



**ECOLE MAROCAINE DES  
SCIENCES DE L'INGENIEUR**  
Membre de 

---

HONORIS UNITED UNIVERSITIES

# **Rapport de projet développement mobile**

***Filière :***

Ingénierie Informatique et Réseaux

***Sous le thème :***

---

***Conception et réalisation d'une application mobile de  
marketplace florale***

---

***Réalisé par :***

Idar Maha

Kachchaf Sara

Chaouki Ghita

**Année universitaire :2024/2025**

# *Dédicace*

*Je dédie ce travail, fruit d'un long effort et d'une persévérance continue,*

***À nos très chers parents.***

*Aucune parole ne saurait traduire l'immense reconnaissance  
que **nous** ressentons pour votre amour inconditionnel, vos sacrifices constants et  
vos prières sincères. Vous êtes **notre** source de force, de courage et  
d'inspiration. Qu'Allah, le Tout-Puissant, vous accorde santé, bonheur et longue  
vie.*

***À nos professeurs,***

***Nous vous exprimons toute notre gratitude pour votre savoir, votre patience et  
votre accompagnement. Votre passion pour l'enseignement et votre dévouement  
resteront toujours une référence et un modèle pour nous.***

# *Remerciements*

Je tiens à exprimer ma profonde gratitude à notre professeure pour son accompagnement attentif tout au long de la réalisation de ce projet. Sa disponibilité, ses orientations méthodologiques ainsi que la clarté et la qualité de ses explications ont joué un rôle essentiel dans le bon déroulement et l'avancement de ce travail.

Je souhaite également remercier l'ensemble des personnes qui ont contribué, de près ou de loin, à la réussite de ce projet. Leurs remarques constructives, leurs suggestions pertinentes et leurs encouragements constants ont constitué un soutien précieux à chaque étape de cette étude.

Qu'il s'agisse d'un apport technique, méthodologique ou moral, chacune de ces contributions a eu une influence significative sur la qualité du travail réalisé. À toutes et à tous, j'adresse mes remerciements les plus sincères.

# *Résumé*

Ce projet consiste en la conception et la réalisation d'une application mobile de type marketplace florale, destinée à faciliter la commercialisation de produits floraux et la mise en relation entre clients et prestataires. L'application permet aux utilisateurs de consulter un catalogue de produits et de passer des commandes, tandis que les prestataires disposent d'un espace dédié pour gérer leurs produits et les commandes reçues.

La solution s'appuie sur une application mobile développée en **React Native**, un backend basé sur **ASP.NET Core** assurant la gestion de la logique métier et l'exposition des **API REST**, ainsi qu'une base de données **SQL Server** garantissant un stockage structuré et sécurisé des données. L'authentification des utilisateurs est assurée par **Firebase Authentication**, permettant une gestion fiable des accès et des rôles.

Ce projet s'inscrit dans une démarche de digitalisation du secteur floral, en proposant une plateforme centralisée visant à améliorer l'organisation des ventes et l'expérience utilisateur.

**Mots-clés :** Marketplace florale, application mobile, React Native, ASP.NET Core, SQL Server, Firebase Authentication.

# *Abstract*

This project focuses on the design and development of a mobile application for a floral marketplace, aimed at facilitating the commercialization of floral products and connecting customers with service providers. The application allows users to browse a product catalog and place orders, while service providers have access to a dedicated space to manage their products and received orders.

The solution is based on a mobile application developed using **React Native**, a backend built with **ASP.NET Core** that handles business logic and exposes **API REST**, and a **SQL Server** database ensuring structured and secure data storage. User authentication is managed through **Firestore Authentication**, providing reliable access control and role management.

This project is part of a digitalization approach within the floral sector, offering a centralized platform designed to improve sales organization and enhance user experience.

**Keywords :**Floral marketplace, mobile application, React Native, ASP.NET Core, SQL Server, Firestore Authentication.

# Table des matières

<b>Dédicace</b> .....	2
<b>Remerciements</b> .....	3
<b>Résumé</b> .....	4
<b>Abstract</b> .....	5
Liste des figures .....	8
Introduction générale .....	9
Chapitre1 : Contexte général du projet .....	10
1. Introduction générale.....	11
2. Présentation du projet .....	11
3. Contexte académique .....	11
4. Problématique .....	12
5. Objectifs du projet .....	12
5.1 Objectif général .....	12
5.2 Objectifs spécifiques .....	13
6. Exigences du projet .....	13
6.1 Exigences fonctionnelles.....	13
6.2 Exigences non fonctionnelles .....	14
7. Solution proposée .....	15
8. Architecture du projet .....	16
9. Planification et conduite du projet .....	16
9.1 Méthodologie Scrum .....	17
9.2 Outil de gestion de projet Taiga .....	17
10.Conclusion .....	18
Chapitre 2 : Analyse et conception.....	19
1.Introduction .....	20
2.Le langage UML : Un outil clé pour la modélisation des systèmes .....	20
2.1 Introduction .....	20
2.2 Définition et objectifs du langage UML .....	20
2.3 Un langage visuel universel.....	20
3. Diagramme des activités.....	21
4. Diagramme des cas d'utilisation.....	22
5.Diagramme de séquence .....	24
5.1 Diagramme de séquence Admin .....	24
5.2 Diagramme de séquence Prestataire.....	25
5.3 Diagramme de séquence Client .....	27
6.Diagramme de classe .....	30
Chapitre 3 : Réalisation .....	33

1. Introduction .....	34
2. Outils et technologies utilisés .....	34
3. Interfaces de l'application .....	38
3.1 Interface d'authentification .....	38
3.1.1 Interface d'inscription Client .....	38
3.1.2 Interface d'inscription Prestataire .....	39
3.1.3 Interface de connexion .....	40
3.1.4 Interface de réinitialisation du mot de passe .....	41
3.2 Tableau de bord Administrateur .....	42
3.2.1 Page d'accueil du tableau de bord Administrateur .....	42
3.2.2 Gestion des clients .....	43
3.2.3 Gestion des prestataires .....	44
3.2.4 Gestion des produits .....	45
3.2.5 Gestion des commandes .....	46
3.2.6 Gestion des notifications .....	47
3.3 Dashboard prestataire .....	48
3.3.1 Tableau de bord du prestataire .....	48
3.3.2 Gestion des produits .....	49
3.3.3 Gestion des commandes .....	50
3.3.4 Liste des promotions .....	51
3.3.5 Ajout d'un nouveau produit .....	52
3.3.6 Création d'une promotion .....	53
3.4 Interface Client .....	54
3.4.1 Page d'accueil (Home – Client) .....	54
3.4.2 Historique des commandes .....	55
3.4.3 Panier .....	56
3.4.4 Détails de livraison .....	57
3.4.5 Choix du mode de paiement – Carte bancaire .....	58
3.4.6 Choix du mode de paiement – Espèces (Cash) .....	59
3.4.7 Confirmation de commande .....	60
3.4.8 Reçu officiel .....	61
4. Conclusion .....	62
Conclusion et perspectives .....	63
Webographie .....	64

## Liste des figures

Figure 1:Hiérarchie des rôles Scrum .....	18
Figure 2:Diagramme de cas d'activité — FlowerMarket.....	22
Figure 3:. Diagramme des cas d'utilisation — FlowerMarket .....	23
Figure 4:Diagramme de séquence Admin .....	25
Figure 5:Diagramme de séquence Prestataire.....	27
Figure 6:Diagramme de séquence Client .....	29
Figure 7:Diagramme de classe .....	32
Figure 8:Logo React Native.....	34
Figure 9:Logo ASP.NET .....	35
Figure 10:Logo SQL Server Management.....	35
Figure 11:Logo FireBase .....	35
Figure 12:Logo Expo.....	36
Figure 13:Logo Visual Studio Code .....	36
Figure 14:Logo Visual Studio .....	37
Figure 15:Logo Github.....	37
Figure 16:Taiga .....	38



# Introduction générale

La transformation numérique joue aujourd'hui un rôle essentiel dans l'évolution des activités commerciales, notamment avec l'essor des applications mobiles et des plateformes en ligne. Les marketplaces mobiles permettent de faciliter les échanges entre vendeurs et clients, tout en améliorant l'accessibilité et l'efficacité des services proposés. Toutefois, le secteur floral repose encore en grande partie sur des méthodes de gestion traditionnelles, souvent peu adaptées aux besoins actuels.

Dans ce contexte, ce projet vise la conception et la réalisation d'une application mobile de marketplace florale destinée à centraliser la commercialisation de produits floraux. L'objectif est de proposer une plateforme unique permettant de simplifier la gestion des produits, des commandes et des interactions entre les différents acteurs du système.

Pour présenter ce travail de manière structurée et claire, le rapport est organisé en trois chapitres principaux :

- Le **premier chapitre** concerne la présentation du contexte général du projet.
- Le **deuxième chapitre** est consacré à la conception détaillée, avec la modélisation UML (diagrammes de cas d'utilisation, d'activités, de séquence et de classes).
- Enfin, le **troisième chapitre** présente la réalisation de l'application, en décrivant les outils et technologies utilisés, ainsi que les principales interfaces développées.

# Chapitre1 : Contexte général du projet

### 1. Introduction générale

La transformation numérique a profondément modifié les modes de consommation et de commercialisation dans de nombreux secteurs. L'essor des applications mobiles et des plateformes en ligne a favorisé l'émergence de solutions innovantes permettant de rapprocher les vendeurs et les clients, tout en simplifiant les processus de gestion et de vente. Parmi ces solutions, les applications de type marketplace occupent une place importante grâce à leur accessibilité et leur efficacité.

Cependant, le secteur floral repose encore largement sur des méthodes traditionnelles de gestion, telles que la vente en magasin physique ou la gestion manuelle des commandes. Ces pratiques peuvent limiter la visibilité des prestataires, compliquer le suivi des ventes et réduire la qualité de l'expérience client. Face à ces constats, la mise en place d'une solution numérique adaptée apparaît comme une nécessité.

C'est dans ce contexte que s'inscrit le présent projet, qui vise la conception et la réalisation d'une application mobile de marketplace florale permettant de centraliser la commercialisation de produits floraux et de faciliter les interactions entre les différents acteurs du système.

### 2. Présentation du projet

Le projet Flower Market consiste en le développement d'une application mobile dédiée à la vente de produits floraux à travers une plateforme de type marketplace. Cette application met en relation des clients souhaitant acheter des produits floraux et des prestataires proposant leurs produits via un espace dédié.

L'application offre une interface simple et intuitive permettant aux utilisateurs de consulter un catalogue de produits, de passer des commandes et d'en suivre l'évolution. Elle intègre également un espace de gestion destiné aux prestataires ainsi qu'un rôle d'administration assurant la supervision globale de la plateforme.

### 3. Contexte académique

Ce projet s'inscrit dans le cadre d'un travail académique dont l'objectif principal est de mettre en application les connaissances théoriques et pratiques acquises au cours du parcours de formation. Il permet d'exploiter et de consolider des compétences variées en développement d'applications, en conception de systèmes informatiques et en gestion de projets informatiques, tout en respectant une approche méthodologique rigoureuse.

La réalisation de ce projet mobilise plusieurs compétences clés, notamment l'analyse et la formalisation des besoins, la modélisation des systèmes, le développement logiciel ainsi que l'organisation et la planification du travail. Elle nécessite également la

prise en compte de contraintes techniques, fonctionnelles et organisationnelles, similaires à celles rencontrées dans un environnement professionnel réel.

Par ailleurs, ce projet offre l'opportunité d'aborder un cas concret inspiré de problématiques actuelles du monde professionnel, en particulier celles liées à la digitalisation des services et à la mise en place de solutions informatiques adaptées aux besoins des utilisateurs. Il permet ainsi de développer une vision globale du cycle de vie d'un projet informatique, depuis l'étude des besoins jusqu'à la conception, la réalisation et la validation de la solution proposée, tout en favorisant l'autonomie, la rigueur et le sens de la responsabilité dans la conduite du projet.

## 4. Problématique

La gestion traditionnelle de la vente de produits floraux repose encore majoritairement sur des pratiques manuelles ou semi-numérisées, telles que la vente directe en magasin, les commandes téléphoniques ou l'utilisation d'outils non spécialisés. Ces méthodes présentent plusieurs limites, notamment une faible visibilité des prestataires auprès d'un large public, une gestion souvent approximative des stocks et des commandes, ainsi qu'un suivi limité des ventes et des performances.

Du point de vue des clients, l'accès aux produits floraux peut s'avérer contraignant, en raison du manque d'informations centralisées, de la difficulté à comparer les offres et de l'absence de suivi en temps réel des commandes. Pour les prestataires, l'absence d'une solution numérique adaptée complique la gestion quotidienne des produits, ralentit le traitement des commandes et limite les opportunités de développement de leur activité.

L'absence d'une plateforme centralisée capable de regrouper l'ensemble des acteurs du secteur floral constitue ainsi un frein à l'optimisation des processus commerciaux. Elle engendre des échanges fragmentés, une perte de temps dans la gestion des informations et une diminution de l'efficacité globale du système.

