

Université Mouloud MAMMERI de Tizi-Ouzou  
Faculté DE Génie Électrique et d'Informatique  
Département d'Informatique.



## **Mémoire de fin d'étude**

*En vue d'obtention du Diplôme de MASTER 2 en Informatique  
Option: Ingénierie des Systèmes d'Information.*

### **Thème:**

*Mise en place d'une solution de paiement mobile  
pour une entreprise commerciale*

*Présenté par*

- ❖ HADJ SAID Nadine.
- ❖ REKOUANE Hichem.

*Sous la direction du professeur*  
SI MOHAMMED.Malik

Année universitaire: 2019/2020.

# Résumé

L'informatique devient de jour en jour un support incontestable de la quasi-totalité des domaines, en offrant des services capables d'améliorer et de faciliter la vie quotidienne.

Le thème de ce projet de fin d'études est la conception et la réalisation d'une solution de paiement mobile destiné aux entreprises commerciales.

Pour ce faire, nous avons fait une étude sur les applications mobiles, leur apparition, leurs différents types ainsi que leurs systèmes d'exploitation et environnements de développement. Nous avons aussi étudié le paiement mobile et ses différents types. Ensuite, nous avons accompli une démarche d'analyse et de conception afin de nous assurer que notre solution informatique réponde aux besoins des clients. Enfin, nous avons étudié les différents outils et langages de programmation qui nous permettent d'atteindre l'objectif voulu. Nous en avons choisi le framework de développement d'application mobile open-source React Native ainsi que le framework Laravel et d'autres outils cités et définis plus loin.

Nous avons implémenté deux applications mobiles natives pour la plateforme Android qui sont:

- ❖ Une application pour le vendeur de l'entreprise.
- ❖ Une application de portefeuille mobile pour son client qui permet à ce dernier de scanner le Qr code généré dans l'application du vendeur afin d'effectuer son paiement.

# *Dédicaces*

*Je dédie ce travail*

*À mes très chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, et leur soutien tout au long de mes études,*

*À mes chères sœurs Dyhia et Maylis pour leurs encouragements permanents, et leur soutien moral,*

*À toute ma famille pour leur soutien tout au long de mon parcours universitaire,*

*À ma très chère amie Yasmina qui m'a toujours soutenue tout au long de mon parcours.*

*À mon binôme Hichem qui m'a toujours compris, soutenu, et me pousse à aller de l'avant et donner le meilleurs de moi même, qui a cru en moi et m'a fait confiance.*

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible. Merci d'être toujours là pour moi.*

*Nadine.H*

# *Dédicaces*

*Je dédie ce travail*

*À mes très chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, et leur soutien tout au long de mes études,*

*À toute ma famille pour leur soutien tout au long de mon parcours,*

*À tous mes amis, pour leur soutien et pour les bons moments et les divers aventures qu'on a eu ensemble durant mon parcours à l'université,*

*À ma binôme Nadine, qui a toujours cru en moi, et m'a soutenu, poussé à me dépasser et sans qui ce travail n'aurait jamais été aussi soigné.*

*À la mémoire de mes deux très chères grand-mères, qui m'aimaient beaucoup et qui auraient aimé voir ce jour.*

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible. Merci d'être toujours là pour moi.*

*Hichem. R*

# *Remerciements*

*Nous adressons nos remerciements tout particulièrement :*

*À nos très chers parents, notre fierté et bien sur la source de notre réussite pour leurs sacrifices et encouragement. Que dieux nous les garde et les protège.*

*À notre Encadreur le professeur SI MOHAMMED.M pour son professionnalisme, ses remarques pertinentes, son soutiens et sa motivation, nous le remercions infiniment de nous avoir fait confiance et d'avoir autant cru en nous et pour chaque petit détail qu'il nous a appris tout au long de notre cursus.*

*Aux membres du jury qui ont accepté d'évaluer notre travail.*

*À nos très chers frères, soeurs, ami (es) et à toutes les personnes qui ont contribué de loin ou de près à la réussite de ce travail.*

*Hichem.R, Nadine.H*

# Table des matières

<b>Table des figures</b>	<b>9</b>
<b>Liste des tableaux</b>	<b>10</b>
<b>Introduction générale</b>	<b>11</b>
<b>Chapitre I: Les applications mobiles.</b>	<b>12</b>
<b>Introduction</b>	<b>12</b>
<b>1. Définition</b>	<b>12</b>
<b>2. Historique</b>	<b>12</b>
<b>3. Les types d'applications mobiles</b>	<b>14</b>
3.1 Les applications mobiles web	14
3.1.1 Présentation	14
3.1.2 Fonctionnement	15
3.1.3 Avantages	16
3.1.4 Inconvénients	16
3.2 Les applications natives	16
3.2.1 Présentation des applications natives	16
3.2.2 Fonctionnement	16
3.2.3 Avantages	17
3.2.4 Inconvénients	17
3.3 Les applications hybrides	17
3.3.1. Présentation	17
3.3.2. Fonctionnement	18
3.3.3 Avantages	18
3.3.4 Inconvénients	19
<b>4. Plateformes d'applications mobiles</b>	<b>19</b>
4.1. Définition	19
4.2 Les principales plateformes d'applications mobiles	19
<b>Conclusion</b>	<b>20</b>
<b>Chapitre II: Le paiement mobile.</b>	<b>21</b>
<b>Introduction</b>	<b>21</b>
<b>1. Le paiement électronique</b>	<b>21</b>
1.1. Définition	21
1.2. Fonctionnement	23
<b>2. Le paiement mobile</b>	<b>24</b>

2.1. Définition	24
2.2. Portefeuille mobile	24
2.2.1. Définition	24
2.2.2. Utilisation	24
2.3. Evolution du paiement mobile	25
2.4. Les types du paiement mobile	26
2.4.1. Le paiement NFC (Near Field Communication)	26
2.4.2 Paiements par transmission magnétique sécurisée	26
2.4.3. Paiements basés sur les ondes sonores	27
2.4.4. Paiements par QR code (Quick Response code)	27
2.4.5. Paiements de mobile à mobile	28
2.4.6. Virements bancaires mobiles	28
2.4.7. Paiements par SMS	29
2.5 Avantages et inconvénients du paiement mobile	30
2.5.1 Avantages du paiement mobile	30
2.5.2. Inconvénients du paiement mobile	31
<b>Conclusion</b>	<b>31</b>
<b>Chapitre III: Analyse et Conception.</b>	<b>32</b>
<b>Introduction</b>	<b>32</b>
<b>1. Définition du langage UML</b>	<b>32</b>
<b>2. Processus du développement de notre solution</b>	<b>33</b>
2.1. Spécification des besoins	33
2.1.1. Etablissement d'un cahier des charges	33
2.1.1.1. Définition	33
2.1.2. Délimiter le contour du système au cours d'étude	33
2.1.3. Résumer le tout dans un diagramme de contexte	35
2.1.3.1 Définition	35
2.1.3.2. Le diagramme de contexte pour notre cas d'étude	35
2.1.4. Décrire le produit à partir des différents scénarios d'utilisation	35
2.1.5. Etablissement du diagramme des cas d'utilisations	38
2.1.5.1. Définition	38
2.1.5.2. Diagramme des cas d'utilisation pour notre cas d'étude	39
2.2 Définition du Produit (analyse des besoins)	39
2.2.1. Décrire les spécifications précises du produit	40
2.2.2. Spécifications fonctionnelles et non fonctionnelles	40
2.2.2.1. Extraire les fonctionnalités du système et les interfaces utilisateur	41
2.2.2.2. Extraire les spécifications non fonctionnelles	44
2.3. La phase de la conception	45
2.3.1. Éclater le logiciel en sous-parties:	45

2.3.2. Architecture du système (composants et liens entre ces sous-parties)	45
2.3.2.1. Définition	45
2.3.2.2. Architecture globale de notre système	46
2.3.2.3. Architecture des sous-parties	46
2.3.2.3.1. L'architecture de l'application côté vendeur	47
2.3.2.3.2. L'architecture de l'application côté client	47
2.3.2.4. Les vues (structurelles) d'une architecture logicielle	48
2.3.2.4.1. La vue logique	48
Diagrammes de classes pour notre cas d'étude	49
Diagrammes de séquences	51
2.3.2.4.2 La vue d'implémentation	52
2.3.2.4.3. La vue de déploiement	52
<b>Conclusion</b>	<b>53</b>
<b>Chapitre 4: Implémentation et réalisation.</b>	<b>54</b>
<b>Introduction</b>	<b>54</b>
<b>1. Outils et langages de développement</b>	<b>54</b>
1.1. Le framework React Native	54
1.2. Le framework Laravel	55
1.3. Postman	56
1.4. MYSQL	56
1.5. HeidiSQL	57
1.6. Git	57
1.7. Visual Studio Code	57
1.8. Pusher	58
1.9. Heroku	58
1.10. L'intervention de ces outils dans notre travail	58
<b>2. Présentation de nos applications</b>	<b>59</b>
2.1. L'application côté vendeur de l'entreprise	59
2.1.1. Interface d'authentification	59
2.1.2. Interface d'accueil	61
2.1.3. Interface des commandes	62
2.1.4. Interface Détails des commandes	63
2.1.5. Interface options de paiement	64
2.1.6. Interface du paiement par QR code	65
2.1.7. Interface de la confirmation du paiement par code QR	66
2.1.8. Interface du paiement en liquide	67
2.1.9. Interface du paiement en chèque	68
2.1.10. Interface de l'historique des transactions	69
2.2. L'application portefeuille mobile pour le client de l'entreprise	70



2.2.1. Interface d'authentification	70
2.2.2. Interface d'accueil	71
2.2.3. Interface de détails de la carte et proposition de paiement	72
2.2.4. Interface d'ajout d'une carte	73
<b>Conclusion</b>	<b>74</b>
<b>Conclusion générale:</b>	<b>75</b>
<b>Webographie</b>	<b>76</b>

# Table des figures

- ❖ *Figure 1. Diagramme de contexte.*
- ❖ *Figure 2. Diagramme des cas d'utilisation.*
- ❖ *Figure 3: Architecture globale du système.*
- ❖ *Figure 4: Architecture de l'application vendeur.*
- ❖ *Figure 5: Architecture de l'application client.*
- ❖ *Figure 6: Diagramme de classes pour l'application vendeur.*
- ❖ *Figure 7: Diagramme de classes pour l'application client.*
- ❖ *Figure 8: diagramme de séquences du cas d'utilisation "Authentification".*
- ❖ *Figure 9: diagramme de séquences du cas d'utilisation " Paiement par Qr code".*
- ❖ *Figure 10: Interface d'authentification du vendeur.*
- ❖ *Figure 11: Interface d'accueil pour de l'application du vendeur.*
- ❖ *Figure 12: Interface des commandes de l'application vendeur*
- ❖ *Figure 13: Interface détails des commandes de l'application du vendeur.*
- ❖ *Figure 14: Interface des options de paiement de l'application vendeur.*
- ❖ *Figure 15: Interface du paiement par QR code de l'application vendeur.*
- ❖ *Figure 16: Interface de la confirmation du paiement par QR code de l'application vendeur.*
- ❖ *Figure 17: Interface du paiement en liquide de l'application vendeur.*
- ❖ *Figure 18: Interface du paiement par chèque de l'application vendeur.*
- ❖ *Figure 19: Interface de l'historique des transactions de l'application vendeur.*
- ❖ *Figure 20: Interface d'authentification du client.*
- ❖ *Figure 21: Interface d'accueil du client.*
- ❖ *Figure 22: Interface détails de la carte client.*
- ❖ *Figure 23: Interface d'ajout d'une carte.*

## Liste des tableaux

- ❖ Table 1. Historique des applications web
- ❖ Table 2. Tableau des tâches et scénarios.
- ❖ Table 3. Tableau des plateformes et type d'application.
- ❖ Table 4. Tableau des interfaces utilisateur et fonctionnalités.
- ❖ Table 5. Tableau de l'utilisation des outils dans le développement de la solution.

# Introduction générale

Les technologies de l'information et de la communication ont été la révolution la plus importante et innovante qui a marqué ces dernières décennies. En effet, loin d'être un phénomène éphémère ces technologies nous ont apportés du confort dans notre vie quotidienne par leur capacité à traiter l'information dans des délais raisonnables.

Cette révolution a permis l'émergence de la notion de la portabilité et de la mobilité qui permet un accès distant, instantanée et un flux sans interruption d'informations. En effet, cela est symbolisé par l'apparition des différents appareils de haute technologie tels que les Smartphones et les tablettes qui sont dotées de plusieurs applications pratiques qu'elles soient gratuites ou payantes.

L'objectif de notre projet présenté dans ce mémoire, est de développer une solution de paiement mobile pour une entreprise commerciale quelconque. Cela va faciliter le processus de paiement au clients et aux vendeurs de l'entreprise, puisqu'ils n'auront besoins que de leurs smartphones, et sera donc d'une grande utilité pour les entreprises commerciales en étant un moyen d'attirer et de satisfaire leurs clientèles.

Ce système sera donc constitué d'une application mobile pour le vendeur de l'entreprise et d'une seconde application de portefeuille mobile pour son client.

Pour mener à terme notre travail, le manuscrit est composé de quatre chapitres :

- ❖ **Chapitre 1 : Les applications mobiles:**

Ce chapitre présente les applications mobiles, leurs différents types et les différents systèmes d'exploitations mobiles.

- ❖ **Chapitre 2 : Le paiement mobile:**

Ce chapitre définit le paiement mobile et présentes ses différents types et leurs fonctionnement..

- ❖ **Chapitre 3 : Analyse et conception:**

Nous montrons dans ce chapitre une description générale de notre application mobile et nous expliquons aussi la phase de conception de ce projet avec ces différents diagrammes.

- ❖ **Chapitre 4 : Implémentation et réalisation:**

Nous présentons les outils et langages utilisés pour la réalisation de nos applications, et nous montrons leurs interfaces.

# Chapitre I: Les applications mobiles.

## Introduction

L'apparition des téléphones mobiles intelligents a révolutionné le monde des technologies de l'information et de la communication, notamment grâce aux développements d'applications mobiles qui a très vite submergé et a totalement changé le quotidien de la population.

Aujourd'hui, l'utilisation des applications mobiles ne cesse de croître et leur développement devient de plus en plus important, ce qui pousse plusieurs entreprises à créer leurs propres applications mobiles.

Dans ce chapitre, nous définissons les applications mobiles, et expliquons leurs différents types, leur mode de fonctionnement, et leurs différentes plateformes.

## 1. Définition [\[1\]](#)

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un smartphone, un baladeur numérique, une tablette tactile.

Elles sont pour la plupart distribuées depuis des plateformes de téléchargement (parfois elles-mêmes contrôlées par les fabricants de smartphones) telles que l'App Store (plateforme d'Apple), le Google Play (plateforme de Google / Android), ou encore le Microsoft Store (plateforme de Microsoft pour Windows 10 Mobile). Mais des applications peuvent aussi être installées sur un ordinateur, grâce par exemple au logiciel iTunes distribué par Apple pour ses appareils.

## 2. Historique

Les applications mobiles sont liées aux développements d'internet et des télécommunications des réseaux sans fils, et à l'apparition et la démocratisation des terminaux mobiles : smartphones, tablettes tactiles... Elles sont apparues dans les années

1990. [2]

L'apparition et l'évolution des applications mobiles se résument en plusieurs phases présentées dans le tableau ci-dessous:

<b>Phase 0: Avant les applications mobiles prédiction et origine: 1938-2008</b>	<b>Phase 1: Le modèle de "l'appareil d'information" 2008-2010</b>	<b>Phase 2: L'ère de "l'écran d'accueil" 2011-2013</b>	<b>Phase 3: application mobile en tant que couches de service 2014</b>
Juin 1983: Steve Jobs prédit un système de distribution de logiciels:	Septembre 2008: 100 millions de téléchargements dans l'App Store	Mars 2011: Lancement d'Amazone App Store	Janvier 2014: L'utilisation des applications mobiles à la hausse
Août 1993: Le Newton MessagePad	Octobre 2008: Le marché Android est lancé	Décembre 2011: L'App Store et la création d'emplois	
Janvier 1996: Le Palm Os	Avril 2009: Lancement du Blackberry World	Mars 2012: Android Market est renommé Google Play Store	
Décembre 1997: Le Nokia 6110	Octobre 2010: Lancement de Windows Phone Store	Mai 2013: L'App Store publie sa liste des meilleurs téléchargement.	
Octobre 1999: WAP (Wireless Application Protocol)			
Octobre 2001: iPod première génération			
Avril 2003: iTunes			

music Store			
Juin 2007: Développeurs tiers créent des programmes pour l'iPhone			
Juin 2007 : Lancement de l'iPhone			
Novembre 2007: Applications natives pour l'iPhone			
Juillet 2008: Lancement de l'App Store			

*Table 1. Historique des applications web* [2]

## 3. Les types d'applications mobiles

Il existe trois principaux types d'applications mobiles, notamment les applications mobiles basées sur le Web, les applications natives, et les applications hybrides.

