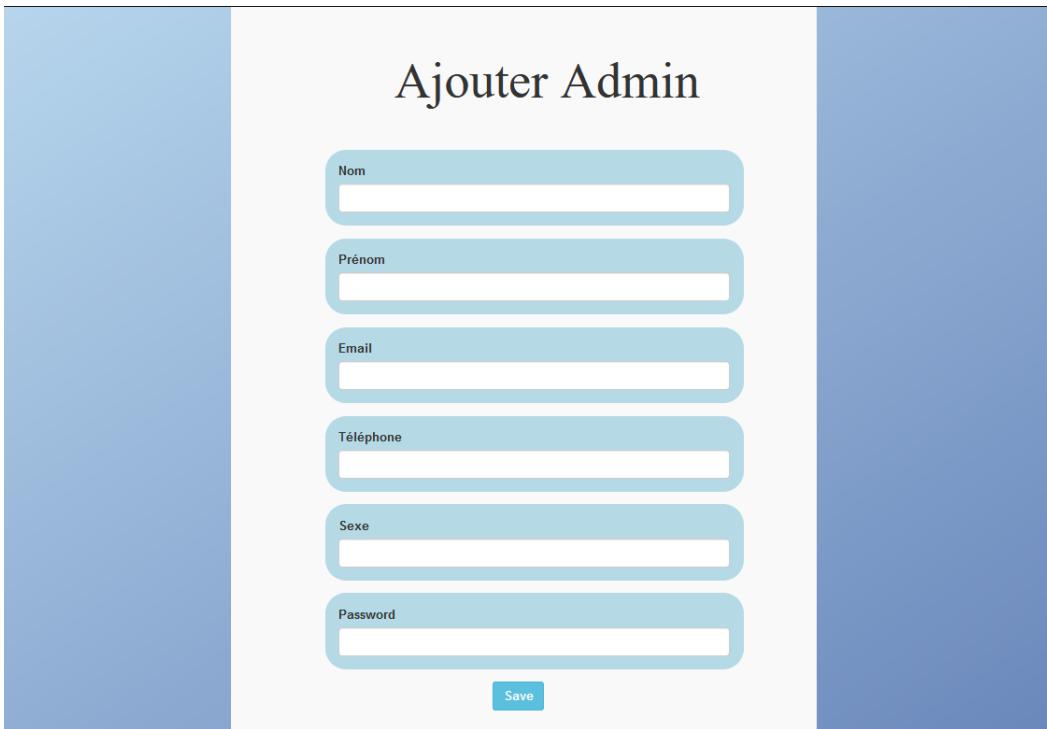


FIGURE 3.22 – Ajout d'un nouveau admin à la liste des admins

Modifier Admin

Nom

Prénom

Email

Téléphone

Sexe

Password

Save

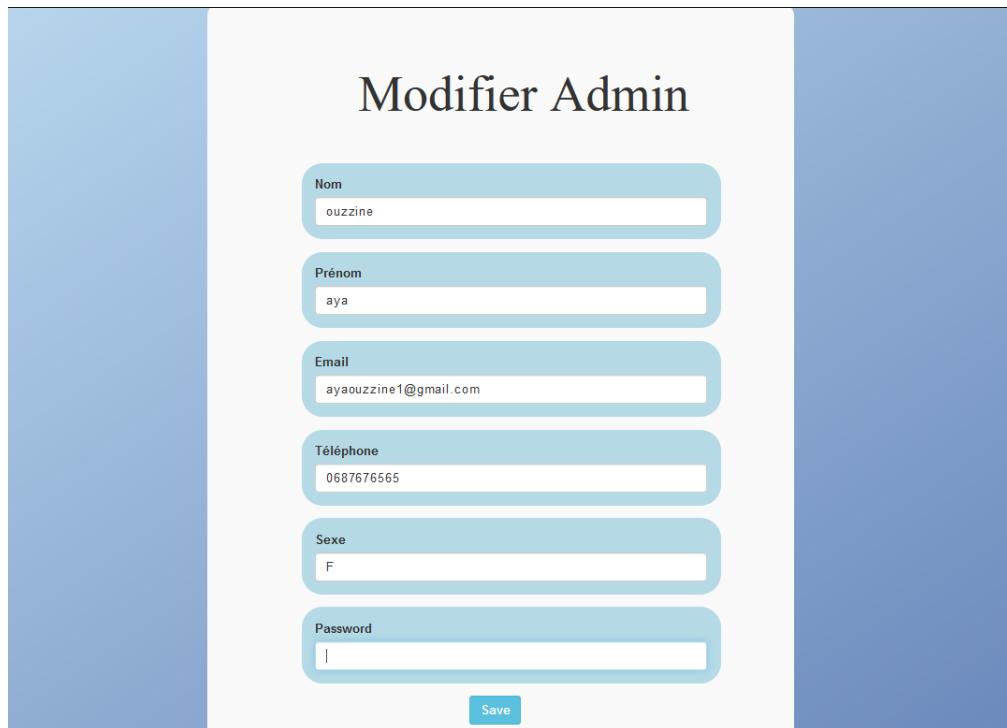


FIGURE 3.23 – Modification d'un admin de la liste des admins

3.3.4.3 Section Voitures

Dans cette section, l'admin consulte la liste des différents véhicules mis à disposition du client avec le matricule, le nombre maximal de personnes à transporter, le genre, une image. Pour chaque offre proposée, il peut modifier les informations correspondantes ou supprimer carrément de la liste.

Matricule	Nombre de personnes	Prix	Genre	Image	Actions
707070A20	4	200.0	4x4		Edit Delete
39872A72	7	500.0	limousine		Edit Delete

FIGURE 3.24 – Section des voitures de l'espace admin

Pour l'ajout et la modification, les interfaces sont similaires à celles traitant la partie des admins.

Matricule	Nombre de personnes	Prix	Genre	Image
707070A20	4	200.0	4x4	https://cdn.pixabay.com/photo/2017/01/28/16/04/range-rover_1660111.jpg

FIGURE 3.25 – Ajout et modification d'un véhicule

3.3.4.4 Section Demandes

Cette section affiche à l'admin l'ensemble des demandes effectuées par les clients ainsi que les détails du client concerné. L'admin peut trier ses demandes et n'afficher que celles qui sont traitées ou que celles qui ne le sont pas

The screenshot shows the MyRide admin interface. At the top, there is a navigation bar with links: ACCUEIL, VOITURES, ADMINS, DEMANDES, and DECONNEXION. Below the navigation bar, the main content area has a title "Liste des demandes effectuées". There are two buttons at the top of this section: "demandes