Face à ces constats, la problématique principale de ce projet peut être formulée comme suit : **Comment concevoir une application mobile permettant de centraliser la commercialisation de produits floraux tout en facilitant la gestion des ventes et en améliorant l'interaction entre les clients, les prestataires et les administrateurs ?**

## 5. Objectifs du projet

### 5.1 Objectif général

L'objectif général de ce projet est de concevoir et de réaliser une application mobile de type marketplace florale visant à digitaliser le processus de vente de produits floraux. Cette application a pour ambition de centraliser la gestion des produits, des commandes et des utilisateurs au sein d'une plateforme unique, tout en optimisant les interactions entre les différents acteurs du système, à savoir les clients, les prestataires et l'administrateur. Elle vise également à améliorer l'efficacité des opérations commerciales et à offrir une meilleure expérience utilisateur grâce à une solution moderne et accessible.

## 5.2 Objectifs spécifiques

Afin d'atteindre l'objectif général, plusieurs objectifs spécifiques ont été définis :

- **Faciliter l'accès aux produits floraux pour les clients** en leur permettant de consulter un catalogue complet, structuré et régulièrement mis à jour, incluant les informations essentielles telles que les descriptions, les prix et la disponibilité des produits.
- **Permettre aux clients de passer des commandes de manière simple et sécurisée**, tout en leur offrant la possibilité de suivre l'évolution de leurs commandes à travers les différentes étapes du processus de vente.
- **Offrir aux prestataires un espace de gestion dédié**, leur permettant d'ajouter, de modifier et de supprimer leurs produits, de gérer les stocks et de traiter efficacement les commandes reçues.
- **Mettre en place un système de gestion des rôles et des accès**, garantissant que chaque utilisateur dispose uniquement des fonctionnalités correspondant à son rôle, afin d'assurer une utilisation sécurisée et cohérente de l'application.
- **Assurer le suivi des commandes et la gestion des paiements**, en permettant un contrôle précis des transactions et une meilleure traçabilité des opérations effectuées sur la plateforme.
- **Proposer une interface utilisateur simple, intuitive et adaptée aux appareils mobiles**, afin de garantir une prise en main rapide de l'application et une expérience utilisateur fluide pour l'ensemble des utilisateurs.

## 6. Exigences du projet

Les exigences du projet définissent l'ensemble des fonctionnalités attendues ainsi que les contraintes non fonctionnelles que la solution doit respecter afin de répondre efficacement aux besoins des utilisateurs et aux objectifs du projet.

### 6.1 Exigences fonctionnelles

Les exigences fonctionnelles décrivent les fonctionnalités principales que l'application doit offrir aux différents acteurs du système.

#### **Consultation du catalogue de produits floraux**

L'application doit permettre aux clients de consulter un catalogue de produits floraux structuré, comprenant les informations essentielles telles que le nom du produit, sa description, son prix, sa disponibilité ainsi que les informations relatives au prestataire.

#### **Gestion des comptes utilisateurs**

Le système doit prendre en charge l'inscription, l'authentification et la gestion des comptes utilisateurs. Chaque utilisateur doit être associé à un rôle précis

(client, prestataire ou administrateur), déterminant les fonctionnalités auxquelles il peut accéder.

### **Authentification et gestion des rôles**

L'application doit permettre aux utilisateurs de s'authentifier via un mécanisme sécurisé. Après authentification, le système doit identifier le rôle de l'utilisateur et lui accorder un accès correspondant à ses droits.

Les rôles pris en charge sont :

- Client : accès aux fonctionnalités de consultation et de commande.
- Prestataire : accès aux fonctionnalités de gestion des produits et des commandes.
- Administrateur : accès aux fonctionnalités de supervision et de gestion globale.

### **Gestion des produits par les prestataires**

Les prestataires doivent disposer d'un espace dédié leur permettant d'ajouter, de modifier et de supprimer leurs produits, de gérer les stocks et de mettre à jour les informations relatives à leurs offres.

### **Gestion des commandes et suivi de leur statut**

L'application doit permettre aux clients de passer des commandes et d'en suivre l'état. Les prestataires doivent pouvoir consulter les commandes reçues et mettre à jour leur statut tout au long du processus de traitement.

### **Validation des prestataires par l'administrateur**

Un mécanisme de validation doit être mis en place afin de permettre à l'administrateur d'approuver, de refuser ou de suspendre les comptes des prestataires avant qu'ils ne puissent proposer leurs produits sur la plateforme.

### **Consultation de statistiques globales par l'administrateur**

L'administrateur doit avoir accès à des statistiques globales relatives à l'activité de la plateforme, telles que le nombre de produits, de commandes et d'utilisateurs, afin de faciliter le suivi et la prise de décision.

## **6.2 Exigences non fonctionnelles**

Les exigences non fonctionnelles définissent les contraintes de qualité, de sécurité et de performance que la solution doit respecter.

### **Sécurité des données et des accès**

L'application doit garantir la protection des données personnelles et des informations sensibles. Les accès aux fonctionnalités doivent être sécurisés, et les échanges entre l'application mobile, le backend et la base de données doivent être protégés contre les accès non autorisés.

### **Fiabilité et disponibilité de l'application**

La solution doit assurer un fonctionnement stable et continu, avec un taux de disponibilité élevé, afin de garantir un accès permanent aux utilisateurs.

### **Performance et temps de réponse acceptable**

Les temps de chargement et de réponse doivent être optimisés afin d'offrir une expérience utilisateur fluide, même en cas de forte utilisation de la plateforme.

### **Ergonomie et facilité d'utilisation**

L'interface utilisateur doit être intuitive, claire et adaptée aux appareils mobiles, permettant une prise en main rapide par des utilisateurs de différents profils.

### **Maintenabilité et évolutivité de la solution**

L'architecture du système doit permettre une maintenance aisée ainsi que l'ajout futur de nouvelles fonctionnalités, afin d'assurer la pérennité et l'évolution de l'application.

## **7. Solution proposée**

La solution proposée consiste en la conception et la réalisation d'une application mobile de type marketplace florale permettant de centraliser la vente de produits floraux au sein d'une plateforme unique. Cette application vise à faciliter la mise en relation entre les clients et les prestataires, tout en offrant des outils de gestion adaptés aux différents acteurs du système.

L'application mobile est développée à l'aide de React Native, ce qui permet de proposer une interface utilisateur moderne, intuitive et compatible avec les principaux systèmes d'exploitation mobiles. Elle constitue la couche de présentation du système et permet aux clients de consulter le catalogue de produits, de passer des commandes et d'en assurer le suivi. Les prestataires disposent quant à eux d'un espace dédié pour gérer leurs produits et les commandes reçues.

Le backend de l'application est basé sur ASP.NET Core, et assure la gestion de la logique métier ainsi que l'exposition des fonctionnalités à travers des API REST. Il joue un rôle central dans le traitement des requêtes, la gestion des utilisateurs, des produits et des commandes, ainsi que dans le contrôle des accès selon les rôles définis (client, prestataire et administrateur).

Les données métier sont stockées dans une base de données relationnelle SQL Server, garantissant un stockage structuré, fiable et sécurisé des informations. L'authentification des utilisateurs est prise en charge par Firebase Authentication, permettant une identification sécurisée des utilisateurs. Le backend assure la validation des jetons d'authentification et la synchronisation des informations utilisateurs avec la base de données.

Cette solution repose ainsi sur une architecture client–serveur claire, favorisant la séparation des responsabilités, la sécurité, la maintenabilité et l'évolutivité du système. Elle permet de répondre efficacement aux besoins de digitalisation du secteur floral en proposant une plateforme performante, centralisée et adaptée aux exigences actuelles du commerce mobile.

## 8. Architecture du projet

L'architecture du projet repose sur une structure modulaire de type client–serveur, organisée autour de trois couches principales : l'application mobile, le backend applicatif et la base de données. Cette organisation permet une séparation claire des responsabilités, facilitant la maintenance, l'évolution et la sécurité du système.

La couche présentation est représentée par l'application mobile développée en React Native. Elle constitue le point d'interaction principal avec les utilisateurs et offre une interface intuitive et adaptée aux appareils mobiles. Cette couche permet aux clients de consulter le catalogue de produits floraux, de passer des commandes et d'en suivre l'état, tandis que les prestataires peuvent gérer leurs produits et les commandes reçues.

La couche logique métier est assurée par un backend basé sur ASP.NET Core. Celui-ci expose l'ensemble des fonctionnalités de l'application sous forme d'API REST, permettant la communication entre l'application mobile et le serveur. Le backend est responsable du traitement des requêtes, de la gestion des utilisateurs, des produits et des commandes, ainsi que de l'application des règles métier et du contrôle des accès selon les rôles définis.

L'authentification des utilisateurs est prise en charge par Firebase Authentication, garantissant une identification sécurisée et fiable. Après authentification, le backend procède à la validation des jetons reçus et à la synchronisation des informations utilisateurs avec la base de données métier, tout en attribuant les droits d'accès appropriés.

La couche données repose sur une base de données relationnelle SQL Server, utilisée pour le stockage des données métier telles que les utilisateurs, les produits, les commandes et les paiements. Ce choix permet d'assurer l'intégrité, la cohérence et la sécurité des informations.

Cette architecture modulaire favorise la scalabilité du système, la réutilisation du code et l'intégration future de nouvelles fonctionnalités. Elle constitue ainsi une base solide pour le développement et l'évolution de l'application Flower Market.

## 9. Planification et conduite du projet

La planification et la conduite de ce projet ont nécessité une organisation rigoureuse afin d'assurer le bon déroulement des différentes phases de réalisation. Compte tenu de la complexité du projet et de la diversité des



fonctionnalités à développer, une méthodologie de gestion de projet adaptée était indispensable pour structurer le travail, suivre l'avancement et garantir l'atteinte des objectifs fixés. Dans ce contexte, une approche agile a été adoptée afin de favoriser la flexibilité, la collaboration et l'amélioration continue.

## 9.1 Méthodologie Scrum

La méthode agile Scrum a été choisie en raison des nombreux avantages qu'elle offre, notamment :

- Une meilleure flexibilité dans la gestion du projet.
- Une transparence accrue grâce à une communication constante entre les membres de l'équipe et les parties prenantes.
- Une visibilité améliorée sur l'avancement du projet à travers des outils et rituels réguliers (comme les revues et les mêlées quotidiennes).
- Une capacité d'adaptation facilitée face à l'évolution des besoins du client ou du marché.
- Une approche centrée sur la valeur ajoutée, permettant de livrer en priorité les fonctionnalités les plus importantes pour le client.
- Une efficacité renforcée par une répartition claire des rôles (Scrum Master, Product Owner, Équipe de développement) et des responsabilités au sein de l'équipe Scrum.

Dans le cadre de Scrum, les rôles sont répartis entre les membres de l'équipe Scrum, qui travaillent ensemble pour atteindre les objectifs du projet. Les trois rôles clés dans Scrum sont :

- Le Product Owner représente les besoins métier et est responsable du Product Backlog. Il définit, priorise et le fait évoluer. En étroite collaboration avec l'équipe de développement, il s'assure que les fonctionnalités livrées répondent aux attentes des parties prenantes et aux objectifs stratégiques.
- Le ScrumMaster est le garant de l'application de la méthodologie Scrum. Il veille au bon déroulement des événements, facilite la collaboration au sein de l'équipe et élimine les obstacles qui pourraient entraver l'avancement du projet.
- L'équipe de développement est composée de professionnels pluridisciplinaires détenant toutes les compétences nécessaires pour réaliser les tâches du projet. Auto-organisée, elle collabore étroitement avec le Product Owner et le Scrum Master pour concevoir et livrer des fonctionnalités à haute valeur ajoutée

## 9.2 Outil de gestion de projet Taiga

Afin de soutenir l'application de la méthodologie Scrum et d'assurer un suivi structuré du projet, l'outil **Taiga** a été utilisé pour la gestion et la planification des tâches. Taiga est une plateforme de gestion de projet agile qui offre des fonctionnalités adaptées à l'organisation du travail selon les principes Scrum.

Dans le cadre de ce projet, Taiga a permis de centraliser l'ensemble des tâches et des fonctionnalités sous forme de **userstories**, organisées au sein d'un *product backlog*. L'outil a également été utilisé pour planifier les sprints, répartir les tâches et suivre leur état d'avancement à travers des tableaux visuels. Cette organisation a facilité la visualisation globale du projet et le suivi précis de chaque étape de développement.

L'utilisation de Taiga a contribué à améliorer la coordination du travail, à assurer une meilleure gestion du temps et à renforcer la transparence du processus de développement. Elle a également permis de respecter la méthodologie Scrum tout en garantissant une conduite de projet structurée, efficace et cohérente avec les objectifs définis.

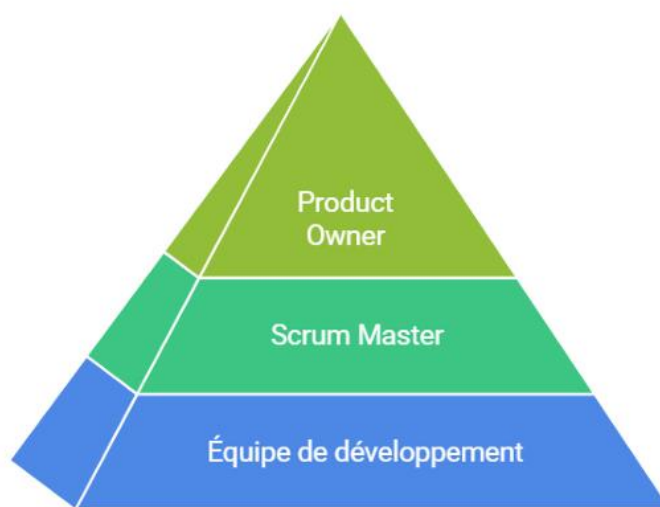


Figure 1: Hiérarchie des rôles Scrum

## 10. Conclusion

Ce chapitre a permis de présenter le contexte général du projet ainsi que les motivations ayant conduit à la réalisation d'une application mobile de type marketplace florale. Il a mis en évidence la problématique liée aux limites des méthodes traditionnelles de gestion de la vente de produits floraux et la nécessité de proposer une solution numérique centralisée.