### 3.1 Les applications mobiles web

#### 3.1.1 Présentation

Ce type d'applications se divisent en deux principales catégories:

- ❖ Les applications web spécialement conçues pour les plateformes mobiles:

Ces dernières ont un nom de domaine qui commence généralement par “mobile.” et il s’agit de la version mobile d’une application web, exemple : “mobile.amazon, mobile.facebook”. [18]

- ❖ Les applications Web dont l’affichage s’adapte automatiquement au format de l’écran et par extension à l’affichage mobile (applications responsives): Ces applications sont accessibles via le navigateur Internet qui est sur l’appareil mobile, elles sont donc indépendantes de la marque et du système d’exploitation de ce dernier et leur accès aux fonctionnalités est limitées à celles du navigateur, elles sont développées à partir de technologies Web et nécessitent donc une connexion internet. [18]

### 3.1.2 Fonctionnement

Les applications web sont basées sur l’architecture client-serveur qui est un modèle informatique dans lequel le serveur héberge, fournit et gère la plupart des ressources et services à consommer par le client. Ce type d’architecture a un ou plusieurs ordinateurs clients connectés à un serveur central via une connexion réseau ou Internet . [9]

Les clients correspondent aux appareils des utilisateurs connectés sur Internet (téléphone mobile connecté à internet dans ce cas) et aux logiciels d’accès au web (par exemple, les navigateurs comme Firefox ou Chrome).

Les serveurs sont des ordinateurs qui stockent des pages web, des sites ou des applications.

L’application Web nécessite un serveur Web pour gérer les demandes du client, un serveur d’applications pour effectuer les tâches demandées, et parfois, une base de données pour stocker les informations. [8]

Le flux d’une application Web typique fonctionne comme suit: [8]

- ❖ L’utilisateur déclenche une demande (requête) au serveur Web via Internet, soit via un navigateur Web ou l’interface utilisateur de l’application mobile.
- ❖ Le serveur Web transmet cette demande au serveur d’applications Web approprié.
- ❖ Le serveur d’applications Web exécute la tâche demandée (comme interroger la base de données ou traiter les données) puis génère les résultats des données demandées.
- ❖ Le serveur d’applications Web envoie les résultats au serveur Web avec les informations demandées ou les données traitées.



- ❖ Le serveur Web envoie la réponse au client avec les informations demandées qui apparaissent ensuite sur l'écran de l'utilisateur.

### 3.1.3 Avantages [\[10\]](#)

- ❖ Accessibilité via tous les terminaux mobiles indépendamment du système d'exploitation et de la marque de celui-ci.
- ❖ Développement et maintenance simples et pas coûteux.

### 3.1.4 Inconvénients [\[10\]](#)

- ❖ Nécessitent une connexion internet.
- ❖ Leur accès aux fonctionnalités est limitées à celles du navigateur.
- ❖ Manque de fluidité sur les anciens modèles de smartphones.

## 3.2 Les applications natives

### 3.2.1 Présentation des applications natives

Celles-ci correspondent à des applications créées uniquement pour une plateforme mobile spécifique. Le développement de ces applications se fait au travers du SDK (software development kit) de la plateforme mobile en question. Le nom de ces applications vient du fait qu'elles sont développées exclusivement avec le langage « natifs », par exemple le langage JAVA ou le langage Objective-C. Elles sont téléchargées à partir d'une plateforme de téléchargement qui est souvent un Store applicatif. C'est par exemple le cas pour l'Apple store ou encore Google Play. [\[19\]](#)

### 3.2.2 Fonctionnement

Les applications mobiles natives sont écrites dans un langage natif, c'est à dire un langage qui sera compilé pour la plateforme cible, elles utilisent les diverses APIs (Application Programming Interface qui sont des ensembles normalisés de classes, de méthodes, et de fonctions qui servent de façades qui font communiquer des services web avec des applications.) fournies par la plateforme en question pour faire communiquer ses différentes composantes avec le système d'exploitation, ce qui leurs donne accès à toutes

les fonctionnalités de l'appareil mobile. [\[10\]](#)

### 3.2.3 Avantages [\[10\]](#)

- ❖ Ergonomiques, conçues spécifiquement pour le terminal qui les supporte. Cela procure une meilleure expérience pour l'utilisateur.
- ❖ Elle permettent d'utiliser et d'intégrer toutes les fonctionnalités du téléphone (accéléromètre, gyroscope, GPS, caméra...).
- ❖ Une fois téléchargées et installées, certaines applications natives peuvent fonctionner sans connexion internet et se lancent plus rapidement que les sites web mobiles.

### 3.2.4 Inconvénients

- ❖ Elles doivent respecter les règles définies par les différentes sociétés des plateformes mobiles . Que ce soit l'approbation nécessaire des Apps Store pour diffuser l'application ou ses mises à jour ou les conditions tarifaires imposées.[\[11\]](#)
- ❖ Le coût lié au développement et à la maintenance d'une application native est plus élevé si elle est portée sur plusieurs plateformes (afin d'être disponible pour un maximum de mobinautes) que le coût d'un site web mobile. [\[10\]](#)
- ❖ Pour que l'utilisateur ait accès à la dernière version, il faut qu'il mette à jour l'application depuis le store.[\[19\]](#)

## 3.3 Les applications hybrides

### 3.3.1. Présentation

Ce sont des applications qui partagent le plus de code possible ou parfois le même code entre plusieurs plateformes. Une application dite hybride, contrairement à une application native, n'est pas dépendante d'une plateforme mobile en particulier. Et contrairement aux applications Web, elles peuvent accéder à plus de fonctionnalités présentes sur le mobile. [\[10\]](#)

### 3.3.2. Fonctionnement

Les applications mobiles hybrides fonctionnent selon deux approches ayant pour but avoir un seul code pour plusieurs plateformes: l'approche "Web views" et l'approche "High level language/ framework into native".

- ❖ L'approche "Web views" émule un navigateur et propose des APIs (Application Programming Interface) pour communiquer avec le système d'exploitation et utiliser ses fonctionnalités natives comme le fait par exemple le framework de développement mobile Apache Cordova. [\[17\]](#)
- ❖ L'approche "High level language/ framework into native": Cette approche s'appuie sur le principe d'écrire le code de l'application dans un langage de haut niveau /framework qui sera traduit en langage natif, prenons deux exemples:
  - Le framework de développement d'application mobile open-source React Native de facebook: Il permet de développer des applications pour Android et iOS en permettant aux développeurs d'utiliser la librairie javascript React qui sera traduit vers les langages natifs des plateformes précédemment citées puis compilé, en utilisant un bridge préfabriqué qui fait communiquer les APIs javascript avec les fonctionnalités natives du système d'exploitation. [\[11\]](#)
  - Le framework de développement d'application mobile open-source multiplateformes Flutter de google: Où l'application est codée en langage dart. Avec Flutter, la partie présentation de l'application est gérée par un code spécifique et n'est pas appelé avec un bridge, elle fait appel aux fonctions natives du système d'exploitation en utilisant des APIs. [\[13\]](#)

### 3.3.3 Avantages [\[10\]](#)

- ❖ Nécessitent moins de ressources pour développer et maintenir sur plusieurs plateformes.
- ❖ Elles ont accès aux fonctionnalités les plus importantes de l'appareil mobile.
- ❖ Partagent parfois l'intégralité du code pour différentes plateformes.

### 3.3.4 Inconvénients [\[10\]](#)

- ❖ Les technologies utilisées ne sont pas matures.
- ❖ Nécessitent parfois l'écriture de parties de code spécifique à certaines plateformes.

## 4. Plateformes d'applications mobiles

### 4.1. Définition

Une plateforme d'applications mobiles est une suite d'outils logiciels utilisés pour concevoir, créer et maintenir des applications mobiles. Il existe de nombreuses plateformes d'applications mobiles différentes, et toutes ces plateformes contiennent un environnement de développement intégré. Cela fournit les outils dont un développeur a besoin pour écrire, tester, déployer et gérer des applications mobiles.[\[10\]](#)

### 4.2 Les principales plateformes d'applications mobiles

Il existe plusieurs plateformes pour le développement et le déploiement des applications mobiles comme: Symbian OS, Blackberry OS, Windows Phone, Ubuntu MID Edition, Bada OS, Palm OS et WebOS dont la plupart sont aujourd'hui abandonnées. Et les plateformes les plus utilisées de nos jours sont:

#### ❖ **iOS**

iOS est le système d'exploitation mobile développé par Apple pour plusieurs de ses appareils. Il est dérivé de macOS dont il partage les fondations (le noyau hybride XNU basé sur le micro-noyau Mach, les services Unix et Cocoa, etc.). [\[20\]](#)

#### ❖ **Android**

Android est un système d'exploitation mobile fondé sur le noyau linux, développé par Google et lancé en 2007. Ce système avait d'abord été conçu pour les smartphones et tablettes tactiles, puis s'est diversifié dans les objets connectés et ordinateurs comme les téléviseurs

(Android TV, les voitures (Android Auto), les ChromeBook (Chrome OS qui utilise les applications Android) et les smartwatch (Wear OS). Cette plateforme met à disposition un kit de développement (SDK) basé sur le langage Java. elle possède donc une bibliothèque de plusieurs classes java servant de base pour plusieurs type d'application (exemple :SSL pour les protocoles de sécurité). [\[21\]](#)

## Conclusion

Les applications mobiles font aujourd'hui partie de notre quotidien et les entreprises se précipitent à en développer d'autant plus qu'il est plus facile de développer et maintenir une meme applications sur plusieurs plateforme grâce à l'émergence des outils de développement multi plateformes.

Nous avons présenté dans ce chapitre les applications mobiles, leurs modes de fonctionnement, leurs types ainsi que les différentes plateformes mobiles en identifiant leurs forces et leurs faiblesses pour enfin déterminer la plateforme qui conviendra le mieux à notre cas d'étude.

Dans le chapitre suivant, nous définirons et nous détaillerons le paiement mobile, son fonctionnement et ses différents types.

# Chapitre II: Le paiement mobile.

## Introduction

Aujourd'hui, il existe de nombreuses possibilités pour payer ses biens et services. Parmi elles, il y a le paiement mobile, aussi connu sous l'appellation m-paiement, une tendance pratique et intéressante qui permet de régler son paiement seulement avec son téléphone mobile.

Nous définissons donc dans ce chapitre, ce type de paiement, son utilité ainsi que ses différents types.

## 1. Le paiement électronique

### 1.1. Définition [\[3\]](#)

C'est un moyen de payer un bien ou un service par voie électronique, au lieu d'utiliser de l'argent liquide ou un chèque, en personne ou par courrier. Il comprend toutes les opérations financières utilisant des appareils électroniques, tels que des ordinateurs, des smartphones ou des tablettes.

Aujourd'hui, de nombreux utilisateurs effectuent des paiements par voie électronique plutôt qu'en liquide ou en chèques. Des centaines de systèmes de paiement électronique ont été développés pour fournir des transactions Internet sécurisées. Les systèmes de paiement électronique sont généralement classés en 3 catégories:

- ❖ **Cartes de crédit et cartes de débit [\[49\]](#):** Un paiement par carte Crédit/Débit est un paiement qui se fait de manière digitale, mettant en accord 4 partis distincts :
  - Le porteur ou propriétaire de la carte Crédit/Débit.,
  - Le commerçant ou locataire/propriétaire du lecteur de carte.
  - La banque du commerçant.,
  - La banque du porteur.

Le montant autorisé des achats payés par carte bancaire est défini à l'avance entre le propriétaire de la carte bancaire et la banque.

Les cartes de débit sont des cartes délivrées par la banque à l'ouverture d'un compte dont les dépenses sont prélevées au fur et à mesure sur ce dernier, Elles présentent soit le logo CB, ce qui indique qu'elles sont un moyen de paiement national, soit les logos VISA ou MASTERCARD, ce qui indique qu'elles permettent les paiements internationaux.

Les cartes de crédit sont également liées à un compte courant mais à la différence des cartes débit, les dépenses ne sont prélevées qu'en une seule fois, le dernier jour du mois ouvré.

❖ **Espèces électroniques:** Transfert électronique de fonds (TEF) : c'est un système informatisé utilisé pour faire des transferts d'argent entre des comptes bancaires.

Le TEF comprend de nombreux autres concepts :

- Prélèvement bancaire : c'est une instruction donnée par le titulaire du compte à sa banque pour transférer une certaine somme d'argent provenant de son compte courant.
- Chèque électronique : c'est un transfert électronique de fonds dans lequel l'argent provient d'un compte bancaire, généralement un compte chèques.
- Facturation électronique : c'est la pratique par laquelle les entreprises prélèvent des paiements en utilisant des méthodes électroniques.
- Monnaie électronique : également connue sous le nom d'argent numérique, elle désigne le stockage des valeurs monétaires en quantité modérée sur un dispositif appartenant au client (en général, le PC de l'utilisateur) et est utilisée pour effectuer des transactions sur Internet.
- Programmes de fidélité E-commerce : ce sont les programmes qui visent à encourager la fidélité des clients, en offrant des récompenses. Il ya beaucoup de stratégies différentes pour augmenter la fidélité des clients, comme les cartes à points, des tombolas, les remises, les prix, les mises à jour, les cadeaux, etc.
- Carte prépayée : c'est une variante de la carte de crédit. Elle dispose d'un montant fixe d'argent codé. Elle est normalement anonyme et a une date d'expiration. L'exemple le plus courant de cartes prépayées sont les cartes-cadeaux. [\[52\]](#)

- ❖ **Systèmes de micropaiement:** Le micropaiement est un mécanisme de paiement qui permet de procéder à l'achat de biens ou de services de faible montant. Ce service de paiement est fréquemment utilisé par les éditeurs de sites internet. Différentes solutions de micropaiement existent : les numéros de téléphone surtaxés, l'envoi de SMS surtaxés, PayPal, la facturation sur la facture du fournisseur d'accès à Internet ou de l'opérateur mobile, ou encore les cartes prépayées. L'utilisation du micropaiement présente l'avantage pour les commerçants de ne pas avoir recours à des solutions de paiement plus coûteuses telles que la carte bancaire. Il permet également aux cybermarchands de gérer avec simplicité leur module de paiement sur leur site internet. Le micropaiement offre aussi des avantages pour les consommateurs. En un simple clic, le client peut acheter un bien ou un service sans avoir à créer un compte personnel ni à fournir des données personnelles. Il s'agit d'un moyen de paiement sécurisé et très utilisé notamment sur les sites de jeux en ligne. [\[50\]](#)

## 1.2. Fonctionnement [\[3\]](#)

Les paiements en ligne sont effectués instantanément, ce qui est pratique et fait gagner beaucoup de temps et ils fonctionnent comme suit:

- ❖ **Du côté du client:** Le processus commence lorsqu'un client visite le site du marchand et ajoute au panier les articles (produits ou services) qu'il souhaite acheter. Il doit ensuite remplir le formulaire de paiement avec certaines informations (par exemple, numéro de carte, date d'expiration, code CVV, adresse). Selon le mode de paiement, le client est redirigé vers un service externe ou le site Web de la banque ou continue le paiement sur le site Web ou dans une application.
- ❖ **Authentification du paiement par l'opérateur:** La passerelle de paiement (avec d'autres parties impliquées) vérifie si les informations de paiement sont valides, dans ce cas, le processus se poursuit et la passerelle de paiement signale la transaction réussie et le client reçoit une confirmation de paiement en temps réel.



- ❖ **Du côté du vendeur:** Un fournisseur de paiement en ligne reçoit un paiement de la banque d'un client et le transfère sur le compte du marchand.