non traitées" on the left and "demandes traitées" on the right. Underneath these buttons, there are three columns of information: "Details Commande", "Details Client", and "Etat". The "Details Commande" column contains a link to "Numéro demande : 1 Date de création : 2021-06-07 Prix : 200.0Dhs Etat de commande : NF". The "Details Client" column contains fields for "Nom : saidii Prénom : said", "Email : said@saidii.com", and "Numéro de téléphone : 0612121212". A blue button labeled "Marquer comme traitée" is located to the right of the client details. The background of the main content area features a blurred image of a car's interior.

FIGURE 3.26 – Section des demandes de l'espace admin

Le dernier traitement que peut effectuer l'admin concerne la validation des demandes saisies par les clients

The screenshot shows a confirmation dialog box with the title "Confirmer le traitement". It contains five input fields: "Nom" (Name) with value "saidii", "Prénom" (First Name) with value "said", "Email" with value "said@saidii.com", "Téléphone" (Phone) with value "0612121212", and "Prix" (Price) with value "200.0". At the bottom of the dialog is a blue "confirmer" (confirm) button.

FIGURE 3.27 – Validation d'une demande par l'admin

3.3.5 Espace Client

3.3.5.1 Section Informations Personnelles

C'est la première page vers laquelle on redirige un client après authentification réussie. Elle lui affiche l'ensemble de ses informations personnelles. Le client a une information concernant le nombre total des demandes qu'il a effectué et a un menu de navigation qui lui permet de parcourir les différentes sections de son espace.