Les objectifs du projet ont été définis, en distinguant l'objectif général et les objectifs spécifiques, afin de clarifier les attentes et les résultats visés. Les exigences fonctionnelles et non fonctionnelles ont ensuite été détaillées, permettant de préciser les fonctionnalités attendues de l'application ainsi que les contraintes de qualité, de sécurité et de performance à respecter.

Ce chapitre a également présenté la solution proposée ainsi que l'architecture générale adoptée pour répondre aux besoins identifiés. Enfin, la planification et la conduite du projet ont été abordées à travers l'adoption de la méthodologie agile Scrum et l'utilisation de l'outil Taiga pour le suivi et l'organisation des tâches, assurant ainsi une gestion structurée et efficace du projet.

# Chapitre 2 : Analyse et conception

### 1.Introduction

Ce chapitre présente une analyse détaillée et la conception d'une application web de gestion de workflow, développée pour répondre aux besoins de digitalisation des processus entre les commerciaux et le support IT. Nous proposons une vue d'ensemble des fonctionnalités essentielles et de l'expérience utilisateur, afin de garantir une meilleure traçabilité, une fiabilité accrue des données et une fluidité des échanges. Les modèles élaborés au moyen du langage UML illustrent les différentes interactions, processus métiers et structures internes du système. Ces représentations visuelles constituent un point de départ concret pour une exploration plus approfondie des mécanismes du projet.

## 2.Le langage UML : Un outil clé pour la modélisation des systèmes

### 2.1 Introduction

Les systèmes et les logiciels devenant de plus en plus complexes, il est indispensable de pouvoir les représenter, les comprendre et les documenter de manière claire et cohérente. UML (Unified Modeling Language) est un langage de modélisation qui répond à ce besoin en offrant une approche visuelle standardisée pour la conception et l'analyse des systèmes. Les différentes facettes d'UML, ses applications et son rôle très important dans la modélisation orientée objet sont abordés ici.

### 2.2 Définition et objectifs du langage UML

UML est un langage de modélisation visuelle standardisé visant à fournir un moyen de communication entre les différentes parties d'un projet de développement de logiciel. Il a été créé dans les années 1990 en fusionnant diverses méthodologies orientées objet, dans le but d'unifier les pratiques de modélisation. UML est riche en syntaxe et en sémantique. Il permet à la fois la description statique d'un système et son comportement dynamique. Il ne se limite pas à être un langage d'architecture logicielle ; il peut également servir à l'analyse fonctionnelle, à la conception technique ou même à la documentation des processus d'entreprise dans des domaines autres que l'informatique.

### 2.3 Un langage visuel universel

Tout comme un plan d'architecte ou un diagramme industriel, les diagrammes UML sont des représentations visuelles des objets d'un système (y compris leurs relations). Leur but est de

créer une compréhension commune entre les membres d'une équipe de projet, de favoriser l'identification des incohérences et de promouvoir une meilleure planification. UML utilise un certain nombre de types de diagrammes différents, classés en deux grandes catégories :

**Diagrammes de structure** : diagrammes qui décrivent certains aspects statiques du système (par exemple, diagrammes de classes, diagrammes de composants, diagrammes de déploiement).

**Diagrammes comportementaux** : diagrammes qui détaillent les interactions ou la dynamique interne du système (diagrammes de cas d'utilisation, diagrammes de séquence, diagrammes d'activité)

### 3. Diagramme des activités

Le diagramme d'activité global du système FlowerMarket illustre le déroulement général des interactions entre les différents acteurs de la plateforme, à savoir le client, le prestataire et l'administrateur. Il met en évidence les principales activités du système depuis l'authentification jusqu'à la gestion des commandes, des paiements et des notifications.

Le processus débute par l'authentification de l'utilisateur. Une fois authentifié, le système identifie le rôle associé à l'utilisateur et oriente le flux d'activités en fonction de ce rôle.

Si l'utilisateur est un client, il peut consulter le catalogue de produits, passer une commande, choisir un mode de paiement et suivre l'état de sa commande.

Si l'utilisateur est un prestataire, il peut gérer ses produits, consulter les commandes reçues, mettre à jour leur statut et consulter son revenu total.

Si l'utilisateur est un administrateur, il peut superviser la plateforme, consulter les commandes globales, valider les prestataires et suivre les revenus de la plateforme.

À chaque étape clé, le système met à jour la base de données et peut déclencher l'envoi de notifications aux acteurs concernés. Le processus se termine lorsque l'utilisateur se déconnecte ou quitte l'application.

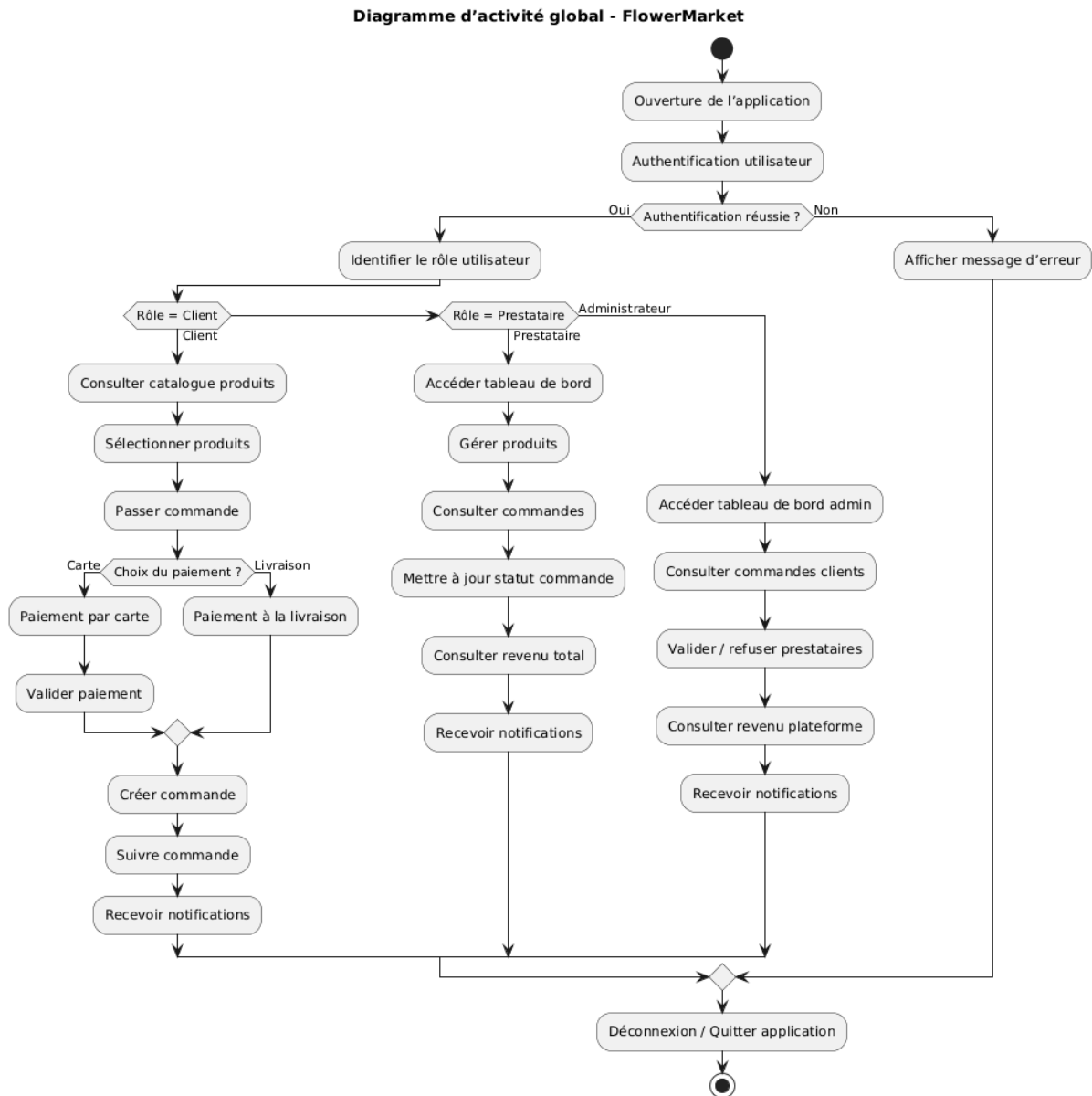


Figure 2:Diagramme de cas d'activité — FlowerMarket

## 4. Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation du système FlowerMarket décrit l'ensemble des interactions possibles entre les différents acteurs de la plateforme et les fonctionnalités proposées par l'application. Trois acteurs principaux interviennent dans le système : le client, le prestataire et l'administrateur. Le client peut s'authentifier, consulter le catalogue de produits floraux, passer des commandes, choisir un mode de paiement, suivre l'état de ses commandes et recevoir des notifications. Le prestataire, après authentification et validation de son compte, dispose d'un espace lui permettant de gérer ses produits, de consulter et de mettre à jour le statut des commandes reçues, de gérer les promotions et de consulter son revenu total. L'administrateur, quant à lui, assure la supervision globale de la plateforme en validant les prestataires, en gérant les

utilisateurs, en consultant les commandes des clients, en suivant les revenus globaux de la plateforme et en recevant des notifications relatives aux événements importants du système. L'ensemble de ces cas d'utilisation permet d'assurer une gestion centralisée, sécurisée et efficace de la marketplace florale.

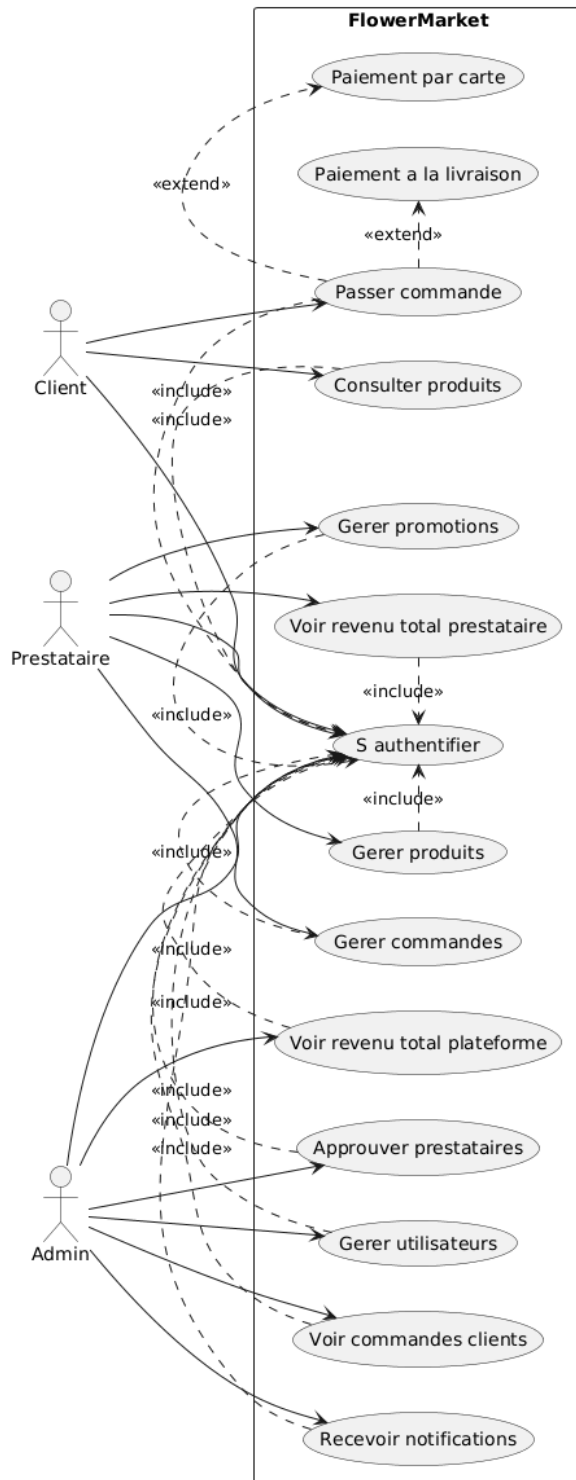


Figure 3.: Diagramme des cas d'utilisation — FlowerMarket

## 5. Diagramme de séquence

### 5.1 Diagramme de séquence Admin

Le diagramme de séquence de l'administrateur illustre de manière détaillée le fonctionnement des interactions entre l'interface administrateur, l'API backend, la base de données et le service de notification au sein du système FlowerMarket. Le processus commence lorsque l'administrateur saisit ses identifiants via l'interface dédiée, lesquels sont transmis au backend à travers une requête d'authentification. Le backend procède alors à la vérification des informations fournies en consultant la base de données, et, en cas de validation réussie, génère un jeton d'authentification permettant à l'administrateur d'accéder à son tableau de bord. Une fois authentifié, l'administrateur peut consulter la liste des commandes clients, ce qui déclenche l'envoi d'une requête vers l'API backend afin de récupérer les données correspondantes depuis la base de données et de les afficher sur l'interface. L'administrateur dispose également de la fonctionnalité d'approbation des prestataires, action qui entraîne la mise à jour du statut du prestataire dans la base de données et le déclenchement d'une notification à destination du prestataire concerné via le service de notification. Par ailleurs, l'administrateur peut consulter le revenu total généré par la plateforme, calculé à partir des données financières stockées. Tout au long de ce processus, le système assure la cohérence et la sécurité des échanges, tout en informant l'administrateur des événements importants grâce à l'envoi de notifications.