## 2. Le paiement mobile

### 2.1. Définition

Les paiements mobiles sont toutes les transactions effectuées depuis un téléphone mobile et débitées soit sur une carte bancaire soit sur la facture opérateur, soit sur un Porte monnaie électronique (défini ci-dessous), qui peut être alimenté avec un dépôt de cash auprès d'un agent ou d'un commerçant. Au lieu de payer par carte de crédit ou bien en argent comptant, un consommateur peut utiliser un téléphone mobile intelligent pour se procurer des biens et des services. [22]

### 2.2. Portefeuille mobile

#### 2.2.1. Définition

Un portefeuille mobile est une application pour appareil intelligent qui stocke les détails de la carte de crédit ou de débit de l'utilisateur afin que les consommateurs puissent payer leurs transactions en utilisant leur appareil intelligent (smartphone ou tablette...). Les portefeuilles mobiles populaires incluent Apple Pay, Google Pay et Samsung Pay [14].

#### 2.2.2. Utilisation

Pour utiliser un portefeuille mobile, l'utilisateur télécharge une application de portefeuille mobile sur son appareil, puis ajoute les détails de sa carte. Les portefeuilles mobiles stockent les données sensibles des cartes en utilisant un processus de remplacement des données sensibles par des symboles d'identification uniques qui conservent toutes les informations essentielles sur les données sans compromettre leur sécurité, appelé la tokenization, garantissant ainsi une sécurité de paiement accrue pour le consommateur et le commerçant. Le numéro de carte de

crédit à 16 chiffres est remplacé par un identifiant alphanumérique généré aléatoirement, qui constitue le jeton.

Pour effectuer un paiement mobile avec un portefeuille mobile, l'utilisateur rapproche son téléphone du terminal de point de vente (POS). Ce point de vente doit être activé avec la technologie NFC ou scanner un de QR code par exemple (que nous détaillerons juste après). [\[14\]](#)

## 2.3. Evolution du paiement mobile [\[15\]](#)

En 1983, l'ingénieur électricien Charles Walton a reçu le premier brevet pour un appareil RFID (identification par radiofréquence), la technologie sur laquelle la plupart des paiements mobiles sont basés.

En 1997, Coca Cola a introduit les paiements mobiles via un nombre limité de distributeurs automatiques. Les clients devaient envoyer un SMS et acheter leurs boissons. La même année, Mobil, la compagnie pétrolière américaine, a introduit le Speedpass, qui a permis aux clients de payer le carburant dans leurs stations-service à l'aide d'un porte-clés pré-chargé en espèces. Quelques années plus tard, Mobil et Exxon ont fusionné et plus de 6 millions de clients américains se sont inscrits pour utiliser Speedpass dans les stations-service participantes.

En 2001, McDonald's a expérimenté une variété de paiements sans contact, y compris Speedpass et une petite baguette en plastique que les clients agitaient à la caisse pour payer leur repas.

En 2011, Barclay's s'est associé à Orange et a lancé le premier paiement mobile sans contact en Europe. Le système a permis aux clients de recharger leurs téléphones jusqu'à 100 £ et de payer des montants allant jusqu'à 15 £ en utilisant leurs téléphones. La même année, Google a lancé Google Wallet, permettant aux utilisateurs d'Android de payer avec leur téléphone portable chez certains détaillants aux États-Unis.

En 2013, Apple Pay a été lancé.

En 2019, 32% des utilisateurs de smartphones utilisent le paiement mobile en Australie, 47% en Belgique et 65% en Turquie [\[4\]](#).

## 2.4. Les types du paiement mobile

Diverses méthodes ont été développées pour permettre aux consommateurs de payer facilement via un téléphone comme les paiement NFC, les Paiements par transmission magnétique sécurisée (MST), les Paiements basés sur les ondes sonores, les Paiements par QR code (Quick Response code), les Paiements de mobile à mobile, les Virements bancaires mobiles, les Paiements par SMS et bien d'autres.

### 2.4.1. Le paiement NFC (Near Field Communication)

Les paiements NFC permettent l'utilisation de portefeuilles mobiles pour des paiements sans contact. Le NFC permet à deux appareils, à savoir un lecteur de carte compatible NFC et un téléphone portable, de communiquer sans fil lorsqu'ils sont proches l'un de l'autre, avec une distance approximative de deux pouces ou moins. Pour que cela fonctionne, les deux appareils doivent être équipés de la puce NFC.

Les paiements mobiles NFC fonctionnent comme suit:

Un appareil alimenté, tel qu'un lecteur de carte, peut lire et écrire sur une puce NFC. Le lecteur de carte lit la carte de crédit compatible NFC et soustrait l'argent du solde inscrit sur cette carte. [\[15\]](#)

### 2.4.2 Paiements par transmission magnétique sécurisée

La transmission magnétique sécurisée (MST) est assez similaire à la NFC. Les paiements MST génèrent des signaux "magnétiques" pour établir une connexion entre l'appareil mobile de l'utilisateur et le terminal de point de vente. Ce signal est similaire à la connexion magnétique se crée lorsque les cartes de crédit traditionnelles sont glissées dans le terminal de point de vente. La technologie MST peut fonctionner avec la plupart des terminaux compatibles NFC qui acceptent les paiements sans contact. [\[15\]](#)

### 2.4.3. Paiements basés sur les ondes sonores

Les paiements mobiles basés sur des ondes sonores (ou basés sur des signaux sonores) permettent des paiements mobiles sans contact via des ondes sonores.

Les transactions sont traitées via des ondes sonores uniques contenant des données de paiement cryptées. Le terminal envoie des ondes sonores à l'appareil mobile pour transmettre en toute sécurité les détails du paiement. Le téléphone de l'utilisateur convertit ensuite ces données en signaux continus en amplitude et en cours du temps dits analogiques qui complètent la transaction.

La technologie de paiement basée sur le son est assez sécurisée car elle utilise la tokenisation parmi d'autres méthodes de cryptage, comme l'authentification par code PIN / mot de passe ou biométrie. [\[15\]](#) Exemple: solution de paiement mobile par ondes sonores de CopSonic. [\[54\]](#)

### 2.4.4. Paiements par QR code (Quick Response code) [\[15\]](#)

Le QR code est un type de code barre matriciel créé en 1994 pour l'industrie automobile japonaise. Depuis, il a été utilisé dans de nombreux contextes, y compris comme moyen de paiement mobile particulièrement populaire en Chine.

Lorsqu'un code QR est scanné, les motifs horizontaux et verticaux de la matrice sont décodés par le logiciel sur le smartphone de l'utilisateur. Ils sont ensuite convertis en une chaîne de caractères lorsqu'ils sont capturés avec l'appareil photo du téléphone. Une fois les informations traitées, le téléphone peut ouvrir un lien de navigateur, confirmer les informations de paiement et vérifier la géolocalisation, entre autres opérations.

Les paiements QR peuvent être traités des manières suivantes:

#### ❖ **L'utilisateur scanne le code QR du commerçant:**

L'utilisateur scanne le code QR sur le smartphone du vendeur avec une application dans son smartphone permettant les

paiements par code QR. Ils confirment le prix, si nécessaire, avant de taper pour finaliser le paiement.

❖ **Le commerçant scanne le code QR du client:**

lorsque le montant total de la transaction est défini par le commerçant, le client ouvre l'application permettant les paiements par code QR. L'application affiche un code QR unique identifiant les détails de la carte de l'utilisateur. Le commerçant scanne ce code avec un scanner de code QR, finalisant la transaction.

#### 2.4.5. Paiements de mobile à mobile

Les paiements de mobile à mobile permettent des paiements rapides et sécurisés à l'aide d'un numéro de mobile. Les consommateurs peuvent s'inscrire à des services de mobile à mobile et peuvent ensuite envoyer et accepter des paiements, tant que les deux parties sont abonnées au même service. [\[15\]](#)

#### 2.4.6. Virements bancaires mobiles

La banque mobile est un service fourni par une banque, permettant à ses clients d'effectuer à distance des transactions financières, telles que des virements bancaires mobiles, à l'aide d'un appareil mobile. [\[15\]](#)

Il existe déjà des applications mobiles comme Lydia ou Pumpkin qui permettent de réaliser un virement bancaire d'un compte à un autre avec un simple numéro de téléphone portable. Moyennant deux jours, le temps que le virement s'effectue. [\[55\]](#)

Paylib a aussi lancé le transfert d'argent immédiat par téléphone. Concrètement, les utilisateurs ont accès à Paylib directement sur l'application de leur banque, et peuvent faire un virement instantané à un bénéficiaire, sans renseigner de coordonnées bancaires, en composant simplement son numéro de téléphone. [\[56\]](#)

#### 2.4.7. Paiements par SMS [7]

Les paiements par SMS (Short Messaging Service) sont l'un des moyens les plus simples et les plus sûrs de payer des articles. Il n'est pas nécessaire pour l'utilisateur de se souvenir des mots de passe ou d'avoir accès à sa carte de crédit ou à ses coordonnées bancaires; en fait, ils n'ont pas du tout besoin d'avoir un compte bancaire.

Le client paie un SMS qui comprend le coût de l'article ou du service qu'il a acheté.

Cet argent est déduit d'un mobile prépayé ou ajouté à la facture du client.

Lorsque la facture est payée, l'opérateur mobile prend sa part du prix premium SMS. L'opérateur de téléphonie mobile envoie ensuite le reste du paiement à la société de traitement des paiements SMS qui exploite la passerelle SMS pour la société auprès de laquelle le client a acheté.

La passerelle SMS est le nom du numéro auquel un client envoie un SMS pour acheter un produit ou un service.

Le fournisseur de paiement par SMS prend sa part des revenus et transfère le reste à l'entreprise qui a fourni le produit ou le service.

À aucun moment, les détails du client ne sont révélés à quiconque, sauf à l'opérateur de téléphonie mobile. C'est pourquoi les paiements par SMS sont sûrs, faciles et sécurisés.

L'utilisateur a donc différentes manières pour effectuer son paiement par SMS. Quelques méthodes sont décrites ci-dessous:

- ❖ **Via un site Internet et un code:**

Au moment du paiement, le client est renvoyé sur une page Web et reçoit un code. Il envoie ensuite ce code au numéro de texte affiché à l'écran. Il reçoit ensuite un autre SMS lui permettant d'accéder à son contenu (montant et nombres d'article etc...) ou confirmant que son article a été payé.

- ❖ **En saisissant son numéro de téléphone portable directement sur un site Internet:**

le fournisseur de paiement SMS envoie ensuite ce numéro par SMS avec un code. L'utilisateur utilise ensuite ce code pour accéder au contenu en ligne.

❖ **Via un texte direct:**

Celui-ci est principalement utilisé par les organismes de bienfaisance et également par les prestataires de services et les prestataires de transport. L'utilisateur envoie simplement un SMS au numéro annoncé et reçoit en retour son courrier ou son ticket. Ou dans le cas des organismes de bienfaisance, un merci de l'organisme de bienfaisance.

## 2.5 Avantages et inconvénients du paiement mobile

Les paiements mobiles sont un moyen de paiement très utilisés de nos jours car ils présentent de nombreux avantages, mais il comporte également quelques faiblesses.

### 2.5.1 Avantages du paiement mobile

- ❖ **Pratiques:** Les clients peuvent désormais choisir d'effectuer des paiements n'importe où, à tout moment avec leurs appareils connectés sans fil. [5]
- ❖ **Rapides:** Le client présente son appareil mobile, authentifie la transaction et termine la transaction. Cela crée une expérience rapide et plus conviviale pour les consommateurs et aide les commerçants à accélérer les transactions. [6]
- ❖ **Peut s'avérer moins coûteux pour les entreprises:** simplement en mettant en œuvre des méthodes de paiement caractéristiques de m-commerce telles que les scanners de QR Codes, les entreprises peuvent économiser les coûts mensuels qui accompagnent les terminaux de cartes de crédit.[5]
- ❖ **Sécurisés:** Les portefeuilles numériques fournissent également des couches supplémentaires d'authentification biométrique. Les informations sensibles du titulaire de carte sont remplacées par des jetons que les fraudeurs ne peuvent pas utiliser s'ils sont interceptés pendant les sessions de paiement, ce qui réduit les risques de fraude pour les commerçants et les consommateurs. [6]

### 2.5.2. Inconvénients du paiement mobile

- ❖ Le risque de ne pas pouvoir effectuer le paiements via mobile quand la batterie du smartphone est faible. En effet les applications de paiement ne seront plus fonctionnelles jusqu'à ce que la batterie se recharge. [6]
- ❖ Les vendeurs en ligne risquent d'être confrontés à d'énormes problèmes s'ils n'ont pas des dispositifs pour détecter les tentatives de fraude. Certains clients frauduleux peuvent acheter sur le site d'un vendeur et si celui-ci n'a pas des moyens de sécurisation, des complications peuvent en découler comme un litige de fraude dans lequel le vendeur sera fortement impliqué. Ces risques peuvent être contrés par des détecteurs de fraude. [16]

## Conclusion

Les paiements mobiles ont révolutionné la façon dont les entreprises envisagent le traitement des paiements en assurant d'un côté des moyens plus sécurisés et authentifiés de traiter les paiements, et de l'autre, des moyens plus rapides et plus pratiques.

Dans ce chapitre nous avons présenté le paiement mobiles, et ses différents types afin de déterminer le type de m-paiement qui conviendra le mieux à notre cas d'étude.

Dans le prochain chapitre, nous aborderons les méthodes et langages utilisées pour la conception de notre application.



# Chapitre III: Analyse et Conception.

## Introduction

Tout processus de développement d'applications ou de systèmes informatiques doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement, afin de le rendre plus fidèle aux besoins du client.

Dans ce chapitre, nous abordons une tâche importante dans l'élaboration de ce travail, à savoir l'analyse et la conception. En effet nous présentons l'approche que nous avons suivie pour concevoir notre application et ses différentes étapes en les illustrant avec quelques diagrammes conformes au langage UML.

## 1. Définition du langage UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.[\[28\]](#)

UML nous fournit donc des schémas appelés diagrammes pour représenter le logiciel à développer: son fonctionnement, sa mise en route, sa limite, sa structure les actions susceptibles d'être effectuées par le logiciel, etc [\[28\]](#)

UML n'est pas une méthode, mais peut être employé dans tout le cycle de développement, indépendamment de la méthode.[\[30\]](#)

Initialement, les buts des concepteurs d'UML étaient les suivants : [\[30\]](#)

- ❖ Représenter des systèmes entiers (pas uniquement logiciels) par des concepts objets.
- ❖ Lier explicitement des concepts et le code qui les implantent.
- ❖ Pouvoir modéliser des systèmes à différents niveaux de granularité, (pour permettre d'appréhender des systèmes complexes).

## 2. Processus du développement de notre solution

Le processus de développement de logiciels met en œuvre un ensemble d'activités qui à partir d'une demande d'informatisation d'un processus permettent la conception, l'écriture et la mise au point d'un logiciel (et donc de programmes informatiques) jusqu'à sa livraison au demandeur. [\[25\]](#)

Le développement de notre application a suivie dans l'ordre les phases suivantes:

- ❖ Spécification des besoins.
- ❖ Analyse des besoin.
- ❖ conception.
- ❖ Réalisation : Codage et tests unitaires.
- ❖ Qualification.

Nous détaillons dans ce chapitre les phases de spécification et d'analyse des besoins ainsi que les phase de conception globale et détaillée.

### 2.1. Spécification des besoins

La définition des besoins et des exigences correspond à l'étape dans laquelle nous discussions avec le client et les futurs utilisateurs afin de comprendre de quoi ils ont besoin, en d'autres termes: QUI doit pouvoir faire QUOI ?

Elle permet d'établir une première description du futur système et contient plusieurs étapes:

#### 2.1.1. Etablissement d'un cahier des charges

##### 2.1.1.1. Définition

Le cahier des charges est un document contractuel rédigé par le client, qui décrit, en langage naturel, les fonctionnalités attendues du produit. [\[35\]](#)

#### 2.1.2. Délimiter le contour du système au cours d'étude

Définir clairement ses frontières et les acteurs avec lesquels il communique. Ces derniers peuvent être soit des acteurs humains, soit d'autres systèmes avec lesquels le futur système communique.

Notre système se compose de deux applications mobiles:

- Une application pour le vendeur de l'entreprise commerciale.
- Une application de portefeuille mobile pour le client de l'entreprise.