The screenshot shows the 'Mes informations' section of the client's space. On the left, a sidebar displays the user's name 'Alex Smith' and social media icons for Twitter, Facebook, Instagram, and LinkedIn. Below these are links for 'Home', 'Mes informations', 'Mes demandes', 'Nouvelle demande', 'Services', and 'Contact'. The main content area is titled 'Mes informations' and contains a message: 'Vous trouvez ici les informations que vous partagez avec nous pour plus d'accessibilité.' Below this, there are two columns: 'Informations personnelles' and 'Demandes'. The 'Informations personnelles' column lists personal details with corresponding values: Nom : Smith, Prénom : Alex, Adresse email : alex_smith@gmail.com, Numéro de téléphone : 0656789320, and Sexe : Masculin. The 'Demandes' column shows a summary: Nombre total de demandes : 1 demande(s).

FIGURE 3.28 – Section informations de l'espace client

3.3.5.2 Section Nouvelle Demande

Dans cette section, le client personnalise sa demande en saisissant ses propres choix.

The screenshot shows the 'Nouvelle Demande' section of the client's space. The sidebar on the left is identical to Figure 3.28. The main content area is titled 'Nouvelle Demande' and includes a sub-instruction: 'Bénéficiez de nos services en effectuant des nouvelles demandes.'. Below this, there is a section labeled '#Création de demande' featuring a car icon. To the right, there are several input fields: 'Type de l'automobile' (dropdown), 'Marque de l'automobile' (dropdown), 'Lieu de départ' (text input), 'Lieu de destination' (text input), and a date input field 'jj/mm/aaaa --:--'. Below these are checkboxes for 'Equipements additifs': 'Wifi' and 'Chaise pour enfant'. At the bottom is a large dark button labeled 'Enregistrer la demande'.

FIGURE 3.29 – Section nouvelle demande de l'espace client

3.3.5.3 Section Demandes

Cette section affiche à l'utilisateur un récapitulatif sur l'ensemble des demandes qu'il a effectué.

The screenshot shows a dark-themed user interface for a client account. On the left, a sidebar menu for 'Alex Smith' includes links for Home, Mes informations, Mes demandes (which is selected), Nouvelle demande, Services, and Contact, along with social media sharing icons. The main content area is titled 'Mes demandes' and displays a table of one demand with columns for '#', Date de la demande, Prix, Lieu de départ, Lieu de destination, and Compléments. The demand details are: #1, Date: 2015-04-01 00:00:00.0, Prix: 100.0, Departure: Boulevard Zerkouni, Destination: 2Mars, Lycée MohamedV, and Compléments: WiFi : 1, Chaise pour enfant : 1.

FIGURE 3.30 – Section demandes de l'espace client

3.3.6 Section Nos Offres

Dans cette section, on regroupe l'ensemble des offres proposées par les admins. Elle est accessible directement depuis la page d'accueil sans avoir besoin à s'authentifier. Une page quasi-similaire est disponible dans l'espace **admin** sauf que ce dernier peut effectuer des traitements que le client ne peut pas faire, notamment la modification et la suppression.

The screenshot shows a web application for vehicle offers. At the top, a navigation bar includes a logo for 'MyRide' and links for ACCUEIL, NOS OFFRES (which is underlined), À PROPOS, CONTACT, and CONNEXION. The main content area is titled 'Nos Voitures' and features a table listing two vehicle options. The table columns are Nombre de personnes, Prix, Genre, and Image. The first row shows a car for 4 people at 200.0, 4x4 genre, and a black Range Rover image. The second row shows a car for 7 people at 500.0, limousine genre, and a white limousine image.

FIGURE 3.31 – Section des offres

Conclusion et perspectives

L'objectif de ce travail était de concevoir et réaliser une plateforme d'une agence privée de transport urbain liant clients et professionnels offrant ce service. Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de plate-formes de développement.

Même si les aspects concrets d'une plateforme de réservation d'automobiles n'ont pas tous vu la lumière, et que nous pouvions améliorer en ajoutant des fonctionnalités supplémentaires, ce projet avait un grand apport au niveau technique tout en développant en nous l'esprit communicationnel par le biais de la collaboration et de l'organisation qu'il nécessitait. Le travail s'est étendu sur une durée de quelques semaines au cours desquelles nous avons appris à bien nous organiser et gérer ensemble le temps dont nous disposions afin d'accomplir nos tâches dans les meilleurs délais.

Finalement, nous souhaitons que ce modeste travail soit à la hauteur des espérances de notre encadrant, à qui nous devons toute notre gratitude pour cette opportunité qu'il nous a offert, et au niveau de ses attentes.