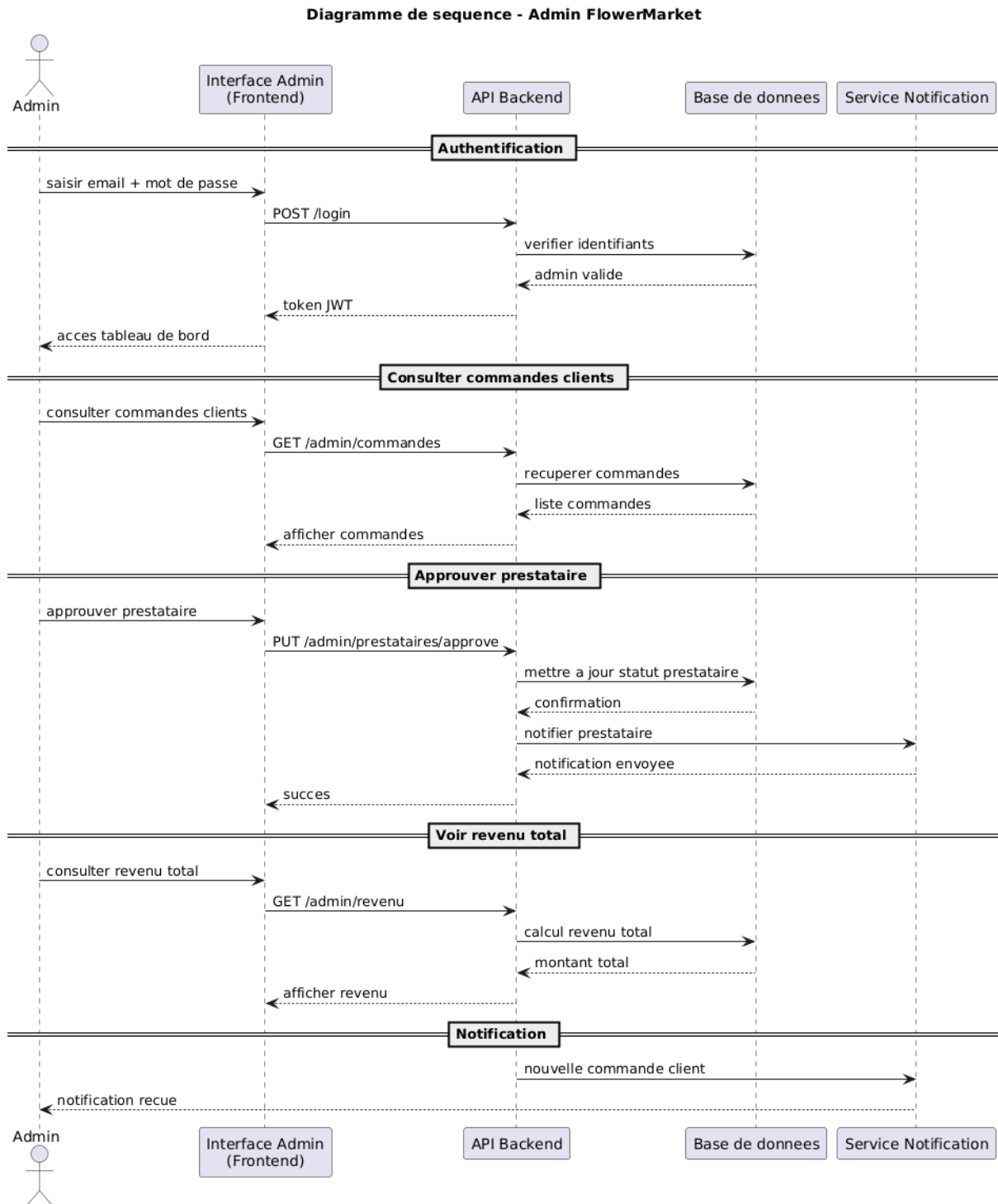


Figure 4:Diagramme de séquence Admin

## 5.2 Diagramme de séquence Prestataire

Le diagramme de séquence du prestataire décrit de manière détaillée les interactions entre l'interface prestataire, l'API backend, la base de données et le service de notification dans le système FlowerMarket. Le processus débute lorsque le prestataire s'authentifie via l'interface dédiée, entraînant l'envoi de ses identifiants au backend pour vérification. Après validation réussie, un jeton d'authentification est généré, permettant l'accès au tableau de bord prestataire. À partir de cet espace, le prestataire peut gérer ses produits en ajoutant, modifiant ou supprimant des articles, actions qui déclenchent des requêtes vers l'API backend afin de mettre à jour les informations correspondantes dans la base de données. Le prestataire peut également consulter les commandes reçues, ce qui implique la récupération des données depuis la base de données et leur affichage sur l'interface. Lors du traitement d'une commande, le prestataire met à jour son statut, entraînant une mise à jour des informations en base de données ainsi que l'envoi d'une notification au client concerné. Par ailleurs, le prestataire peut consulter son revenu total, calculé à partir des transactions enregistrées. Tout au long de ces interactions, le système assure la cohérence des données, la sécurité des échanges et la transmission des notifications nécessaires afin de garantir un suivi efficace des activités du prestataire.

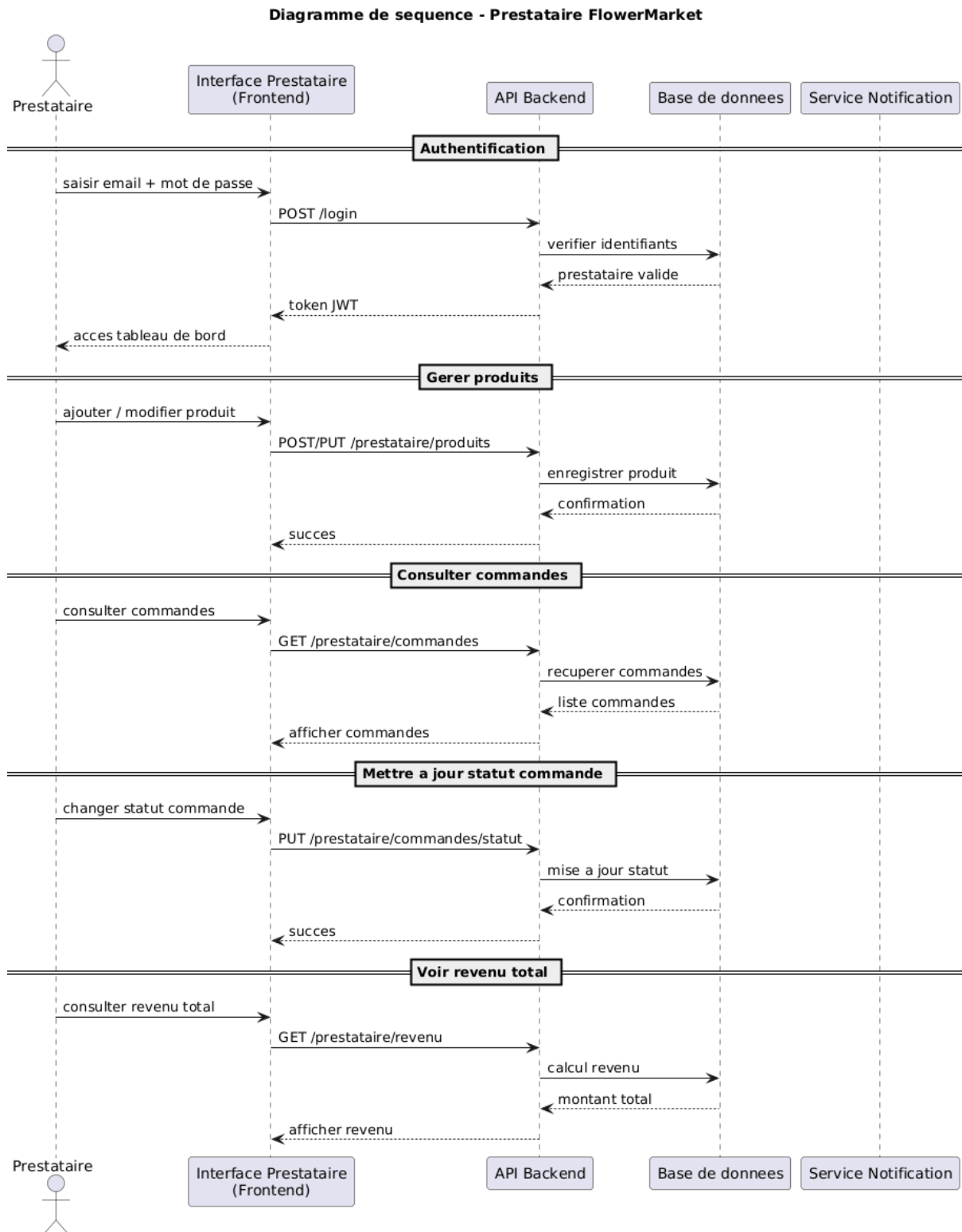


Figure 5: Diagramme de séquence Prestataire

### 5.3 Diagramme de séquence Client

le diagramme de séquence du client illustre le déroulement des interactions entre l'interface client, l'API backend, la base de données et le service de paiement au sein du

système FlowerMarket. Le processus commence par l'authentification du client via l'application mobile, suivie de la transmission de ses informations d'identification au backend pour vérification. Une fois l'authentification validée, un jeton d'accès est généré, permettant au client d'accéder aux différentes fonctionnalités de l'application. Le client peut alors consulter le catalogue de produits floraux, ce qui déclenche une requête vers l'API afin de récupérer les données correspondantes depuis la base de données et de les afficher sur l'interface. Lorsqu'un produit est sélectionné, le client peut passer une commande, action qui entraîne la création d'une nouvelle commande dans la base de données. Le client choisit ensuite un mode de paiement, et le système procède au traitement du paiement via le service dédié avant de confirmer la commande. Tout au long du processus, le client peut suivre l'état de sa commande, les mises à jour de statut étant récupérées depuis la base de données et affichées en temps réel. Enfin, le système envoie des notifications au client afin de l'informer des différentes étapes de traitement de sa commande, garantissant ainsi une expérience utilisateur fluide et transparente.

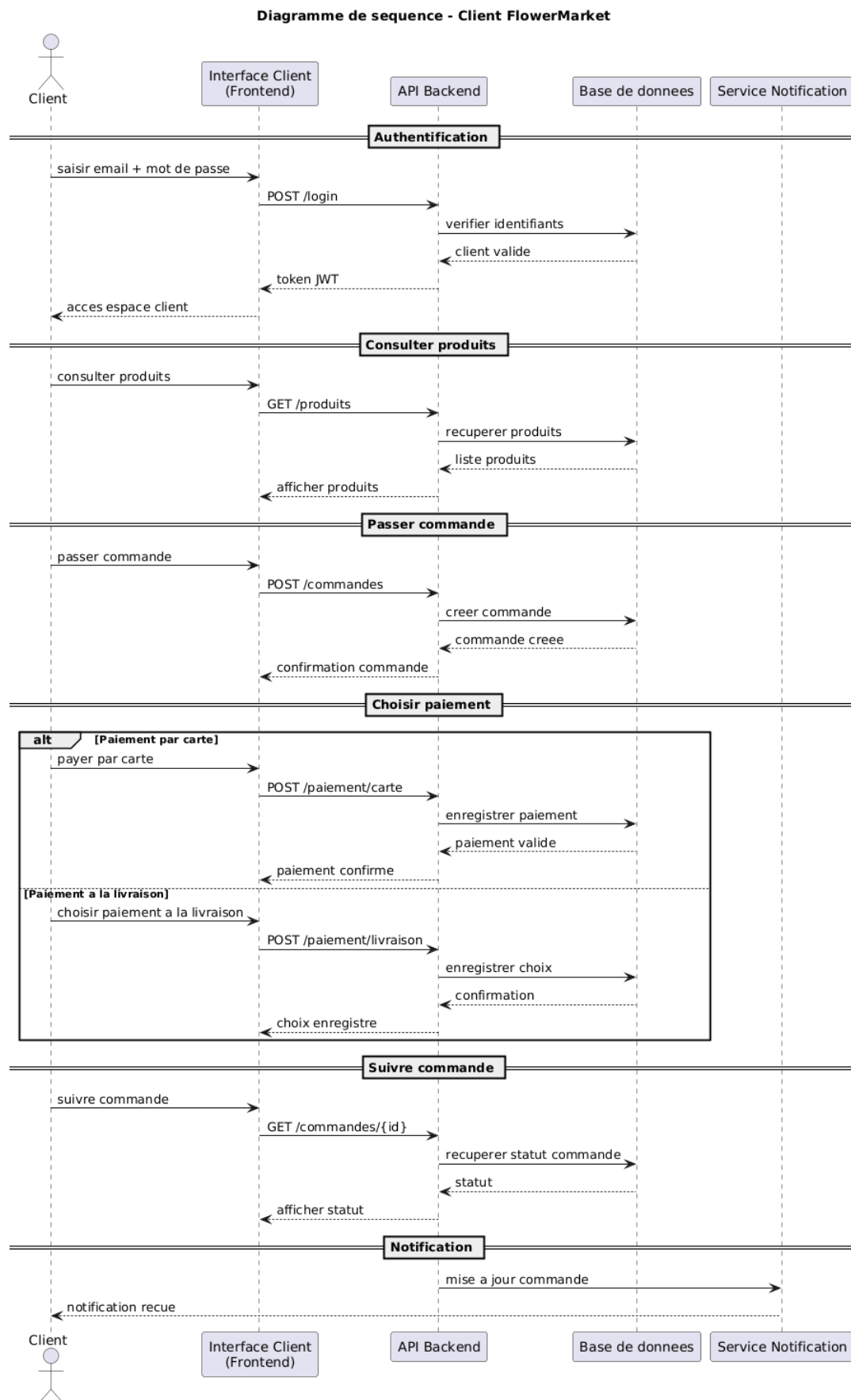


Figure 6: Diagramme de séquence Client

## 6. Diagramme de classe

Ce diagramme de classes modélise la structure interne de l'application **FlowerMarket** en représentant les principales entités du système, leurs attributs ainsi que les relations qui les lient. Il permet de décrire l'organisation des données et de mettre en évidence les responsabilités de chaque classe dans le fonctionnement global de la marketplace florale.