❖ L'application côté vendeur communique avec:

- Un utilisateur: le vendeur de l'entreprise commerciale.
- L'application côté client de l'entreprise: L'application côté vendeur génère un code QR qui sera scanné avec l'application du client pour valider le paiement.
- Le système informatique de l'entreprise: L'application du vendeur communique avec la base de données de l'entreprise pour récupérer les commandes associées au vendeur par exemple.

❖ L'application de portefeuille mobile pour le client de l'entreprise communique avec:

- Un utilisateur: Le client de l'entreprise désirant payer sa commande avec le paiement mobile.
- L'application côté vendeur de l'entreprise: Le client de l'entreprise, en utilisant son application, scanne le code QR généré par l'application du vendeur pour effectuer un paiement.
- Un système de banque ou de cartes de crédit: L'application du client communique alors avec un système de banque ou de carte de crédit pour valider la transaction et confirmer le paiement.

### 2.1.3. Résumer le tout dans un diagramme de contexte

#### 2.1.3.1 Définition

Le diagramme de contexte sert à délimiter le contour du système en cours d'étude, ie définir clairement ses frontières et les acteurs avec lesquels il communique. Ces derniers peuvent être soit des acteurs humains, soit d'autres systèmes avec lesquels le futur système doit communiquer. [28]

#### 2.1.3.2. Le diagramme de contexte pour notre cas d'étude

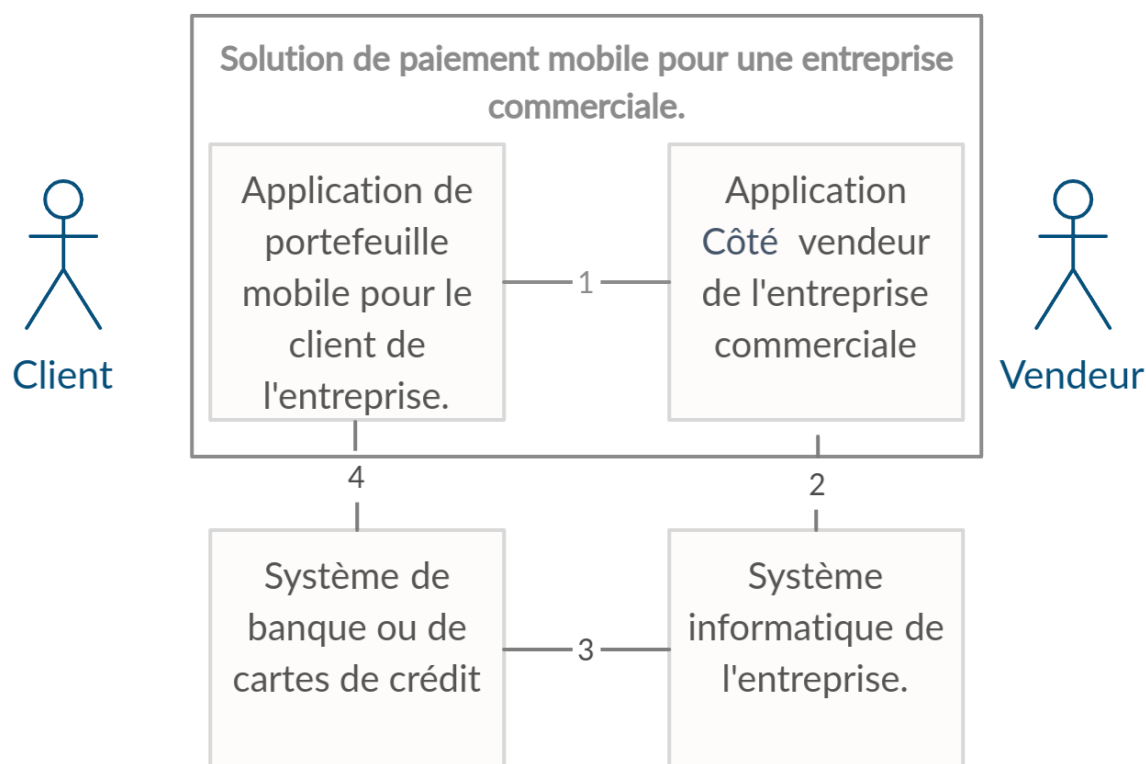


Figure 1. Diagramme de contexte.

### 2.1.4. Décrire le produit à partir des différents scénarios d'utilisation

Nous présentons dans le tableau ci-dessous les différentes tâches qu'effectuent les acteurs de notre système ainsi que leurs scénarios.

Acteur	Tâche	Scénario
Vendeur	<p>-T1: Accéder à l'application du vendeur</p> <p>-T2: S'authentifier</p> <p>-T3: Accéder à la commande du client</p> <p>-T4: Afficher les détails de la commande.</p> <p>T5: Afficher les options de paiement.</p> <p>T6: Sélectionner le moyen de paiement par Qr code.</p> <p>T7: Sélectionner le moyen de paiement en liquide.</p> <p>T8: Sélectionner le moyen de paiement par chèque.</p> <p>T9: Consulter l'historique des transactions qu'il a effectué.</p> <p>T10: Consulter les informations de son compte.</p>	<p>-Télécharger et installer l'application.</p> <p>-Saisir le nom d'utilisateur et mot de passe.</p> <p>-Cliquer sur le bouton "Commandes" et sélectionner la commande voulue.</p> <p>-Cliquer sur la commande et laisser apparaître les détails de celles-ci.</p> <p>-Cliquer sur le bouton "passer au paiement" et laisser apparaître l'interface affichant les options du paiement.</p> <p>-Cliquer sur l'icône "Qr code " et laisser apparaître le qr code spécifique à la commande sélectionnée.</p> <p>-Cliquer sur l'icône "liquide" et remplir le formulaire correspondant puis le confirmer.</p> <p>-Cliquer sur l'icône "Chèque" et remplir le formulaire correspondant puis le confirmer.</p> <p>-Cliquer sur l'icône "historique" et voir les transactions qu'il a effectué et leurs détails.</p> <p>-Cliquer sur l'icône "Informations" et consulter</p>

		les informations de son compte.
	T11: se déconnecter.	-Cliquer sur l'icône "Déconnexion" dans la page d'accueil.
Client	<p>-T12: Accéder à l'application de portefeuilles mobile.</p> <p>-T12: Saisir les informations de sa carte de crédit/débit.</p> <p>-T13: Choisir une de ses cartes bancaires avec laquelle il veut payer.</p> <p>-T14: Effectuer un paiement</p> <p>-T15: Retour à la page principale.</p>	<p>-Télécharger et installer l'application.</p> <p>-Remplir un formulaire avec ses coordonnées.</p> <p>-La liste de ses cartes bancaires s'affiche à l'écran et le client peut cliquer sur celle qui lui convient pour payer avec.</p> <p>-Cliquer sur le bouton "Payer", et l'appareil photo va se lancer pour scanner le code QR du vendeur.</p> <p>-Un message de confirmation lui sera affiché pour l'informer de la validation du paiement.</p> <p>-Après avoir payé, le client a le choix entre effectuer un nouveau paiement ou retourner à la page principale en cliquant sur le bouton "Retour".</p>

*Table 2. Tableau des tâches et scénarios.*

## 2.1.5. Etablissement du diagramme des cas d'utilisations

Après avoir défini les tâches à effectuer par les utilisateurs et leurs scénarios, nous résumons celles-ci en UML en établissant un diagramme des cas d'utilisation.

### 2.1.5.1. Définition

Le diagramme des cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (Humain ou Machine) et un système. Il est une entité significative de travail. Dans un diagramme de cas d'utilisation, il existe des acteurs qui interagissent avec des cas d'utilisation. Les cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs du système. Une fois identifiés et structurés, ces besoins permettent d'identifier les fonctionnalités principales ou critiques du système. [\[29\]](#)

### 2.1.5.2. Diagramme des cas d'utilisation pour notre cas d'étude

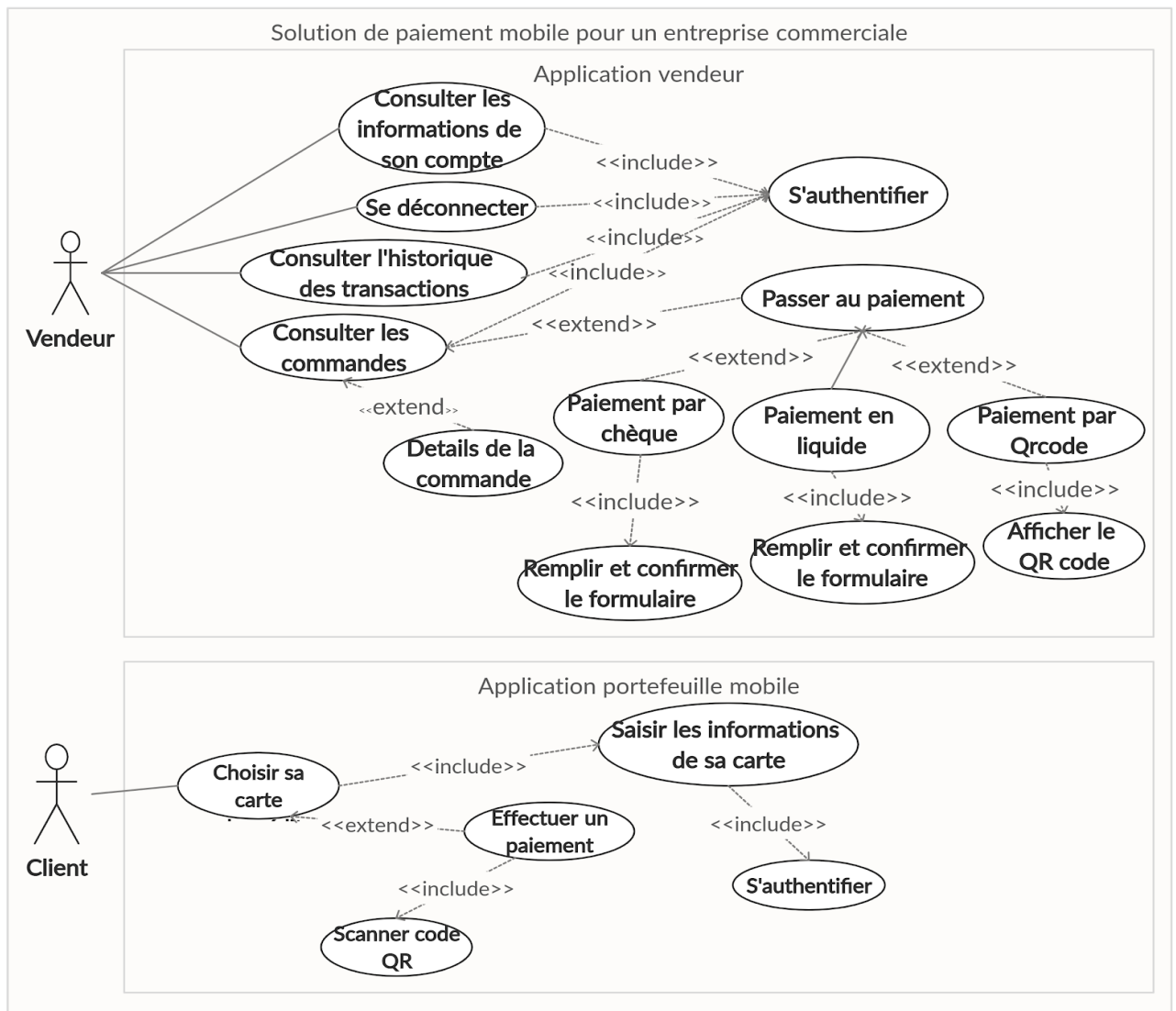


Figure 2. Diagramme des cas d'utilisation.

## 2.2 Définition du Produit (analyse des besoins)

L'analyse des besoins permet d'affiner ce qui a été défini dans l'étape précédente, en détaillant davantage le fonctionnement interne du futur système (en d'autres termes, comment cela doit-il fonctionner?) Elle se divise en plusieurs autres étapes:



### 2.2.1. Décrire les spécifications précises du produit

Décrire le type de nos applications mobiles et leurs environnement, et ressources disponibles ainsi que les contraintes de réalisations.

Après avoir effectué une étude sur les types d'applications mobiles et leurs plateformes, ainsi que sur les types de paiements mobiles, nous avons effectué les choix qui correspondent à nos critères de sélection qui sont:

- ❖ développement et maintenance rapides, open source et nécessitant moins de ressources.
- ❖ Type de paiement pratique, convenable et moins coûteux en ressources.

Le tableau ci-dessous représente le type de nos applications ainsi que leurs plateforme:

Application	Type	Plateforme	Type de paiement mobile
Application côté vendeur de l'entreprise	Application mobile native	Android	Paiement mobile par scan de QR code.
Application de portefeuille mobile pour le client de l'entreprise	Application mobile native	Android	

*Table3 . Tableau des plateforme et type d'applications.*

### 2.2.2. Spécifications fonctionnelles et non fonctionnelles

Préciser les spécifications fonctionnelles (extraire les fonctionnalités ou fonctions ainsi que les interfaces utilisateur) et non fonctionnelles (temps de réponses, contraintes mémoire..) .

### 2.2.2.1. Extraire les fonctionnalités du système et les interfaces utilisateur

Le tableau ci-dessous représente les fonctionnalités de notre système ainsi que ses interfaces utilisateur:

Application	Interfaces utilisateur	Fonctionnalités
Application côté vendeur de l'entreprise.	-Interface d'authentification	-Authentification du vendeur avec un nom d'utilisateur et mot de passe.
	-Interface d'accueil	<ul style="list-style-type: none"> <li>-Cliquer sur le l'icône "Commandes" et afficher une autre interface.</li> <li>-Cliquer sur le l'icône "Historique" et afficher une autre interface.</li> <li>-Cliquer sur le l'icône "informations" et afficher une autre interface.</li> <li>-Cliquer sur le l'icône "paramètres" et afficher une autre interface.</li> </ul>
	-Interface des commandes	-En cliquant sur l'icône "commandes" une interface apparaît affichant toutes les commandes liées à ce vendeur.
	-Interface détails de la commande.	-En cliquant sur une commande, une autre interface apparaît et affiche les détails de la

	commande et un bouton "options de paiement".
-Interface options de paiement.	-En cliquant sur le bouton "option de paiement" une interface s'affiche proposant le paiement par Qr code, en liquide, ou par chèque.
-Interface du code QR	-En cliquant sur l'icône "Qr code", une autre interface apparaît, affichant un code QR. -Un message de confirmation s'affiche une fois que le paiement est effectué.  -Retour à la page principale.
-Interface du paiement en liquide	-En cliquant sur l'icône "liquide", une autre interface s'affiche contenant un formulaire que le vendeur remplira puis confirmera. -Un message de confirmation s'affiche une fois que le paiement est effectué.  -Retour à la page principale.
-Interface du paiement par chèque	-En cliquant sur l'icône "chèque", une autre interface s'affiche contenant un formulaire que le vendeur remplira puis confirmera. -Un message de confirmation s'affiche une fois que le paiement est effectué.

		-Retour à la page principale.
	-Interface de l'historique des transactions	-En cliquant sur l'icône "historique" dans la page d'accueil, une interface s'affiche contenant toutes les transaction que ce vendeur a effectué.
	-Interface détails de la transactions.	-En cliquant sur une transaction, une interface s'affiche contenant les détails de celle-ci.
	-Interface des informations supplémentaires	En cliquant sur l'icône "Informations" dans la page d'accueil, une interface s'affiche lui affichant des informations sur son compte.
Application de portefeuille mobile pour le client de l'entreprise.	-Interface de saisie des informations bancaire	-Remplir un formulaire avec ses coordonnées bancaires.
	-Interface d'accueil	-Choisir avec quel carte de crédit/débit que l'utilisateur veut utiliser pour payer.
	-Interface détails de la carte et proposition de paiement	-En cliquant sur une carte, une autre interface apparaît et affiche les détails de la carte et un bouton "payer".

	-Interface de paiement	-En cliquant sur le bouton "payer", l'application accède à l'appareil photo du téléphone mobile pour pouvoir scanner le code Qr et effectuer le paiement en enregistrant la transaction. -Un message de confirmation lui sera affiché une fois la transaction effectuée.
	-Interface de confirmation du paiement	-En scannant le QR code, le client patiente quelque seconde et une autre interface apparaît affichant un message de confirmation du paiement.

*Table4 . Tableau des interfaces utilisateur et fonctionnalités*

#### 2.2.2.2. Extraire les spécifications non fonctionnelles

La solution à réaliser respecte les critères suivants :

- ❖ **Ergonomie de l'interface:** Les application doivent être faciles à utiliser, et les interfaces utilisateurs doivent être conviviales, simples, ergonomiques et adaptées à l'utilisateur.
- ❖ **Disponibilité:** L'application doit être disponible à tout instant.
- ❖ **Performance:** Les applications doivent être performantes en terme temps de réponse, nombre de transactions par seconde, nombre d'utilisateurs simultanés, nombre d'accès aux bases de données par seconde.

## 2.3. La phase de la conception

La phase de la conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, et son architecture afin d'en faciliter la réalisation.

Nous avons divisé cette phase en plusieurs étapes:

### 2.3.1. Éclater le logiciel en sous-parties:

Dans cette étape, nous divisons notre système en sous-systèmes appelées modules ou composants logiciels.

Cette étape aide à mieux comprendre le système et permet aux développeurs de travailler sur des parties individuelles du système en isolation et les faire communiquer une fois prêts. [\[31\]](#)

Nous avons divisé notre solution informatique en deux modules:

- ❖ **Module 1:** Application mobile côté vendeur de l'entreprise commerciale.
- ❖ **Module 2:** Application de portefeuille mobile pour le client de l'entreprise.

### 2.3.2. Architecture du système (composants et liens entre ces sous-parties)

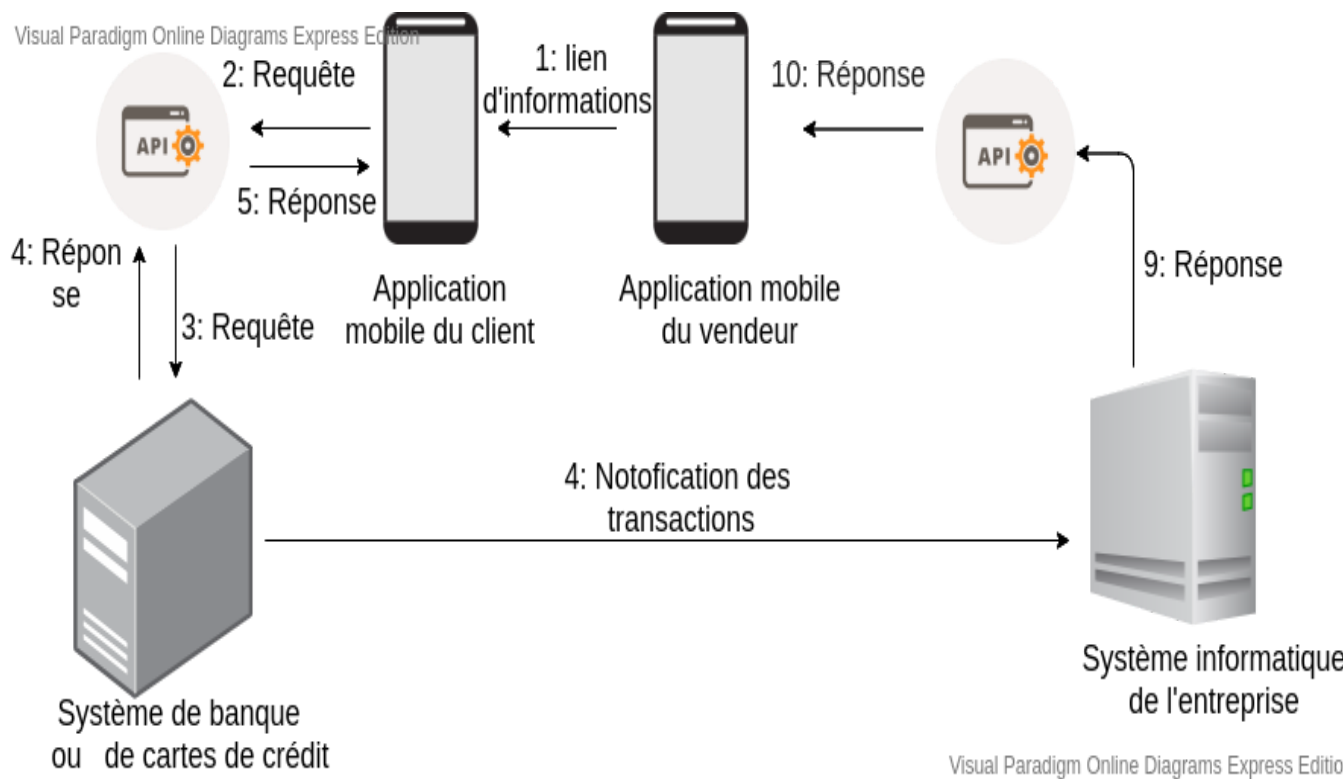
#### 2.3.2.1. Définition

Une architecture logicielle est une représentation abstraite d'un système exprimée essentiellement à l'aide de composants logiciels en interaction via des connecteurs. L'architecture logicielle réduit la complexité du système abordé en le structurant en composants logiciels. Elle ne fournit que les propriétés externes des éléments structurants et ne se préoccupe pas des détails d'implantation.

Les composants sont des spécifications d'unités fonctionnelles clairement définies, sémantiquement cohérentes et compréhensibles.

Les connecteurs sont des objets du premier ordre qui assurent les interactions entre composants, et qui peuvent être de complexité variable . Ils permettent la flexibilité et l'évolution. [32]

#### 2.3.2.2. Architecture globale de notre système



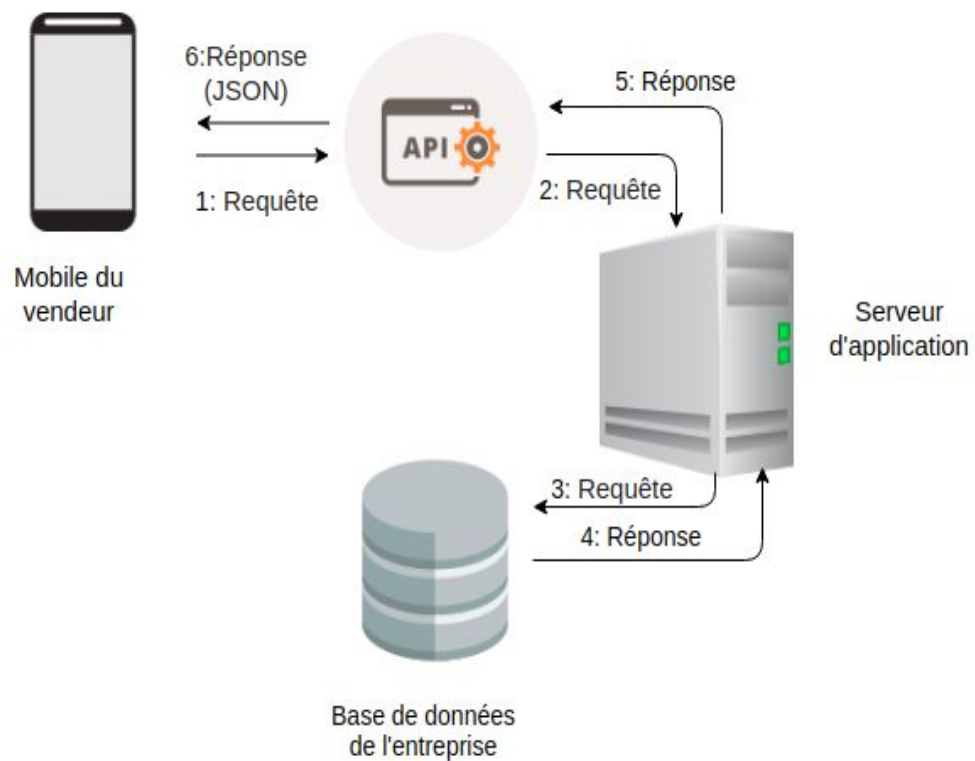
*Figure 3: Architecture globale du système.*

#### 2.3.2.3. Architecture des sous-parties

Après avoir défini l'architecture globale de notre système ainsi que ses différents modules, nous détaillons dans cette phase l'architecture de ces derniers.

#### 2.3.2.3.1. L'architecture de l'application côté vendeur

Le schéma ci-dessous représente l'architecture de l'application côté vendeur de l'entreprise:

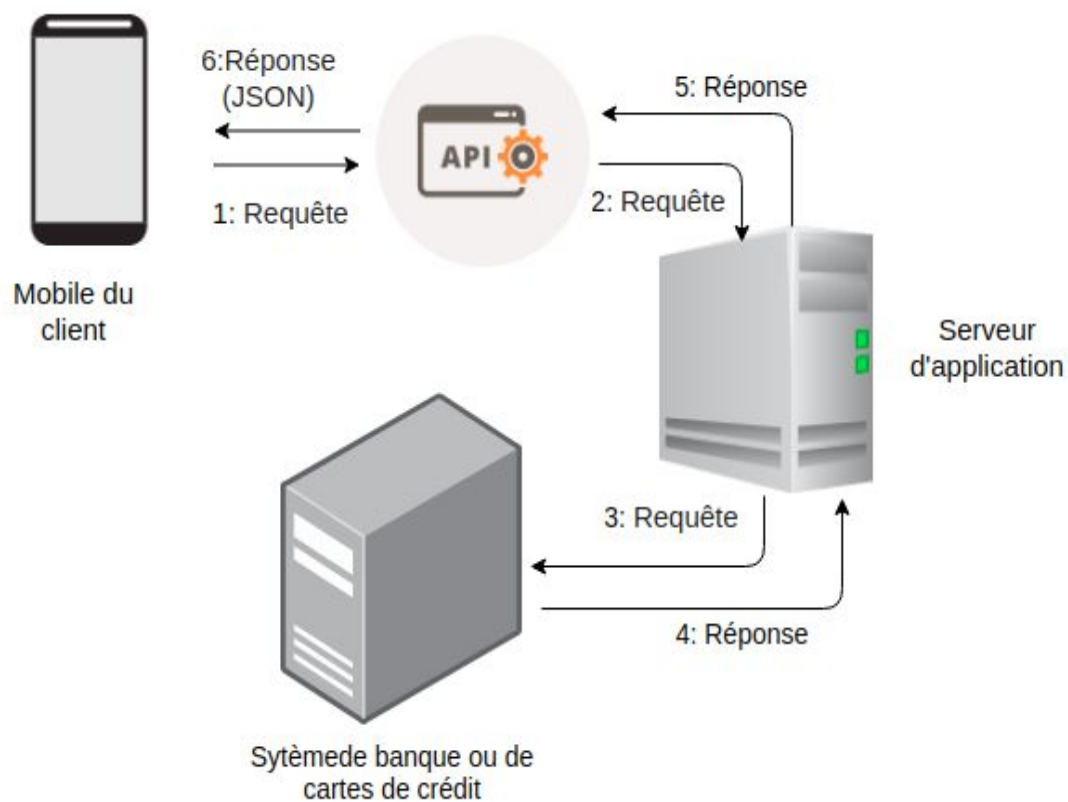


*Figure 4: Architecture de l'application vendeur.*

#### 2.3.2.3.2. L'architecture de l'application côté client

Le schéma ci-dessous représente l'architecture de l'application côté client de l'entreprise:





*Figure 5: Architecture détaillée de l'application client.*

#### 2.3.2.4. Les vues (structurelles) d'une architecture logicielle

Celles-ci permettent d'organiser la description du système en plusieurs vues complémentaires, chacune présentant l'architecture du système selon un point de vue différent. L'utilisation de vues permet de traiter séparément les intérêts des divers groupes d'intervenants (architectes, utilisateurs, développeurs, chefs de projets, etc.) [31] Nous en citons:

##### 2.3.2.4.1. La vue logique [31]

Description logique du système décomposé en sous-systèmes (modules + interface). Elle décrit les aspects statiques( qui identifient les propriétés des objets et leurs

liaisons avec les autres objets [\[51\]](#)) et les aspects dynamiques (qui définissent le cycle de vie des objets en précisant: leur comportement, les différents états par lesquels ils passent et les événements qui déclenchent ces changements d'états [\[51\]](#)) d'un système en termes de classes, d'objets, de connexions et de communications.

En UML: Etablissement du diagramme de paquetages, diagramme de classe, diagrammes de séquences etc.

#### ❖ **Diagrammes de classes pour notre cas d'étude**

Le diagramme des classes est un diagramme structurel (statique) qui montre la structure interne du système. Il permet de représenter les classes (attributs + méthodes) et les associations (relations) entre ces classes. Il est nécessaire lors de la modélisation objet d'un système.

Une classe est une représentation abstraite d'un ensemble d'objets, elle contient les informations nécessaires à la construction de l'objet (c'est-à-dire la définition des attributs et des méthodes). La classe peut donc être considérée comme le modèle, le moule ou la notice qui va permettre la construction d'un objet. On dit également qu'un objet est l'instance d'une classe (la concrétisation d'une classe). [\[33\]](#)

Nous présentons ci-dessous les diagrammes de classes de notre cas d'étude:

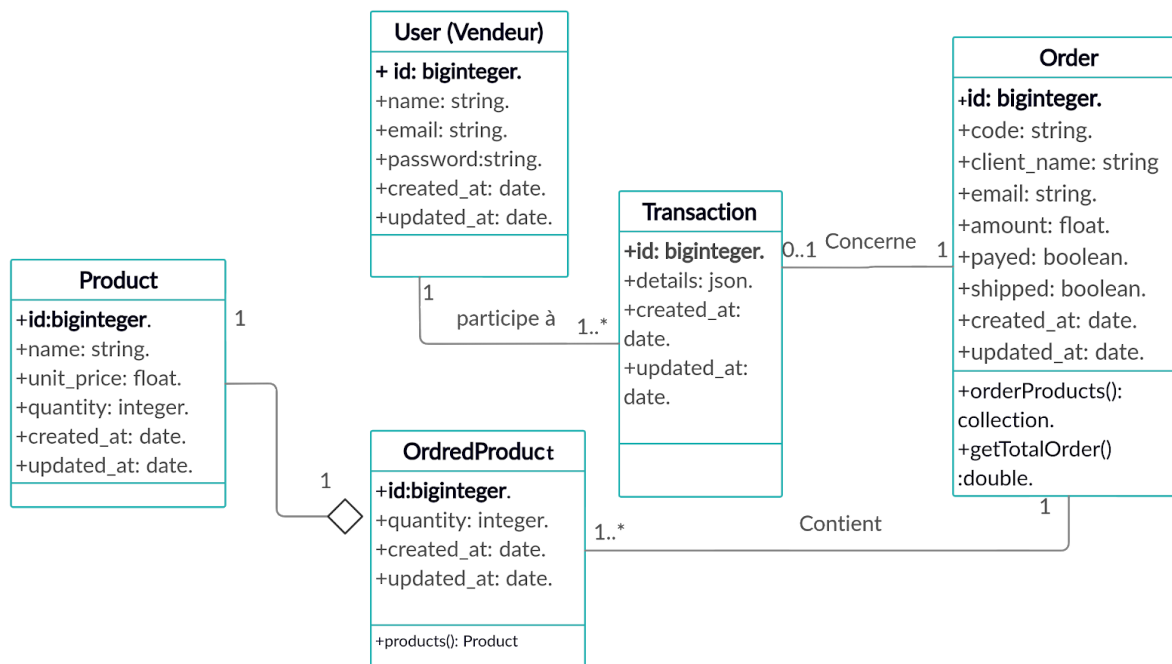


Figure 6: Diagramme de classes pour l'application vendeur.