- **AppUser** : cette classe représente un utilisateur générique de l'application. Elle regroupe les attributs communs à tous les utilisateurs, tels que l'identifiant, le nom complet, l'adresse e-mail, l'état de validation du compte (*IsApproved*) et la confirmation de l'adresse e-mail (*EmailConfirmed*). Elle constitue la classe mère dont héritent les différents types d'utilisateurs du système.
- **Client** : classe héritée de *AppUser*, elle représente les utilisateurs finaux de la plateforme. Le client est responsable de la passation des commandes et de l'interaction avec les produits proposés. Il est associé à la classe *Order*, reflétant le fait qu'un client peut passer plusieurs commandes au cours de son utilisation de l'application.
- **Prestataire** : classe héritée de *AppUser*, elle modélise les vendeurs de la plateforme. Le prestataire possède une boutique (*Store*) et est responsable de la gestion des produits qu'il propose. Cette relation traduit le lien direct entre le prestataire et sa boutique, qui contient l'ensemble des produits mis en vente.
- **Admin** : classe héritée de *AppUser*, représentant l'administrateur du système. Il est chargé de la supervision globale de la plateforme, notamment à travers la validation des prestataires et la gestion des notifications reçues.
- **Store** : cette classe représente la boutique associée à un prestataire. Elle contient des informations telles que l'identifiant, le nom, la description et l'adresse de la boutique. Une boutique contient plusieurs produits, traduisant la relation de composition entre *Store* et *Product*.
- **Product** : cette classe modélise les produits floraux proposés sur la plateforme. Elle contient les attributs essentiels tels que le nom, le prix et l'URL de l'image. Un produit peut être associé à plusieurs promotions et recevoir plusieurs avis (*Review*).
- **Promotion** : cette classe représente les promotions appliquées aux produits. Elle contient des informations telles que le titre, le pourcentage de réduction ainsi que les dates de début et de fin. Une promotion est associée à un produit donné.
- **Order** : cette classe représente une commande passée par un client. Elle contient des informations relatives à la commande, telles que la date de création, la quantité totale et le prix total. Une commande est liée à un client, à un produit et à une boutique, et peut déclencher l'envoi de notifications.

- **Review** : cette classe permet de gérer les avis laissés par les clients sur les produits. Elle contient la note attribuée, le commentaire ainsi que la date de création de l'avis. Un produit peut recevoir plusieurs avis, reflétant l'interaction entre les clients et les produits.
- **Notification** : cette classe modélise les notifications envoyées aux utilisateurs. Elle contient le message, la date de création, l'état de lecture et le type de notification. Les notifications peuvent être reçues par les clients, les prestataires ou les administrateurs, notamment lors de la création ou de la mise à jour d'une commande.

Les relations entre ces classes traduisent le fonctionnement global de l'application : les clients passent des commandes contenant des produits proposés par des prestataires à travers leurs boutiques, les prestataires gèrent leurs produits et leurs ventes, tandis que l'administrateur supervise la plateforme. Les notifications assurent la communication entre les différents acteurs lors des événements importants. Ce modèle de classes offre ainsi une organisation claire, cohérente et évolutive, facilitant la maintenance et l'évolution future de l'application FlowerMarket.

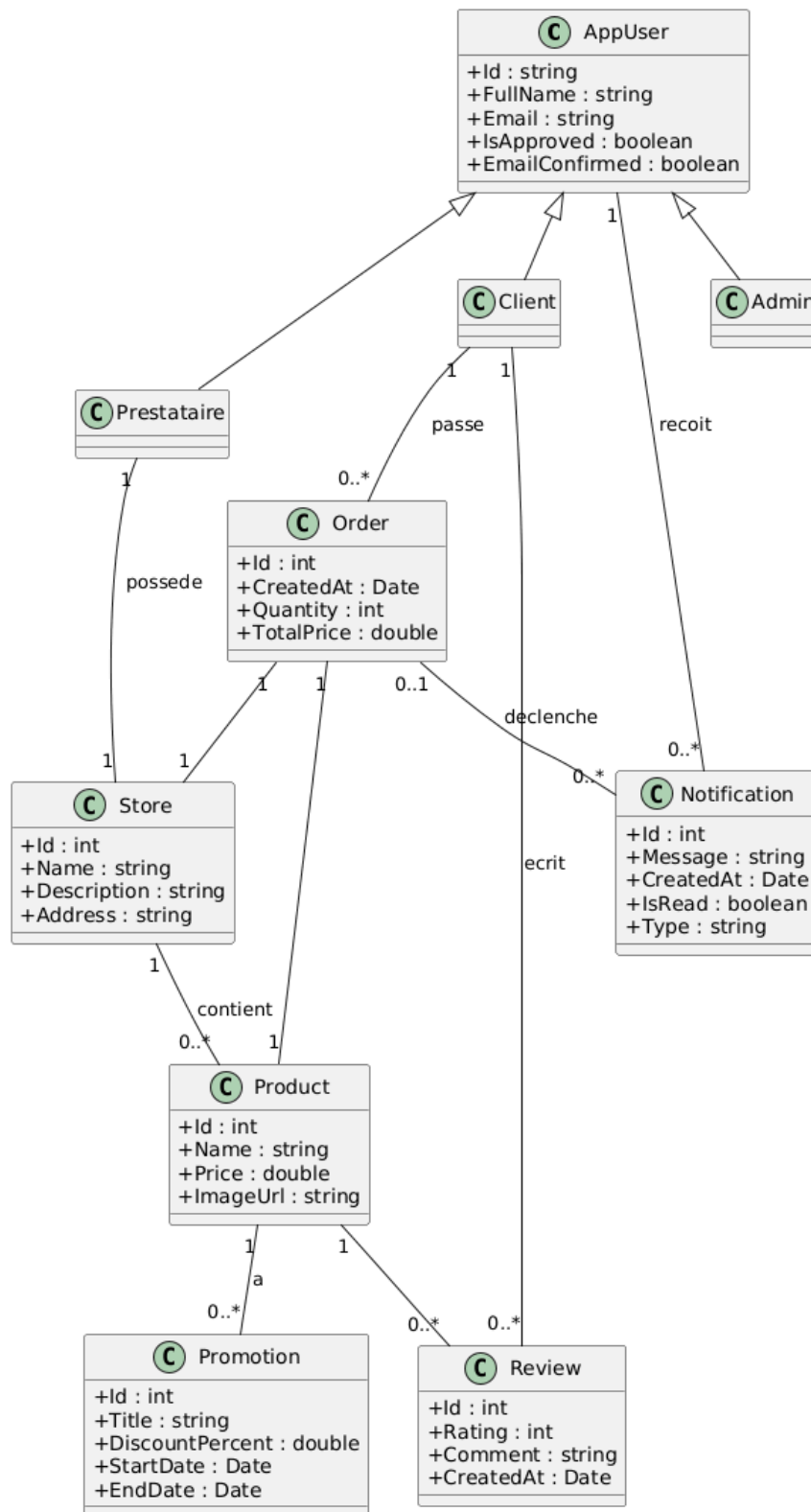


Figure 7: Diagramme de classe



# Chapitre 3 : Réalisation

## 1. Introduction

Ce chapitre est consacré à la phase de réalisation du projet FlowerMarket. Après les étapes d'analyse et de conception, cette phase vise à concrétiser les choix techniques et fonctionnels définis précédemment à travers le développement effectif de l'application. Elle consiste à mettre en œuvre l'architecture retenue, à intégrer les différents composants du système et à assurer leur bon fonctionnement.

La réalisation du projet repose sur une approche structurée permettant de garantir la cohérence entre les besoins exprimés et les fonctionnalités implémentées. Elle inclut le développement de l'application mobile, la mise en place du backend, la gestion de la base de données ainsi que l'intégration des mécanismes d'authentification et de sécurité.

## 2. Outils et technologies utilisés

Le développement de l'application a nécessité l'utilisation d'un ensemble d'outils et de technologies modernes, choisis pour leur robustesse, leur compatibilité et leur pertinence par rapport aux besoins du projet.

**React native : utilisé pour le développement du frontend :**



Figure 8: Logo React Native

React Native a été utilisé pour le développement de l'application mobile FlowerMarket. Ce framework permet de créer des applications mobiles multiplateformes à partir d'un code source unique, offrant ainsi une interface utilisateur fluide et performante. Son utilisation facilite le développement et la maintenance de l'application tout en garantissant une expérience utilisateur adaptée aux appareils mobiles.

## ASP.NET Core : utilisé pour la partie backend :



Figure 9:Logo ASP.NET

ASP.NET Core constitue la base du backend de l'application. Il est chargé de la gestion de la logique métier, du traitement des requêtes et de l'exposition des fonctionnalités sous forme d'API REST. Cette technologie a été choisie pour sa robustesse, ses performances élevées et sa capacité à gérer efficacement les échanges entre l'application mobile et la base de données.

## SQL Server – Système de gestion de base de données relationnelle :



Figure 10:Logo SQL Server Management

Utilisé comme **Système de Gestion de Base de Données Relationnelle (SGBDR)**, SQL Server centralise toutes les informations du projet. Il garantit l'intégrité, la cohérence et la persistance des données concernant les utilisateurs, les demandes et leur statut, tout en permettant une exploitation optimisée grâce à des requêtes relationnelles.

## Firebase Authentication



Figure 11:Logo FireBase

Firebase Authentication est utilisé pour la gestion de l'authentification des utilisateurs. Il permet d'assurer une identification sécurisée et fiable des utilisateurs grâce à la génération de jetons

d'authentification. Le backend se charge ensuite de la validation de ces jetons et de la gestion des rôles afin de contrôler l'accès aux différentes fonctionnalités de l'application.

## **Expo**



*Figure 12:Logo Expo*

Expo a été utilisé comme environnement de développement pour l'application mobile FlowerMarket. Il s'agit d'un ensemble d'outils et de services facilitant le développement, le test et le déploiement d'applications React Native. Expo permet d'accélérer le cycle de développement grâce à des fonctionnalités intégrées telles que le rechargement à chaud, la gestion simplifiée des dépendances et l'exécution de l'application sur des appareils physiques ou des émulateurs sans configuration complexe. Son utilisation a contribué à améliorer la productivité et à simplifier les phases de test et de débogage de l'application

## **Visual Studio Code (VS Code) - Environnement de développement :**



*Figure 13:Logo Visual Studio Code*

Utilisé principalement pour le développement du **frontend**, cet éditeur de code léger et extensible a facilité la gestion des fichiers React Native, l'autocomplétions et l'intégration des dépendances.

### **Visual Studio : utilisé pour la partie backend**



*Figure 14:Logo Visual Studio*

Cet IDE complet a permis de développer en C# avec ASP.NET Core, de compiler et exécuter les API, de gérer les dépendances et de tester les fonctionnalités avec SQL Server. Son intégration native avec l'écosystème Microsoft en fait un outil central pour ce projet.

### **GitHub – Gestion du code source et workflow collaboratif :**



*Figure 15:Logo Github*

GitHub a été utilisé comme plateforme d'hébergement du code source et de gestion de versions. Il permet de suivre l'évolution du projet, de gérer les différentes versions du code et de faciliter le travail collaboratif à travers l'utilisation de branches et de commits.

### **Taiga**



Figure 16:Taiga

Taiga est un outil de gestion de projet agile utilisé pour la planification et le suivi des tâches. Il permet d'organiser le travail selon la méthodologie Scrum, de gérer le backlog, de planifier les sprints et de visualiser l'avancement du projet à travers des tableaux Kanban.

### 3. Interfaces de l'application

Les interfaces de l'application FlowerMarket ont été conçues afin d'offrir une expérience utilisateur fluide, intuitive et adaptée aux différents profils d'utilisateurs. L'application distingue quatre interfaces principales : l'interface d'authentification, le tableau de bord administrateur, le tableau de bord prestataire et l'interface client. Chaque interface répond à des besoins spécifiques tout en conservant une cohérence graphique globale.

#### 3.1 Interface d'authentification

##### 3.1.1 Interface d'inscription Client

Cette interface est destinée aux utilisateurs souhaitant créer un compte client. Le client renseigne ses informations personnelles, notamment le nom complet, l'adresse e-mail et le mot de passe. L'inscription est gérée par **Firestore Authentication**, qui crée le compte et sécurise les données d'accès. Une fois l'inscription réussie, le client peut se connecter à l'application et accéder directement aux fonctionnalités de consultation des produits et de passation de commandes. Cette interface a été conçue pour offrir un processus d'inscription simple et rapide.