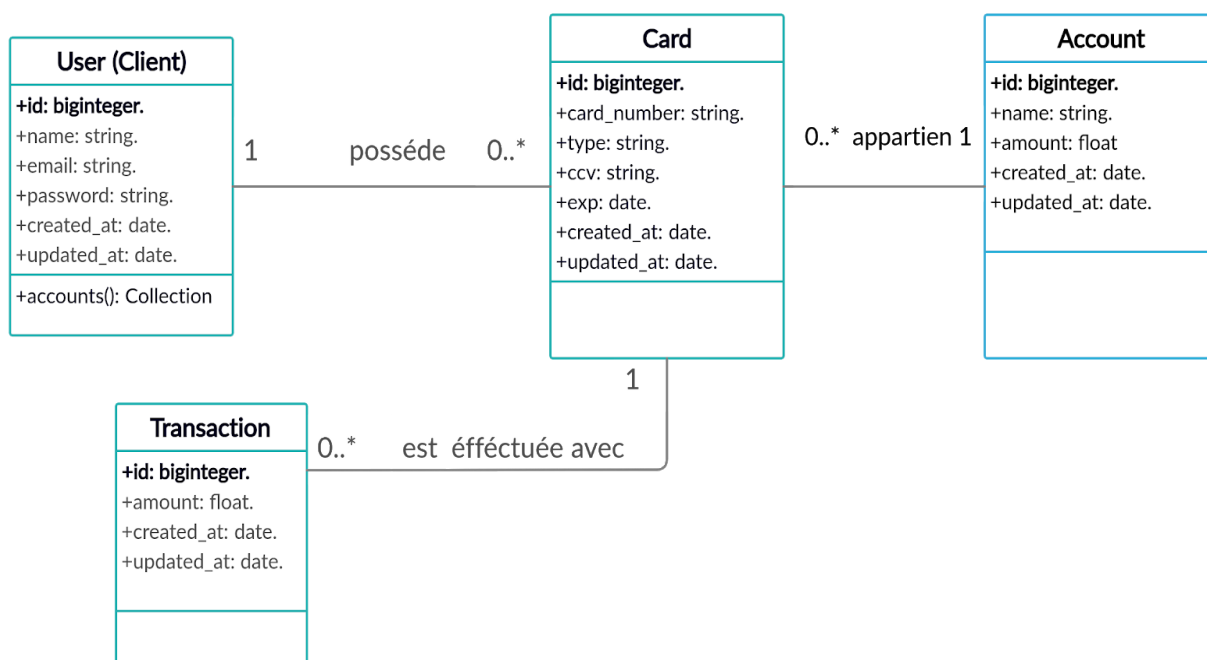


Figure 7: Diagramme des classes pour l'application client.

### ❖ Diagrammes de séquences

Les diagrammes de séquences permettent de décrire l'aspect dynamique du système qui représente comment les éléments du système interagissent entre eux et avec les acteurs. Les objets au coeur d'un système interagissent en s'échangeant des messages. Les acteurs interagissent avec le système au moyen des IHM (Interfaces Homme-Machine).<sup>[34]</sup> Nous présentons ci-dessous les diagrammes de séquences de notre cas d'étude liés au cas d'utilisation "Authentification du vendeur" et "effectuer un paiement par Qr code".

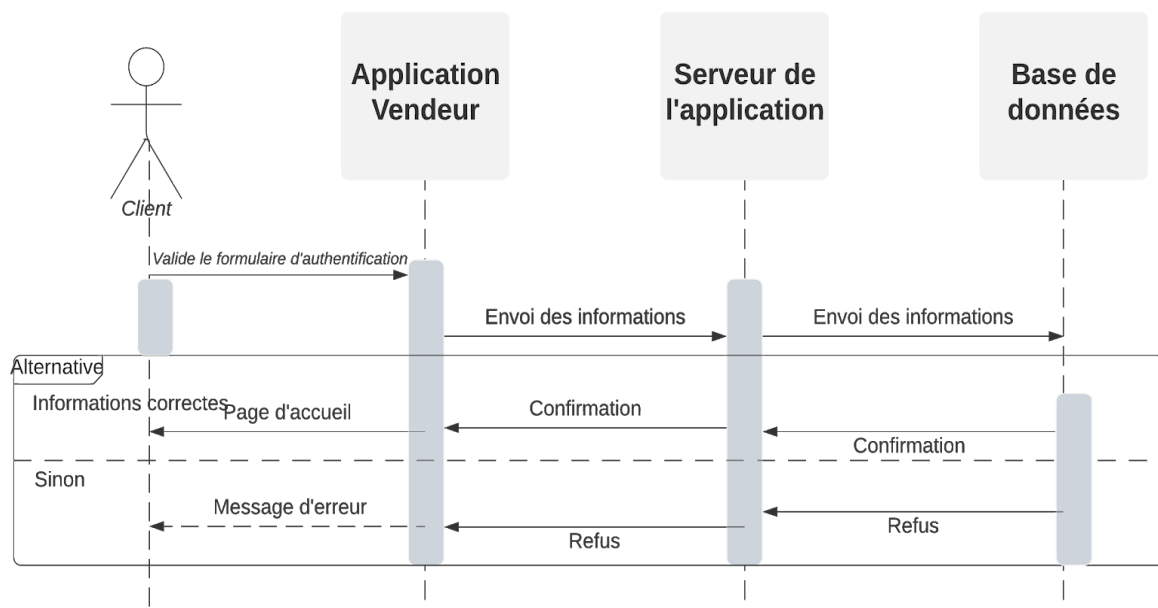
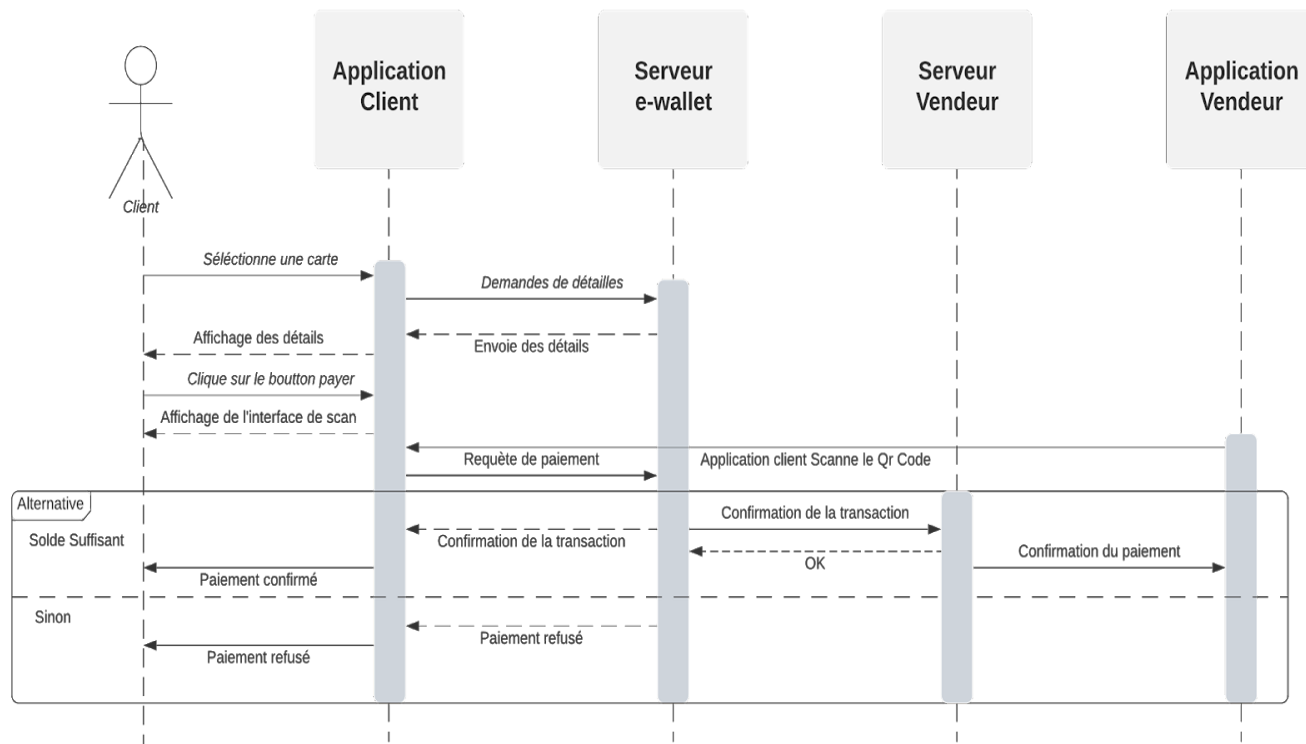


Figure 8: Diagramme de séquences du cas d'utilisation "Authentification".



*Figure 9: Diagramme de séquences du cas d'utilisation “Effectuer un paiement par Qr code”.*

#### 2.3.2.4.2 La vue d’implémentation

Description de l’implémentation (physique) du système logiciel en termes de composants et de connecteurs.

En UML: Etablissement du diagramme de composants. [\[31\]](#)

#### 2.3.2.4.3. La vue de déploiement

Description de l’intégration et de la distribution de la partie logicielle sur la partie matérielle . [\[31\]](#)

En UML: Etablissement du diagramme de composants et de déploiement.

# Conclusion

Dans ce chapitre, nous avons présenté l'analyse et la conception de notre solution, en les illustrant avec quelques diagrammes en utilisant le formalisme UML.

Au niveau de l'analyse, nous avons défini les besoins de notre client, et détaillé les spécifications fonctionnelles et non fonctionnelles que doit contenir notre solution. Et nous avons tracé deux diagrammes UML à savoir le diagramme de contexte et le diagramme de cas d'utilisations.

Au niveau de la conception, nous avons défini l'architecture globale et détaillée de notre solution, et nous avons réalisé deux diagrammes de séquences et deux diagrammes de classes.

Le chapitre suivant est consacré à phase de l'implémentation et la réalisation de nos deux applications, à la présentation des différentes interfaces, ainsi que les outils utilisés durant la réalisation.

# Chapitre 4: Implémentation et réalisation.

## Introduction

La réalisation de notre solution est le résultat de la mise en œuvre de l'analyse et de la conception présentées dans le chapitre précédent. C'est l'aboutissement final de notre projet et la raison d'être du projet lui-même.

Pour pouvoir mener à bien cette étape, il est nécessaire de choisir des technologies adaptées à sa réalisation.

Nous présentons dans ce chapitre la réalisation de notre application. Pour ce faire, nous avons structuré notre chapitre comme suit :

- ❖ Une première section consacrée à la présentation de l'environnement de développement (outils, langages..) utilisés durant cette phase.
- ❖ Une seconde section consacrée à présentation des fonctionnalités de notre système à travers ses interfaces utilisateur.

## 1. Outils et langages de développement

Dans cette section, nous présentons les différents outils logiciels, langages de programmation et frameworks utilisés pour réaliser notre solution de paiement mobile.

### 1.1. Le framework React Native

React Native est un framework JavaScript (qui est un langage de script léger, principalement connu comme le langage de script des pages web, mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web. Le code JavaScript est interprété ou compilé à la volée. C'est un langage à objets utilisant le concept de prototype, disposant d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation: fonctionnelle, impérative et orientée objet.[\[37\]](#) ) permettant d'écrire des applications mobiles natives pour iOS et Android. Il est basé sur React, la bibliothèque JavaScript de Facebook, pour créer des applications front-end, mais au lieu de cibler le navigateur, il cible les plates-formes mobiles. En d'autres termes, les

développeurs Web peuvent désormais écrire des applications mobiles qui semblent et se sentent vraiment «natives», le tout dans le confort d'une bibliothèque JavaScript. De plus, étant donné que la plupart du code écrit peut être partagé entre les plateformes, React Native facilite le développement simultané pour Android et iOS. Les applications React Native sont écrites à l'aide d'un mélange de balises JavaScript et HTML (qui signifie Hypertext Markup Language. Il permet à l'utilisateur de créer et de structurer des sections, des paragraphes, des en-têtes, des liens et des citations pour les pages et les applications Web. [38]), connu sous le nom de JSX. Ensuite, le «pont» React Native invoque les API de rendu natives dans Objective-C (pour iOS) ou Java (pour Android). Ainsi, l'application restituera en utilisant de vrais composants d'interface utilisateur mobile, et non des vues Web, et ressemblera à toute autre application mobile. React Native expose également des interfaces JavaScript pour les API de plate-forme, afin que les applications React Native puissent accéder aux fonctionnalités de la plate-forme telles que la caméra du téléphone ou l'emplacement de l'utilisateur. [36]

## 1.2. Le framework Laravel

Laravel est un framework PHP (PHP signifie PHP Hypertext Preprocessor. Il s'agit d'un langage de script serveur open source utilisé pour le développement d'applications Web. Par langage de script, nous entendons un programme basé sur un script ou lignes de code écrit pour l'automatisation des tâches. [40]) open source conçu pour rendre le développement d'applications Web plus facile et plus rapide grâce aux fonctionnalités intégrées. Il suit le modèle de conception "modèle-vue-contrôleur". Ses fonctionnalités le rendent largement utilisé par les développeurs Web, nous en citons certaines:

- ❖ Un système d'emballage modulaire avec gestion des dépendances: Cela signifie que nous pouvons facilement ajouter des fonctionnalités à notre application Laravel sans les écrire à partir de zéro. Nous pouvons soit créer nos propres packages pour le code que nous utilisons régulièrement, soit installer des packages prêts à l'emploi via Composer (gestionnaire de dépendances).
- ❖ Un système d'authentification complet.
- ❖ Mappage relationnel-objet: Eloquent ORM inclus avec Laravel, présente les tables de base de données sous forme de classes pour un accès et une manipulation des données plus faciles.



- ❖ Une interface de ligne de commande (CLI) qui comprend des dizaines de commandes prédéfinies (Artisan).
- ❖ Test automatique. Des tests automatisés font partie intégrante de Laravel.
- ❖ Un environnement de développement virtuel portable. Homestead fournit aux développeurs tous les outils nécessaires pour développer Laravel dès la sortie de la boîte. [\[39\]](#)

### 1.3. Postman

Postman est un outil de développement logiciel. Il permet aux utilisateurs de tester les appels aux APIs. Les utilisateurs de Postman saisissent des données, les données sont envoyées à une adresse de serveur Web spéciale. Les informations sont renvoyées, et Postman les présente à l'utilisateur.

Postman aide à créer, tester et modifier des API. Presque toutes les fonctionnalités dont tout développeur pourrait avoir besoin sont encapsulées dans cet outil. Il est utilisé par plus de 5 millions de développeurs chaque mois pour rendre leur développement d'API facile et simple. Il a la capacité de faire différents types de requêtes HTTP (GET, POST, PUT, PATCH), d'enregistrer des environnements pour une utilisation ultérieure, de convertir l'API en code pour différents langages (comme JavaScript et Python). [\[45\]](#)

### 1.4. MYSQL

MYSQL est un Système de Gestion de Base de données (SGBD) qui permet d'entreposer des données de manière structurée (Base, tables, champs, enregistrements, etc.). Le noyau de ce système permet d'accéder à l'information entreposée via un langage spécifique qu'on appelle SQL (Structured Query Language). MySQL doit sa popularité à sa rapidité, sa robustesse, la facilité de son utilisation, ses performances et son caractère open source. [\[41\]](#)

## 1.5. HeidiSQL

HeidiSQL est un outil d'administration de base de données possédant un éditeur SQL et un constructeur de requête. Il a été développé et optimisé pour être utilisé avec le SGBD relationnel MySQL disponible commercialement ou gratuitement. [\[42\]](#)

## 1.6. Git [\[44\]](#)

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux. Git est un système de contrôle de version distribué et open source:

- ❖ Système de contrôle: cela signifie essentiellement que Git est un outil de suivi de contenu. Git peut donc être utilisé pour stocker du contenu. Il est principalement utilisé pour stocker du code en raison des autres fonctionnalités qu'il fournit.
- ❖ Système de contrôle de version: Le code qui est stocké dans Git continue de changer à fur et à mesure que l'on en ajoute. De nombreux développeurs peuvent également ajouter du code en parallèle. Ainsi, le système de contrôle de version aide à gérer cela en conservant un historique des changements qui se sont produits.
- ❖ Système de contrôle de version distribué: Git possède un référentiel distant qui est stocké sur un serveur et un référentiel local qui est stocké dans l'ordinateur de chaque développeur. Cela signifie que le code n'est pas seulement stocké sur un serveur central, mais la copie complète du code est présente sur tous les ordinateurs des développeurs.