The screenshot shows the 'Flower Market' account creation page. At the top, the title 'Flower Market' is displayed in a large, white, serif font against a dark green background. Below the title, the text 'Créez votre compte' is written in a smaller, white, sans-serif font. The main content area is a white card with rounded corners. It features a section titled 'Créer un compte' in bold black text. Below this title are three input fields: 'Nom complet' with a person icon, 'Email' with an envelope icon, and 'Mot de passe' with a lock icon and a toggle eye icon. A dark green button labeled 'Créer mon compte' is positioned below the input fields. At the bottom of the card, there is a link that reads 'Déjà un compte ? Se connecter'.


# Flower Market

Créez votre compte

## Créer un compte

 Nom complet

 Email

 Mot de passe 

Créer mon compte

[Déjà un compte ? Se connecter](#)

### 3.1.2 Interface d'inscription Prestataire

Cet écran permet aux prestataires de créer un compte vendeur afin de proposer leurs produits sur la plateforme. Le formulaire d'inscription recueille les informations essentielles telles que le nom complet, l'adresse e-mail, le mot de passe et l'adresse. La création du compte est assurée par **Firestore Authentication**, qui enregistre les informations d'authentification de manière sécurisée. Toutefois, le compte prestataire reste inactif après l'inscription et nécessite une validation par l'administrateur avant de pouvoir accéder aux fonctionnalités de vente. Cette étape garantit un meilleur contrôle des prestataires présents sur la plateforme.

**Flower Market**  
Inscription Prestataire

**Créer un compte vendeur**  
Votre compte sera soumis à validation avant d'être activé.

Nom complet

Email

Mot de passe ☐

Adresse

S'inscrire

[← Retour à la connexion](#)

### 3.1.3 Interface de connexion


Cette interface correspond à l'écran de connexion principal de l'application FlowerMarket. Elle permet aux utilisateurs existants de s'authentifier en saisissant leur adresse e-mail et leur mot de passe. Les informations saisies sont transmises à **Firestore Authentication**, qui se charge de vérifier la validité des identifiants. En cas de succès, Firestore génère un jeton d'authentification permettant à l'utilisateur d'accéder à l'application. Une fois authentifié, l'utilisateur est redirigé vers l'interface correspondant à son rôle (client , prestataire ou admin ). L'écran propose également un accès à la création de compte ainsi qu'à la récupération du mot de passe.



**Flower Market**  
Votre marché floral

**Connexion**

✉ Adresse email


🔒 Mot de passe 

Se connecter

[Mot de passe oublié ?](#)

ou

Créer un compte

 se connecter en tant que prestataire


#### 3.1.4 Interface de réinitialisation du mot de passe


Cet écran permet aux utilisateurs ayant oublié leur mot de passe de récupérer l'accès à leur compte. L'utilisateur saisit son adresse e-mail, qui est transmise à **Firestore Authentication**. Le service envoie automatiquement un lien de réinitialisation à l'adresse fournie, permettant à l'utilisateur de définir un nouveau mot de passe de manière sécurisée. Cette fonctionnalité améliore l'expérience utilisateur tout en renforçant la sécurité globale du système.


# Flower Market

Votre marché floral

## Connexion


 maha@gmail.com






Se connecter


Mot de passe oublié ?





### Réinitialiser le mot de passe

Entrez votre email et nous vous enverrons un lien pour réinitialiser votre mot de passe.

 maha@gmail.com

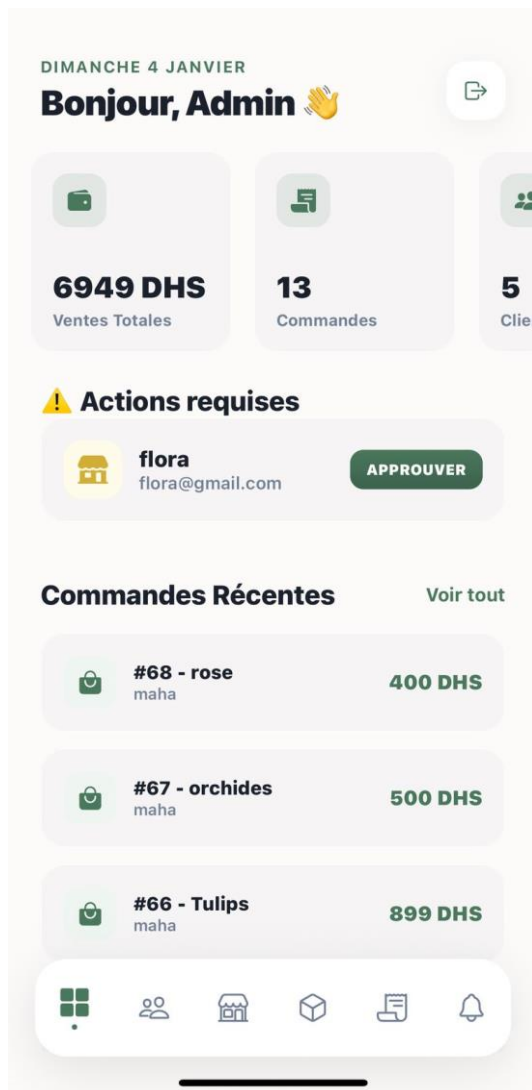
Envoyer le lien

Retour

## 3.2 Tableau de bord Administrateur

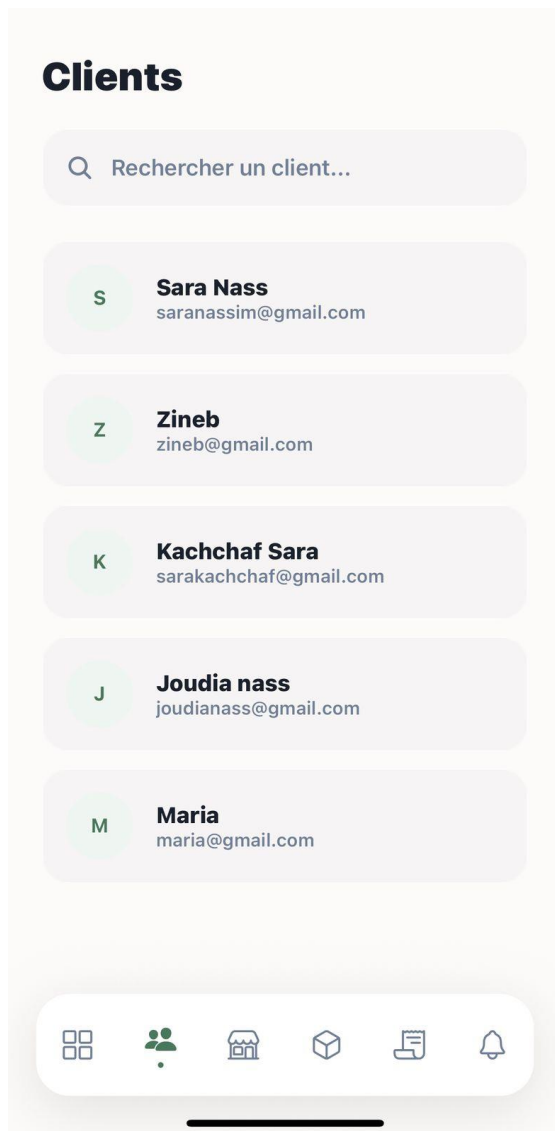
### 3.2.1 Page d'accueil du tableau de bord Administrateur

Cette page correspond à l'accueil du tableau de bord administrateur de l'application FlowerMarket. Elle permet à l'administrateur d'avoir une vue globale sur l'activité de la plateforme grâce à des indicateurs clés tels que le total des ventes, le nombre de commandes et le nombre de clients. Elle met également en évidence les actions nécessitant une intervention immédiate, comme la validation des prestataires, ainsi que les commandes récentes, facilitant ainsi le suivi et la supervision de la marketplace.



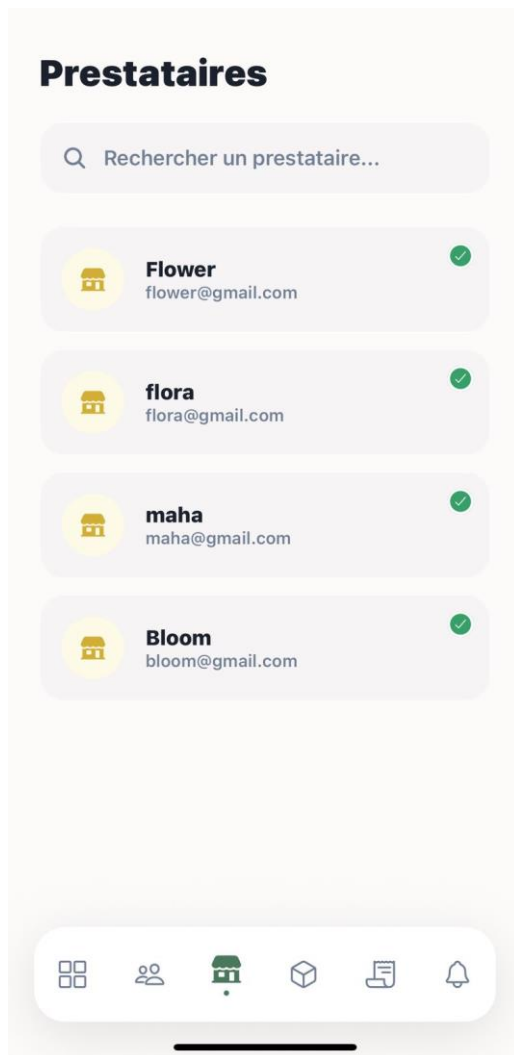
### 3.2.2 Gestion des clients

Cette interface présente la liste des clients inscrits sur la plateforme FlowerMarket. Elle permet à l'administrateur de consulter les informations essentielles des utilisateurs, telles que le nom et l'adresse e-mail. Une fonctionnalité de recherche est également disponible afin de retrouver rapidement un client spécifique. Cet écran contribue à une meilleure supervision des utilisateurs finaux de l'application.



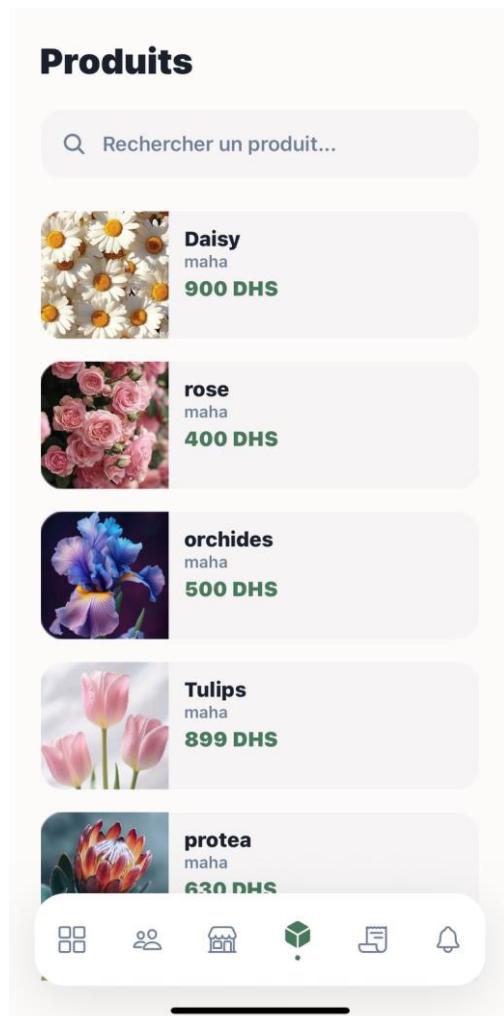
### 3.2.3 Gestion des prestataires

Cette interface est dédiée à la gestion des prestataires inscrits sur la plateforme. Elle affiche la liste des vendeurs avec leurs informations principales, notamment le nom et l'adresse e-mail. Un indicateur visuel permet d'identifier rapidement les prestataires validés. À travers cet écran, l'administrateur peut approuver ou refuser les demandes d'inscription des prestataires, garantissant ainsi la fiabilité et la qualité des vendeurs présents sur la plateforme.



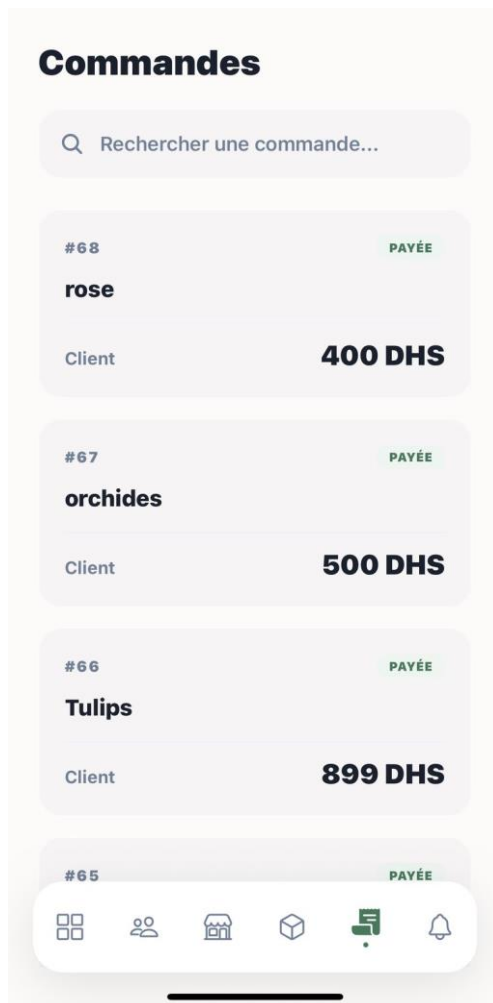
#### 3.2.4 Gestion des produits

Cet écran permet à l'administrateur de consulter l'ensemble des produits disponibles sur la plateforme FlowerMarket. Chaque produit est affiché avec son image, son nom, le nom du prestataire associé ainsi que son prix. Une barre de recherche est mise à disposition afin de faciliter la recherche d'un produit spécifique. Cette interface offre à l'administrateur une vue globale sur les produits publiés et permet d'assurer un contrôle sur le contenu proposé au sein de la marketplace.