## 1.7. Visual Studio Code

Visual Studio Code est un éditeur de code open-source et extensible développé par Microsoft pour Windows, Linux et macOS. Il supporte un très grand nombre de langages grâce à des extensions. Il supporte aussi l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git. Il est aussi bien performant et agréable à utiliser. [\[43\]](#)

## 1.8. Pusher

Pusher est une infrastructure en tant que service, qui propose des API haut niveau pour créer des applications temps réel (application ou ensemble d'applications informatique dont le fonctionnement est assujetti à l'évolution dynamique d'un procédé extérieur qui lui est connecté et dont il doit contrôler le comportement. [\[48\]](#) ) sans se soucier de l'infrastructure ni des détails bas niveau des implémentations. [\[47\]](#)

## 1.9. Heroku

Heroku est une plateforme en tant que service, qui permet d'héberger des applications web sans avoir à configurer les serveurs et les ressources de ce dernier, et permet aussi le déploiement continu (qui est une stratégie de développement logiciel où toute validation de code qui réussit le cycle de test automatisé est automatiquement transférée dans l'environnement de production, propulsant ainsi les modifications vers les utilisateurs du logiciel. [\[53\]](#)). La plate-forme Heroku prend en charge le développement dans Ruby on Rails, Php, Java, Python, Node.js, Python, Go, Scala et Clojure. [\[46\]](#)

## 1.10. L'intervention de ces outils dans notre travail

Nous présentons dans le tableaux ci-dessous l'utilisation des outils définis précédemment dans le développement de notre solution.

Outil	Utilisation dans le développement
Visual Studio Code	-Editeur de text ou l'on écrit notre code.
Le framework React Native	-Construction des applications mobiles
Le framework Laravel	-Création et remplissage de la base de données. -Ecriture des APIs qui permettent à nos applications mobiles de communiquer avec la base de données.
Postman	-Outils logiciel utilisé pour le test des APIs.

MYSQL	Gestion de la base de données.
HeidiSQL	Visualisation de la base de données.
Pusher	Confirmation en temps réel du paiement dans les deux applications mobiles.
Git	Traçage des différentes versions du code, et importation du projet sur la plateforme GitHub pour en garder copie.
Heroku	Hébergement des applications web à fur et à mesure du développement.

*Table 5. Tableau de l'utilisation des outils dans le développement de la solution.*


## 2. Présentation de nos applications


Nous présentons dans cette partie les deux applications mobiles que nous avons conçues à travers leurs interfaces utilisateur:

### 2.1. L'application côté vendeur de l'entreprise

#### 2.1.1. Interface d'authentification

# Connexion

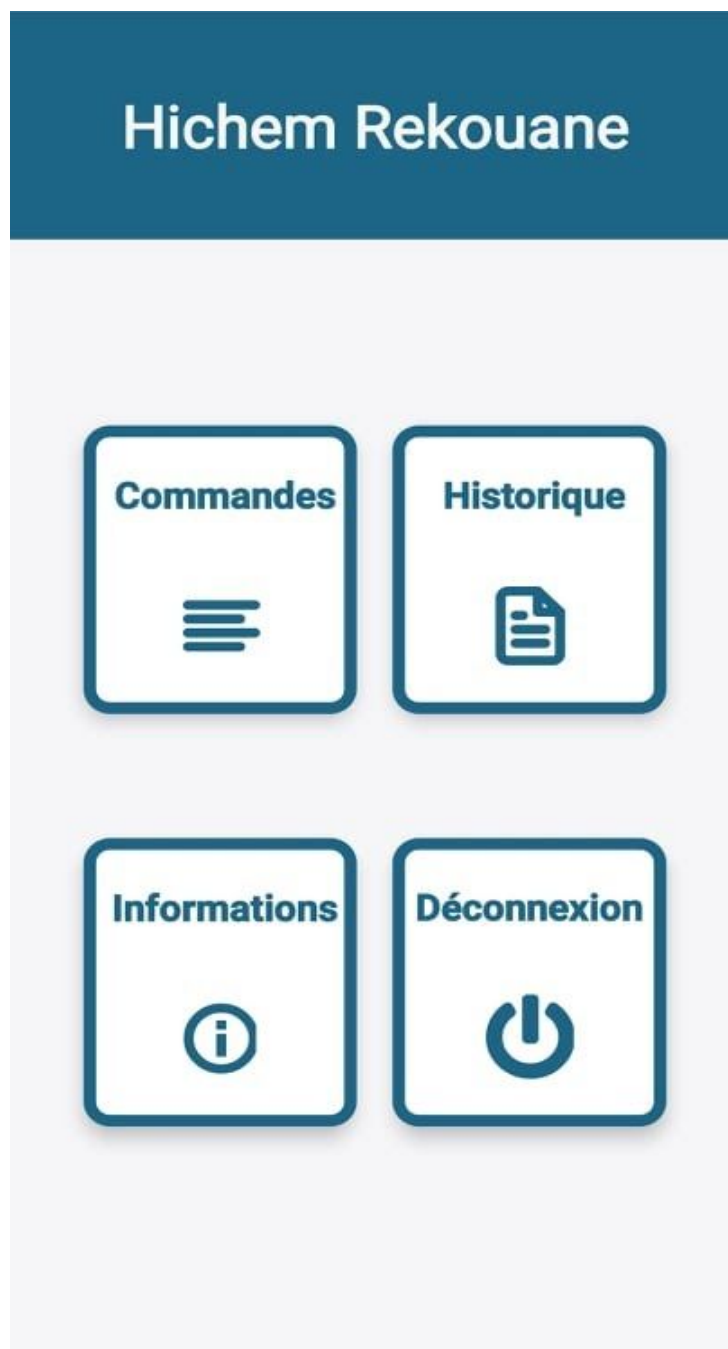
 E-mail

 Mot de passe

CONNEXION

*Figure 10: Interface d'authentification du vendeur.*

### 2.1.2. Interface d'accueil



*Figure 11: Interface d'accueil pour de l'application du vendeur.*

### 2.1.3. Interface des commandes

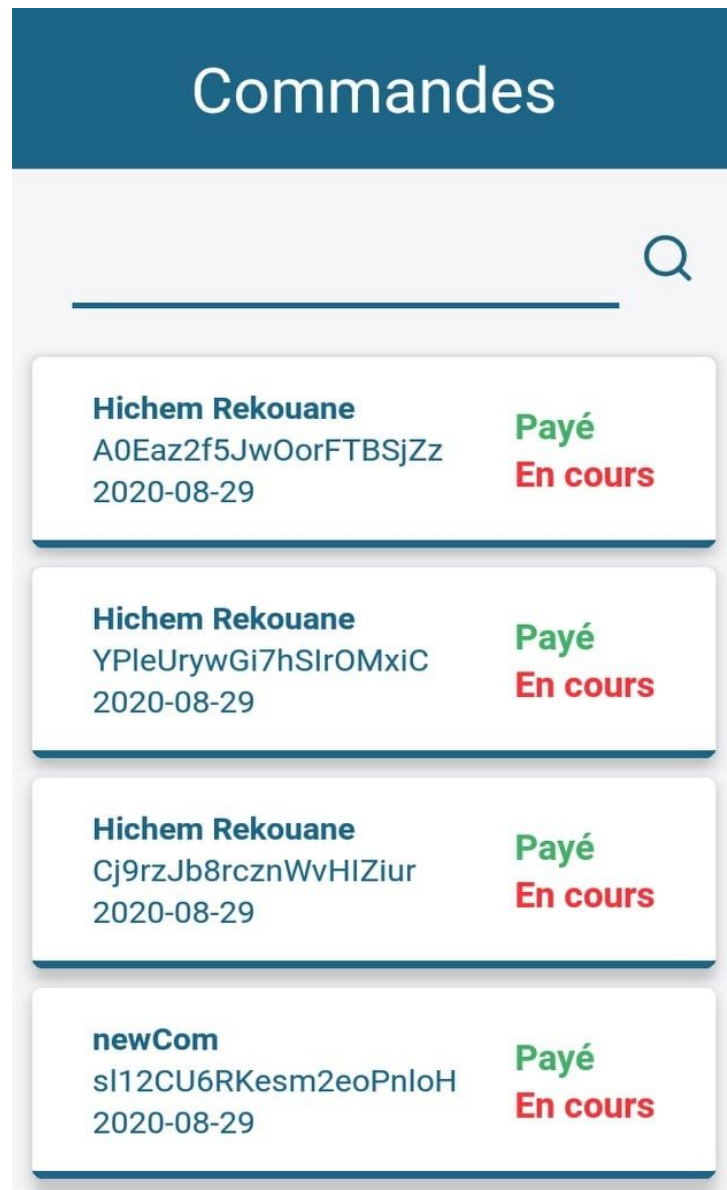


Figure 12: Interface des commandes de l'application vendeur

#### 2.1.4.Interface Détails des commandes

## Détaille de la commande

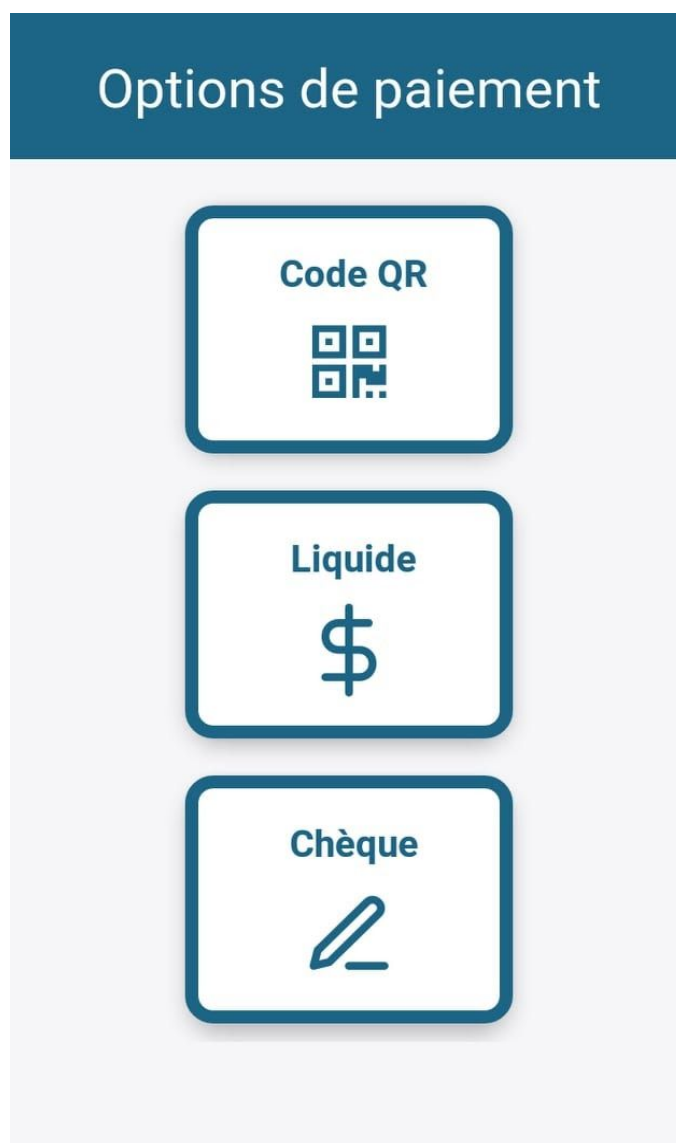
Ref	CtwVkpoOx4fuNjYviFaZ
Au nom de	Hichem Rekouane
Somme à payer	91000 DA
Etat de livraison	En cours
Etat de paiement	Non payé

OPTIONS DE PAIEMENT

Figure 13: Interface détails des commandes  
de l'application du vendeur



### 2.1.5. Interface options de paiement



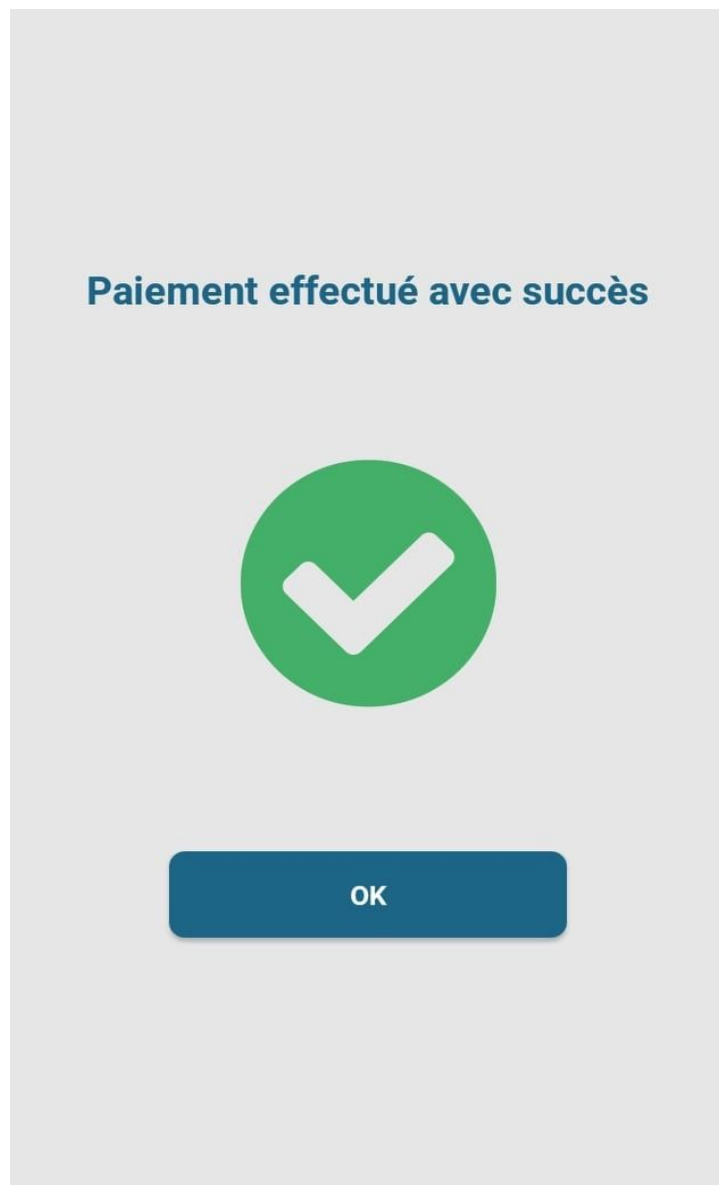
*Figure 14: Interface des options de paiement de l'application vendeur.*

### 2.1.6. Interface du paiement par QR code



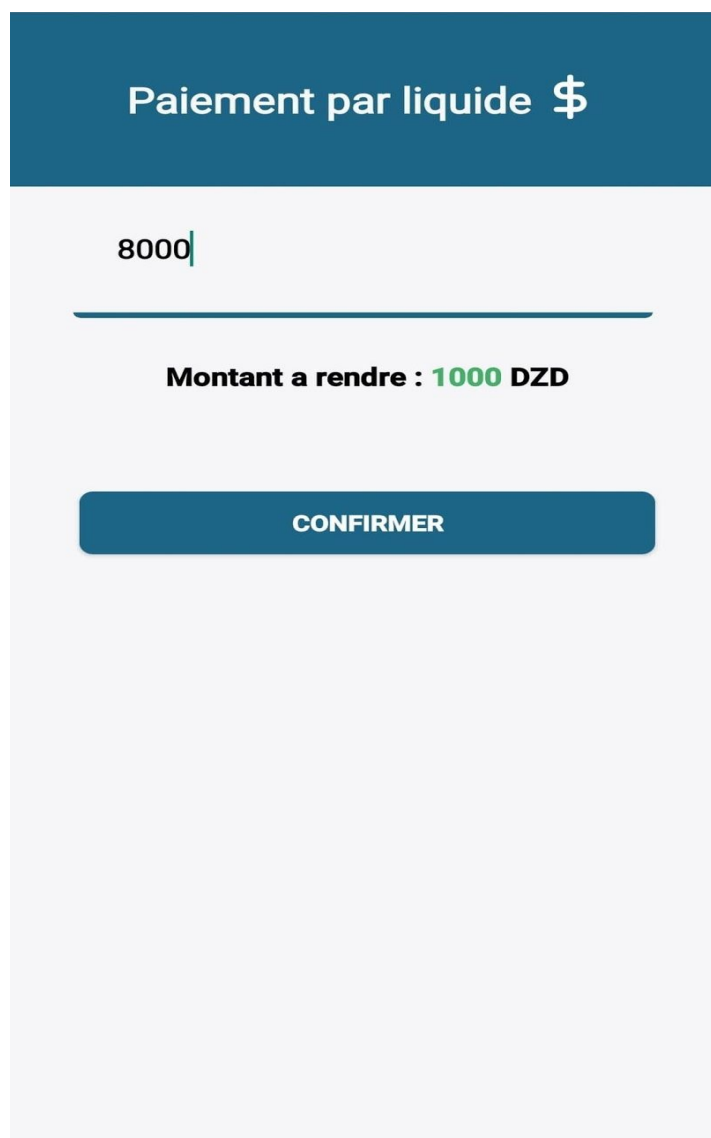
*Figure 15: Interface du paiement par QR code de l'application vendeur.*