### 3.2.5 Gestion des commandes

Cet écran permet à l'administrateur de consulter l'ensemble des commandes passées sur la plateforme. Chaque commande est affichée avec son identifiant, le produit concerné, le client associé, le montant total ainsi que le statut du paiement. Cette interface offre une vue globale sur l'activité commerciale de la marketplace et permet à l'administrateur de suivre l'évolution des transactions et des paiements.



### 3.2.6 Gestion des notifications

Cet écran permet à l'administrateur de consulter les notifications liées aux événements importants de la plateforme, tels que la réception de nouvelles commandes ou les actions nécessitant une intervention administrative. Chaque notification indique la date, l'heure et le contenu de l'événement, avec la possibilité de marquer les notifications comme lues. Cette interface facilite le suivi en temps réel de l'activité de la marketplace.

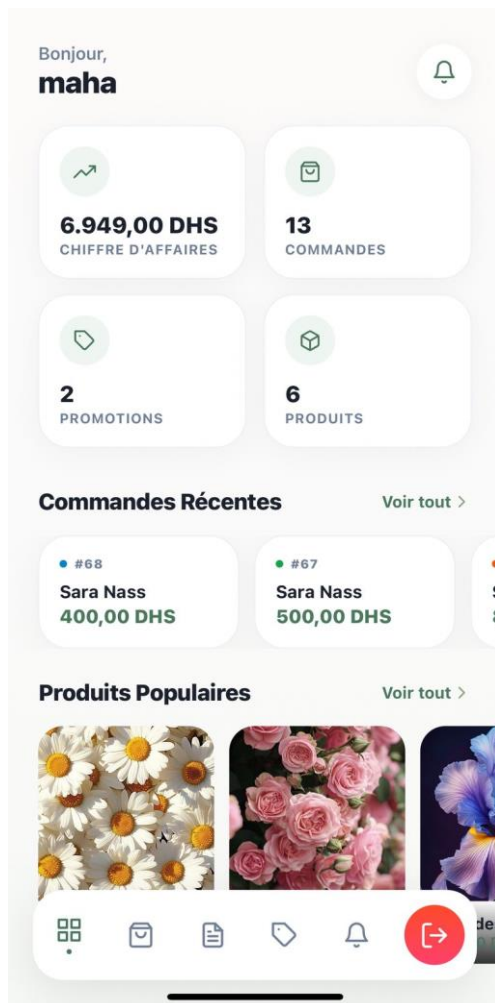


### 3.3 Dashboard prestataire

#### 3.3.1 Tableau de bord du prestataire

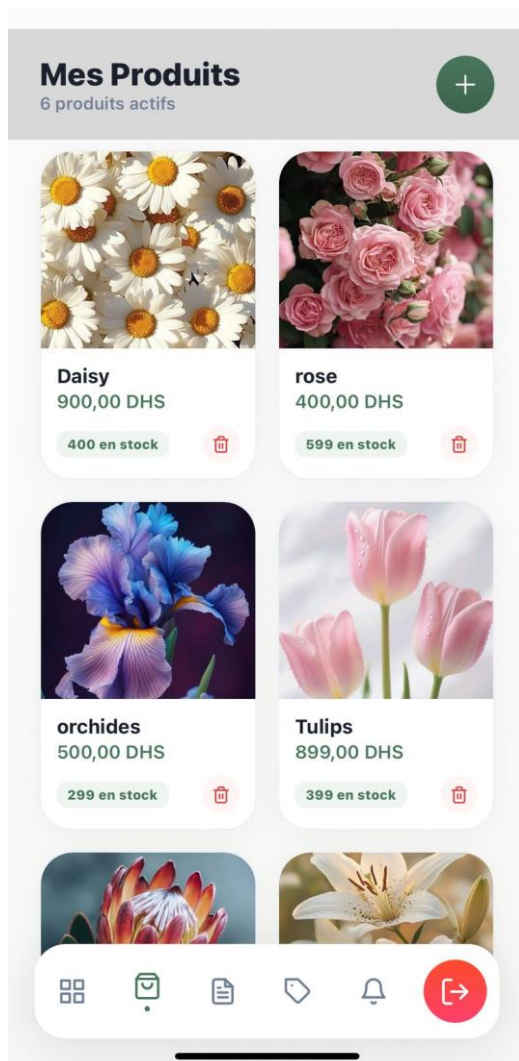
Cette interface constitue la page d'accueil du prestataire après authentification. Elle présente des indicateurs clés tels que le chiffre d'affaires réalisé, le nombre total de commandes, le nombre de promotions actives et le nombre de produits disponibles. Ces informations offrent au prestataire une vue synthétique de l'activité de sa boutique et facilitent la prise de décision rapide.





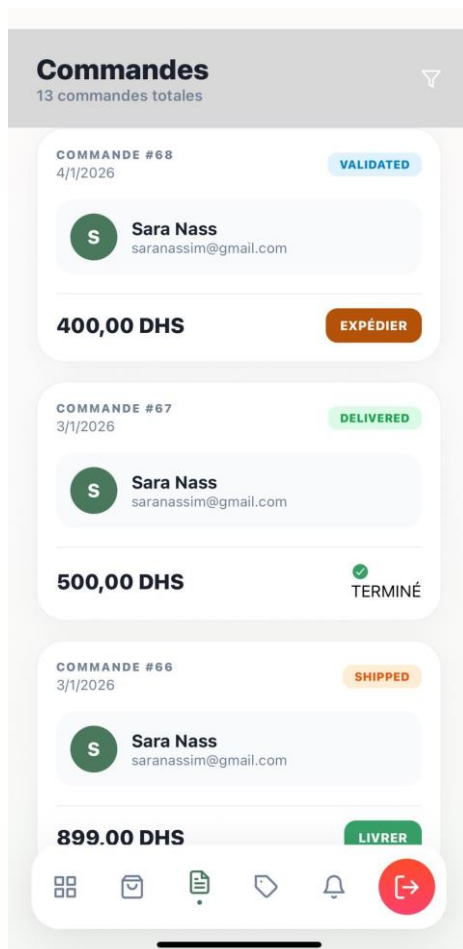
### 3.3.2 Gestion des produits

Cette page permet au prestataire de consulter et gérer l'ensemble de ses produits. Chaque produit est affiché avec son image, son nom, son prix et la quantité disponible en stock. Des actions telles que la suppression ou la modification d'un produit sont disponibles, ce qui garantit une gestion efficace et à jour du catalogue proposé aux clients.



### 3.3.3 Gestion des commandes

Cette page présente l'ensemble des commandes reçues par le prestataire. Chaque commande inclut les informations du client, le montant total et le statut de la commande (validée, expédiée, livrée ou terminée). Le prestataire peut mettre à jour le statut afin d'assurer un suivi précis du processus de livraison.



#### 3.3.4 Liste des promotions

Cette interface affiche toutes les promotions actives du prestataire. Chaque promotion est présentée avec le produit concerné, le taux de réduction appliqué et la date de fin de validité. Elle permet au prestataire de suivre et de gérer efficacement ses offres promotionnelles.




### 3.3.5 Ajout d'un nouveau produit

Cette interface est dédiée à la création d'un nouveau produit. Le prestataire peut renseigner les informations essentielles telles que le nom du produit, le prix, la quantité en stock, la catégorie ainsi qu'une description détaillée. Il est également possible d'ajouter une image afin d'améliorer la visibilité du produit sur la plateforme.

01:43

< **Nouveau Produit**

VISUEL DU PRODUIT



**Ajouter une photo haute qualité**  
JPG ou PNG, max 5MB

INFORMATIONS DE BASE

**Nom du produit**

Ex: Roses Rouges Premium

**Prix (DHS)** **Stock**

0.00 0

**Catégorie**

Fleurs  
Plantes  
Bouquets

### 3.3.6 Création d'une promotion

Cette page permet au prestataire de lancer une nouvelle promotion sur un produit sélectionné. Il peut définir un titre de promotion, un pourcentage de remise ainsi qu'une période de validité. Cette fonctionnalité favorise l'augmentation des ventes et la mise en avant des produits auprès des clients.

Nouvelle Promotion

SÉLECTION DU PRODUIT

Daisy (900 DH)

rose (400 DH)

orchides (500 DH)

DÉTAILS DE L'OFFRE

Titre de la promotion

Ex: Soldes d'Automne

Remise (%)

Ex: 20

Période de validité

DÉBUT

4/1/2026

FIN

11/1/2026

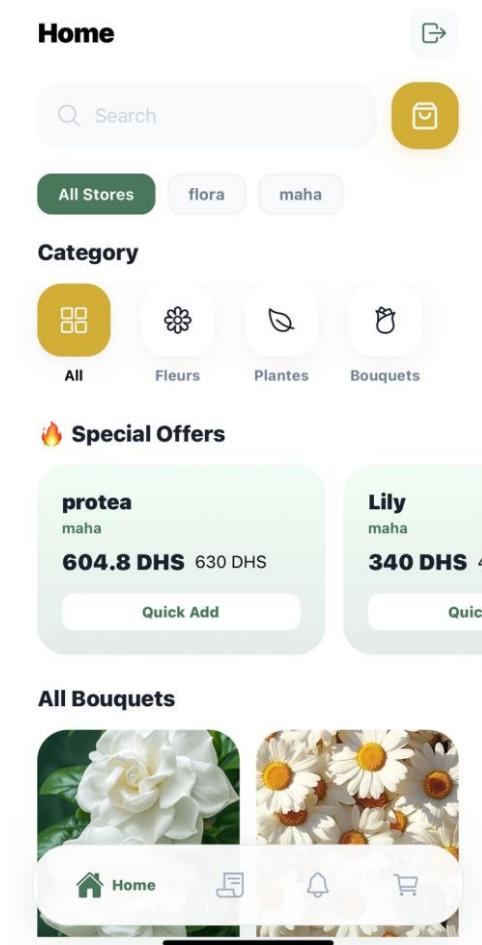
Description (Optionnel)

Conditions ou détails supplémentaires...

## 3.4 Interface Client

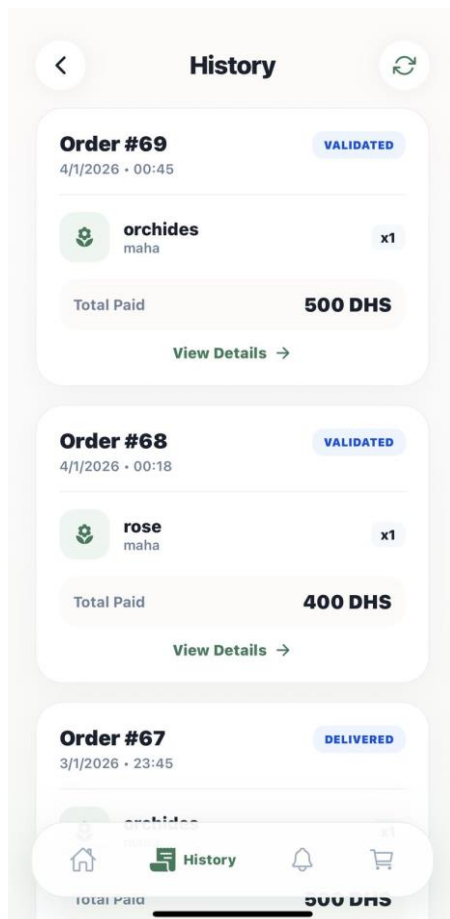
### 3.4.1 Page d'accueil (Home – Client)

Cette interface constitue le point d'entrée principal de l'application pour le client. Elle permet de rechercher des produits, filtrer par boutique (flora, maha) ou par catégorie (fleurs, plantes, bouquets). Les offres spéciales sont mises en avant afin d'attirer l'attention de l'utilisateur et faciliter l'accès rapide aux promotions et produits populaires.



#### 3.4.2 Historique des commandes

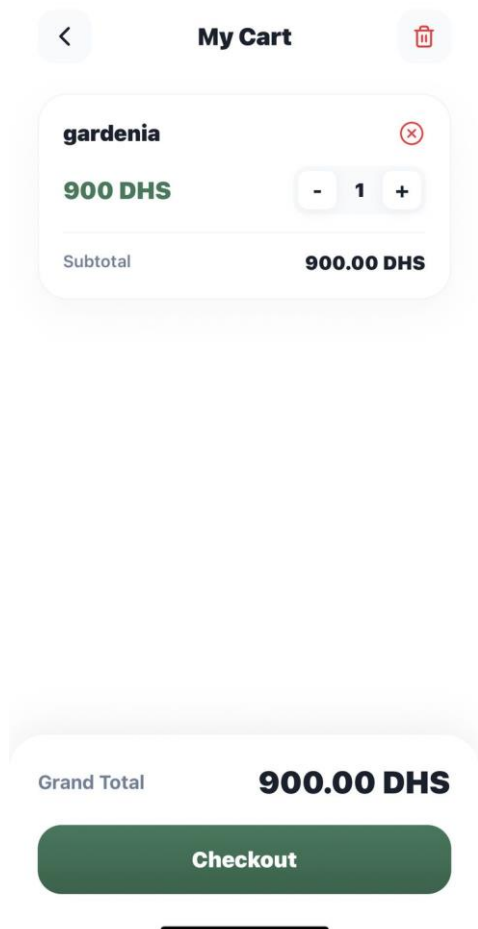
La page d'historique permet au client de consulter toutes ses commandes passées avec leur statut (validée, livrée, etc.). Elle offre une vision claire du suivi des achats et permet d'accéder aux détails de chaque commande.



### 3.4.3 Panier

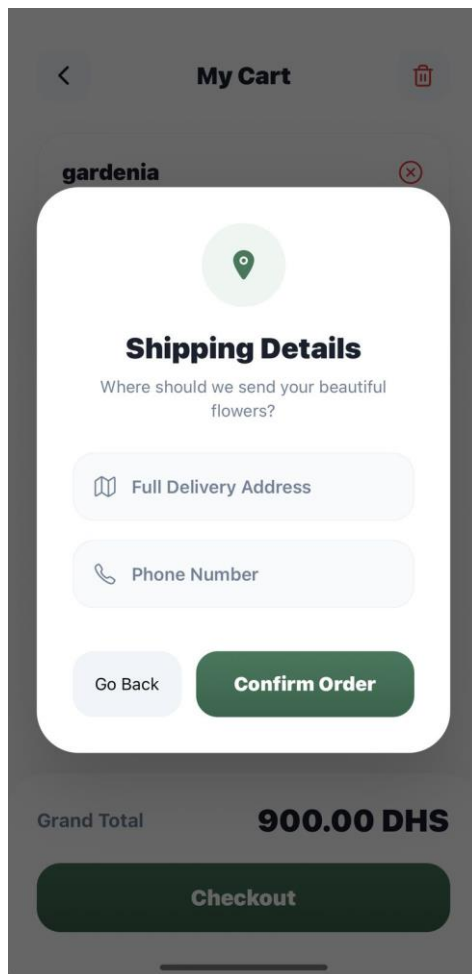
La page panier affiche les produits sélectionnés par le client avec leur prix et la quantité choisie. L'utilisateur peut ajuster la quantité, supprimer un article ou consulter le total de la commande en temps réel. Cette interface joue un rôle clé dans la préparation de la commande avant le passage au paiement.





#### 3.4.4 Détails de livraison

Cet écran permet au client de saisir les informations nécessaires à la livraison, notamment l'adresse complète et le numéro de téléphone. Ces données sont essentielles pour assurer une livraison correcte et rapide. Une confirmation est requise avant de poursuivre le processus de commande.



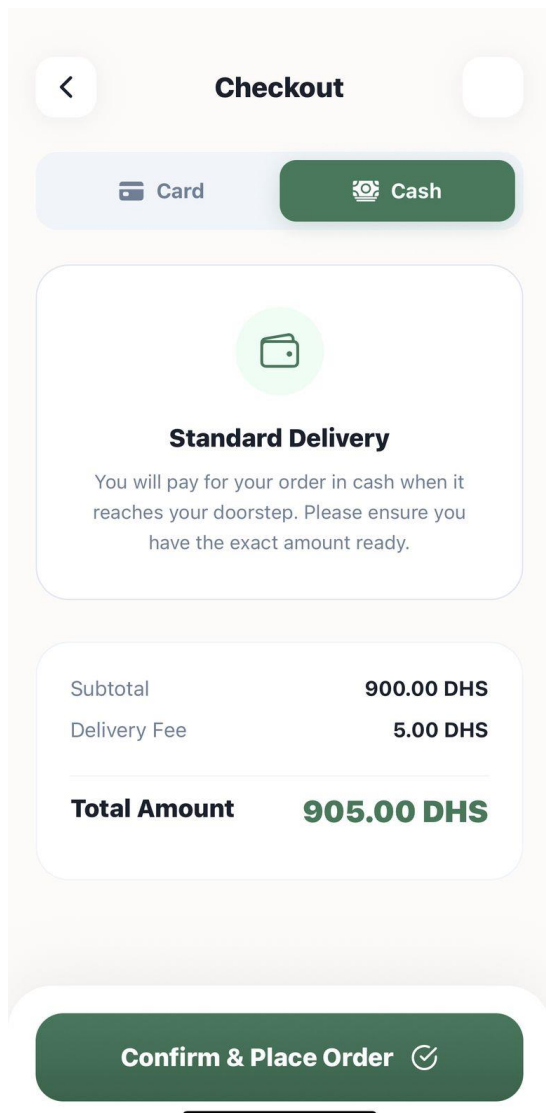
#### 3.4.5 Choix du mode de paiement – Carte bancaire

Cet écran permet au client de payer par carte bancaire en renseignant les informations requises (numéro de carte, titulaire, date d'expiration et CVV). Il offre une solution de paiement moderne et sécurisée, adaptée aux utilisateurs préférant le paiement électronique.

The image shows a mobile application interface for a checkout process. At the top, there is a navigation bar with a back arrow on the left, the title "Checkout" in the center, and a white square icon on the right. Below the navigation bar, there are two buttons: "Card" (highlighted in green) and "Cash" (light blue). Under the "Card" button, there is a dark blue card template. The card template includes a yellow chip icon on the left, a red and orange circular logo on the right, a series of ten dots in the center, and the text "CARD HOLDER NAME SURNAME" and "EXPIRES MM/YY" at the bottom. Below the card template, there are four input fields: "Card Number" (placeholder: 0000 0000 0000 0000), "Cardholder Name" (placeholder: Enter full name), "Expiry Date" (placeholder: MM/YY), and "CVV" (placeholder: 123). At the bottom of the screen, there is a large green button labeled "Authorize Payment" with a white checkmark icon.

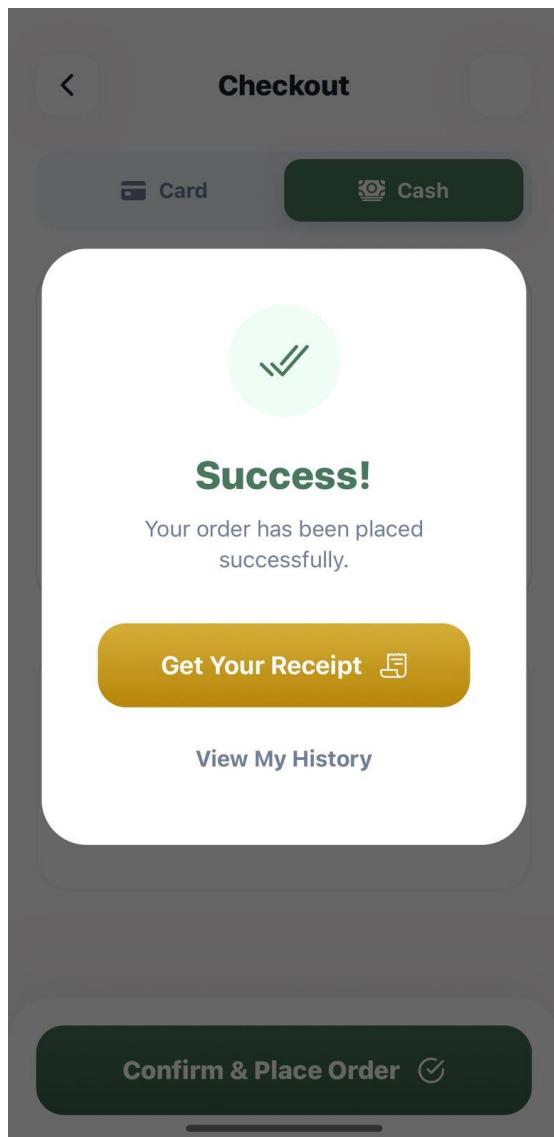
#### 3.4.6 Choix du mode de paiement – Espèces (Cash)

Cette interface présente l'option de paiement en espèces à la livraison. Elle informe clairement le client qu'il devra régler le montant total lors de la réception de la commande et l'invite à préparer le montant exact, garantissant ainsi une expérience simple et accessible.



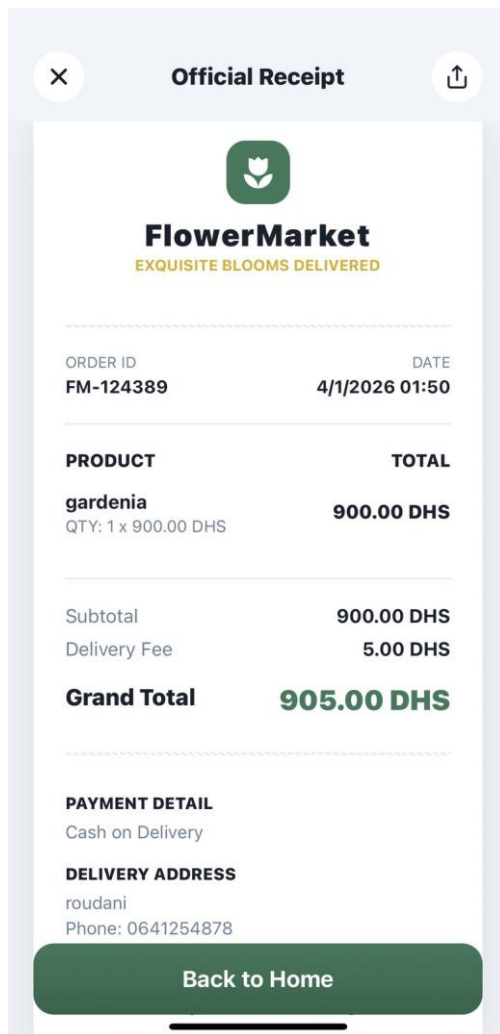
#### 3.4.7 Confirmation de commande

Après validation du paiement, cet écran confirme que la commande a été effectuée avec succès. Il rassure l'utilisateur et propose des actions supplémentaires telles que consulter l'historique des commandes ou télécharger le reçu.



#### 3.4.8 Reçu officiel

Cette interface affiche le reçu détaillé de la commande, comprenant l'identifiant, la date, les produits achetés, le montant total, le mode de paiement et l'adresse de livraison. Elle constitue une preuve officielle de l'achat pour le client.



## 4. Conclusion

Ce chapitre a présenté en détail la phase de réalisation de l'application *FlowerMarket*, en mettant l'accent sur les outils et technologies utilisés ainsi que sur la conception des interfaces utilisateur. Les choix technologiques adoptés, notamment **Expo**, **React Native** et les services backend, ont permis de développer une application performante, intuitive et multiplateforme.

La description des interfaces a mis en évidence une organisation claire et cohérente des fonctionnalités, aussi bien pour les prestataires que pour les clients. Chaque écran a été conçu dans le but de simplifier l'expérience utilisateur, depuis la consultation des produits jusqu'au processus de commande, de paiement et de suivi.

Ainsi, cette phase de réalisation a permis de concrétiser les objectifs fonctionnels définis dans les chapitres précédents, tout en garantissant une application ergonomique, fiable et adaptée aux besoins des utilisateurs. Le chapitre suivant sera consacré à l'évaluation de l'application, aux difficultés rencontrées et aux perspectives d'amélioration

## Conclusion et perspectives

Ce projet a permis de concevoir et de réaliser une application mobile de type marketplace dédiée à la vente de produits floraux, offrant une solution complète et intuitive pour les clients, les prestataires et l'administrateur. Grâce à une démarche méthodologique structurée et à l'utilisation de technologies modernes telles que React Native avec Expo, l'application *FlowerMarket* propose des fonctionnalités essentielles incluant la gestion des produits, des promotions, des commandes, des paiements et du suivi des achats, tout en garantissant une expérience utilisateur fluide et ergonomique. Les objectifs fixés au départ ont été atteints, tant sur le plan fonctionnel que technique, et le système mis en place constitue une base solide et évolutive. Toutefois, plusieurs perspectives d'amélioration peuvent être envisagées, notamment l'intégration de services de paiement en ligne sécurisés, l'ajout de notifications en temps réel, le suivi de livraison et l'enrichissement des fonctionnalités analytiques et de recommandation, afin d'optimiser davantage la performance, la sécurité et la portée de l'application.

## Webographie

Documentation .NET – <https://learn.microsoft.com/fr-fr/dotnet>

Documentation SQL Server – <https://learn.microsoft.com/fr-fr/sql>

Méthodologie Agile Scrum (Scrum.org) –  
<https://www.scrum.org/resources/what-is-scrum>

GitHub Documentation – <https://docs.github.com>

Visual Studio Code – <https://code.visualstudio.com>

Visual Studio – <https://visualstudio.microsoft.com>

UML Resource (Object Management Group) – --  
<https://www.omg.org/spec/UML>

React Native Documentation  
<https://reactnative.dev/docs/getting-started>

Expo Documentation  
<https://docs.expo.dev>

Firebase Documentation  
<https://firebase.google.com/docs>

Firebase Authentication  
<https://firebase.google.com/docs/auth>

Taiga Project Management Tool  
<https://taiga.io>