### 2.1.7. Interface de la confirmation du paiement par code QR



*Figure 16: Interface de la confirmation du paiement par QR code de l'application vendeur.*

### 2.1.8. Interface du paiement en liquide



Paieement par liquide \$

8000

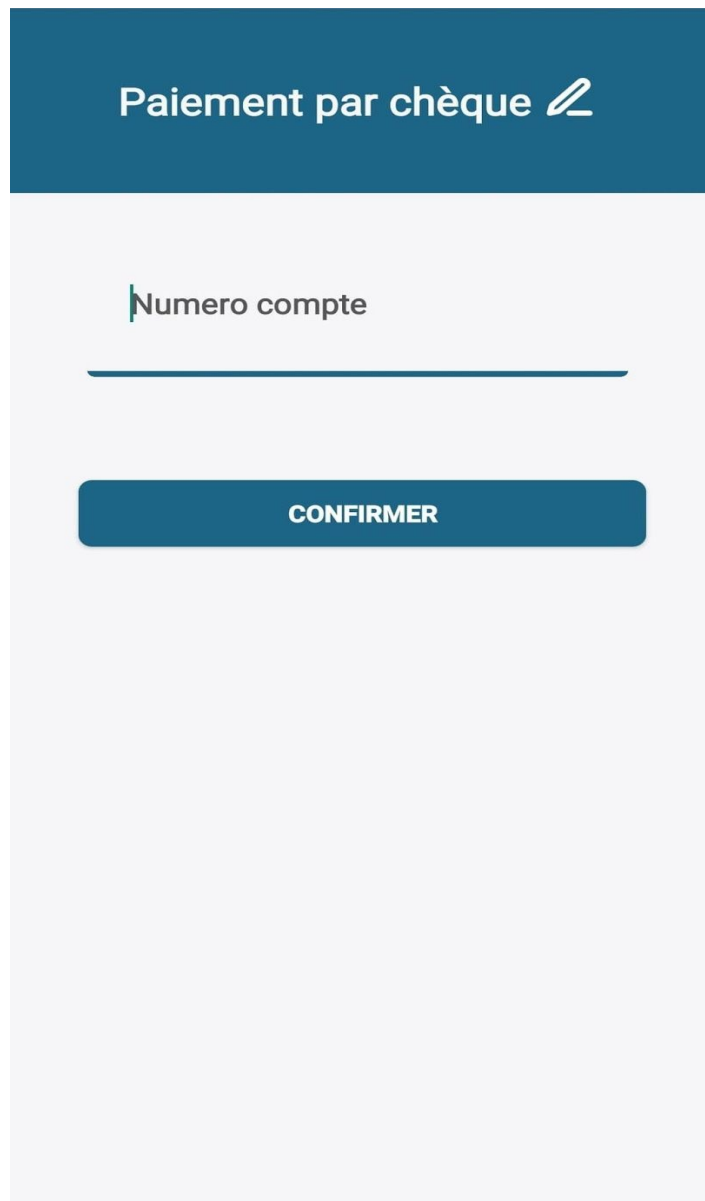
---

Montant a rendre : 1000 DZD

CONFIRMER

*Figure 17: Interface du paiement en liquide de l'application vendeur.*

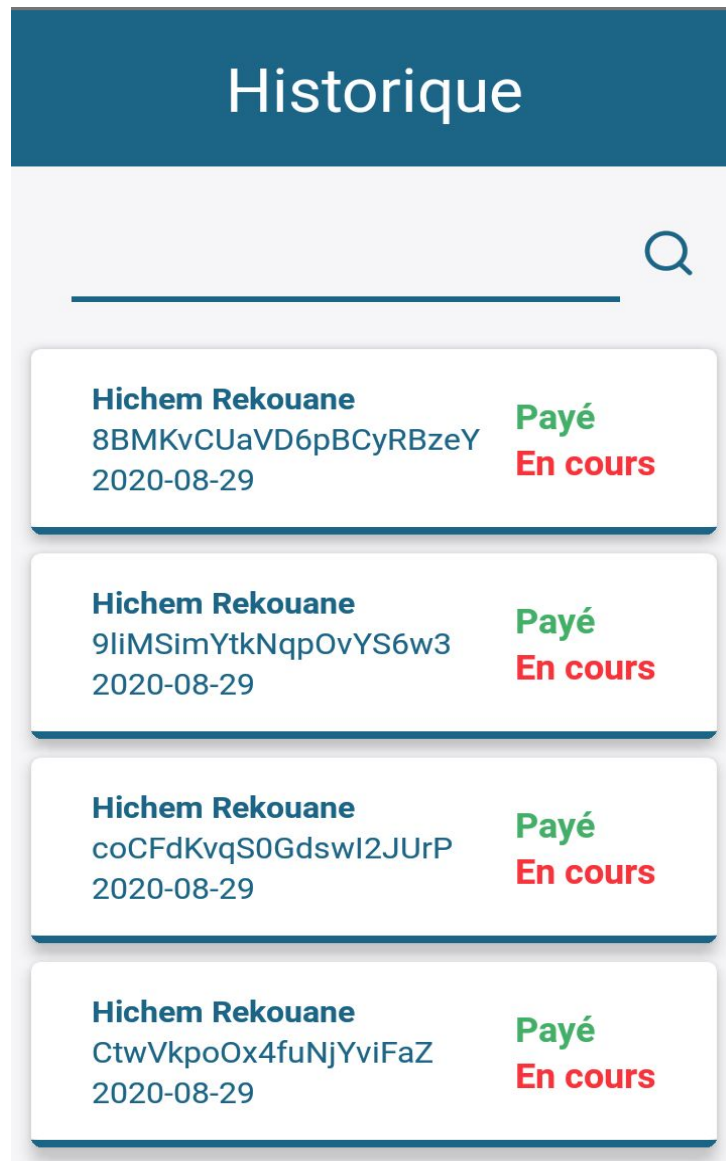
### 2.1.9. Interface du paiement en chèque



The screenshot displays a mobile application interface for making a payment by check. At the top, a dark blue header bar contains the text 'Paieement par chèque' followed by a check icon. Below this, the label 'Numero compte' is positioned above a horizontal input field. A blue underline is visible beneath the input field. At the bottom of the interface, there is a prominent blue button with the text 'CONFIRMER' in white capital letters.

*Figure 18: Interface du paiement en chèque de l'application vendeur.*

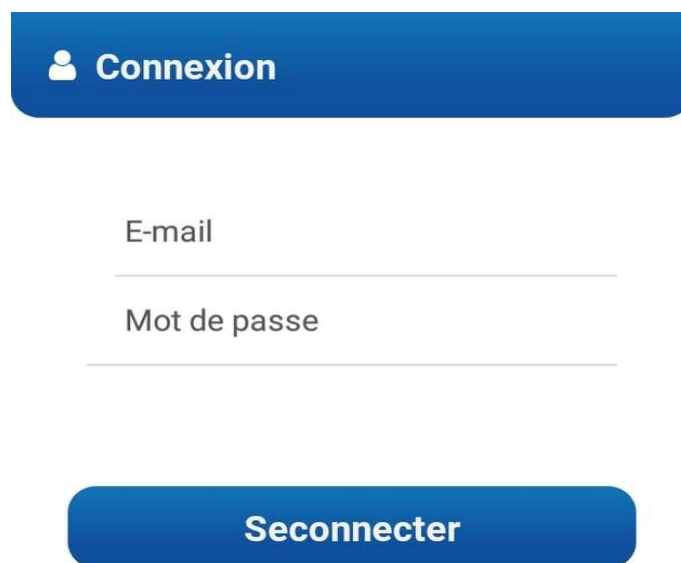
## 2.1.10. Interface de l'historique des transactions



*Figure 19: Interface de l'historique des transactions de l'application vendeur.*

## 2.2. L'application portefeuille mobile pour le client de l'entreprise

### 2.2.1. Interface d'authentification



The image shows a mobile application interface for authentication. At the top, there is a blue header bar with a white user icon and the text "Connexion". Below this, there are two input fields: "E-mail" and "Mot de passe". Each field has a horizontal line underneath it. At the bottom, there is a blue button with the text "Seconnecter".

*Figure 20: Interface d'authentification du client*

### 2.2.2. Interface d'accueil

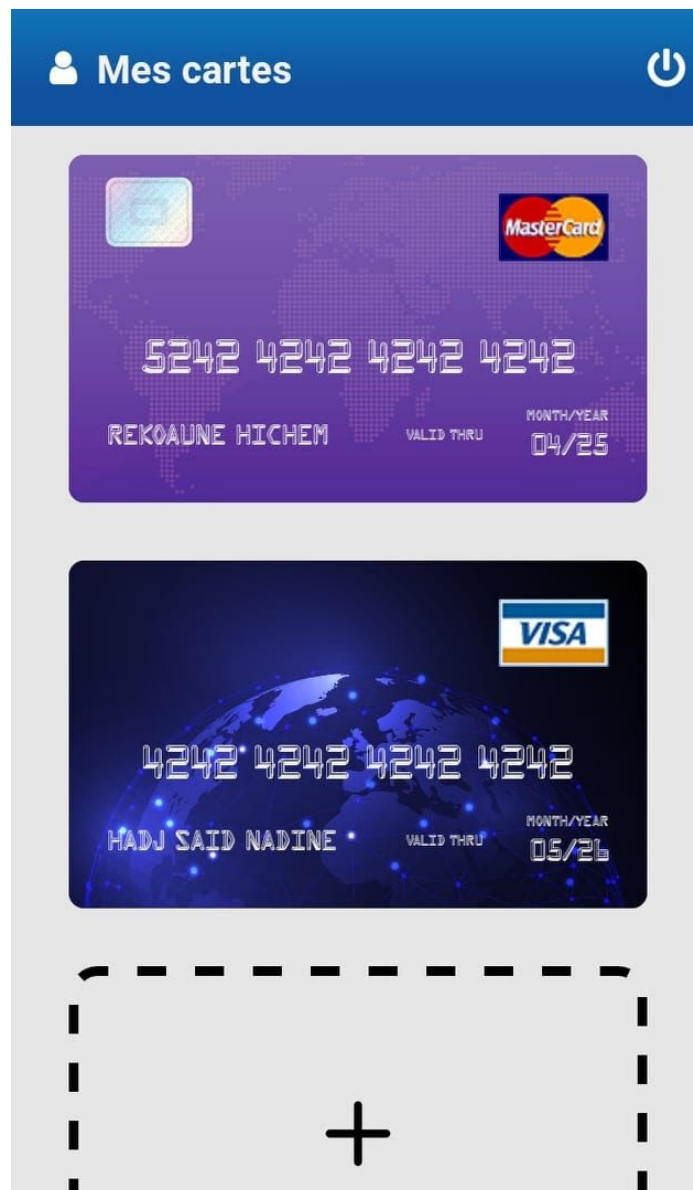


Figure 21: Interface d'accueil du client



### 2.2.3. Interface de détails de la carte et proposition de paiement

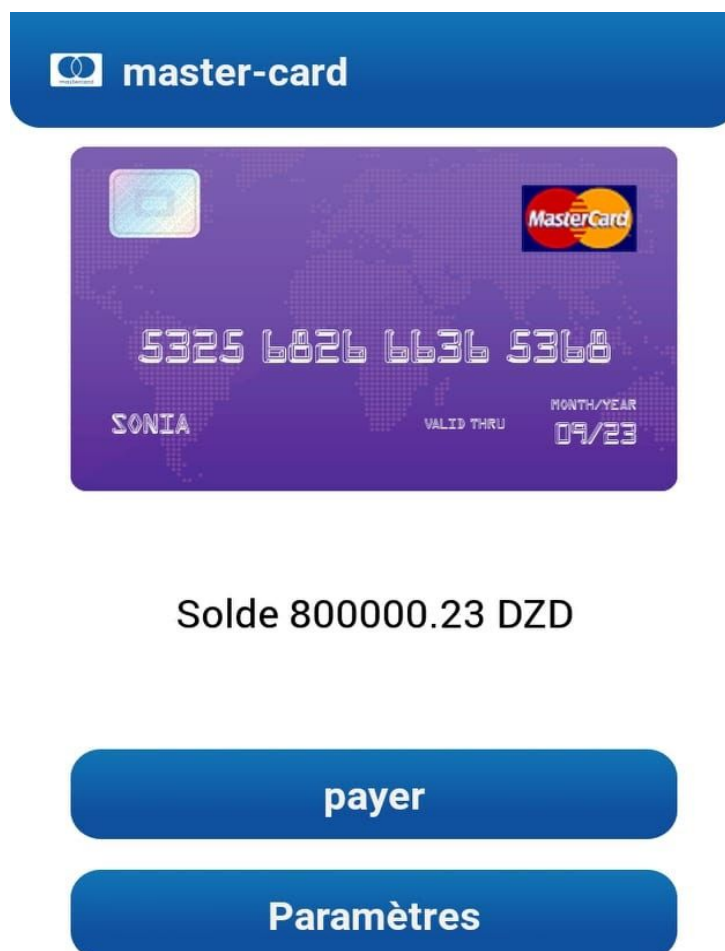



Figure 22: Interface détails de la carte du client

#### 2.2.4. Interface d'ajout d'une carte

**+ Ajouter une carte**



exp	cvc	Nom Sur la carte
04/25	653	Hadj said nadine

**Confirmer**

Figure 23: Interface d'ajout d'une carte.

# Conclusion

Dans ce dernier chapitre, nous avons présenté la dernière phase de la conception de notre système à savoir la réalisation, ou nous avons cité et défini les outils logiciels que nous avons utilisé pour développer notre solution, et nous avons aussi présenté nos deux applications mobiles à travers leurs interfaces utilisateurs.

## Conclusion générale

L'objectif de notre travail était de développer une solution de paiement mobile pour une entreprise commerciale quelconque, permettant aux futurs utilisateurs de payer leurs commandes en utilisant seulement leur appareil mobile.

Pour commencer notre travail, il nous a fallu faire une étude théorique sur tous les outils nécessaires pour le développement d'une application mobile.

En second terme, nous avons étudié le paiement mobile et ses différents types, afin de comprendre leur fonctionnement et pouvoir choisir le type qui convient le mieux à notre cas d'étude.

Ensuite, nous avons utilisé le langage de modélisation UML qui nous a permis de décrire le plan d'élaboration et construction de notre application afin de concevoir cette dernière.

Enfin, pour réaliser notre solution, plusieurs technologies nous ont été nécessaires, on citera le framework de développement d'application mobile open-source React Native, le framework Laravel etc.

Après le passage par les différentes étapes précédemment citées, notre solution a abouti à des logiciels fonctionnels qui répondent globalement aux attentes des utilisateurs pour leur permettre d'effectuer un paiement avec leurs smartphones de façon pratique et rapide, qui comporte une application mobile sur la plateforme Android destinée au vendeur de l'entreprise, et une application de portefeuille mobile pour le client.

La solution que nous proposons présente certaines limites, dont la non connexion à un vrai système informatique de banque, en effet le paiement effectué avec cette solution est un paiement simulé et ne peut s'appliquer dans le monde réel, en plus de sa disponibilité sur la plateforme Android uniquement. Elle reste d'ailleurs en développement dans le but d'avoir un vrai système totalement utilisable par les entreprises commerciales et leurs clients, et peut être améliorée comme suit:

- ❖ Connecter l'application portefeuille mobile à un vrai système de banque.
- ❖ Ajouter d'autres couches de sécurité au niveau du paiement.
- ❖ Rendre les deux applications disponibles sur la plateforme iOS.
- ❖ Ajouter d'autres types de paiement mobile comme le paiement NFC.

Le travail que nous avons accompli nous a amené à approfondir nos connaissances, sur les technologies de développement mobile, le génie logiciel, les architectures logicielles,

les APIs Restful, le modèle client/serveur ainsi que les systèmes de gestion de bases de données, le déploiement continu, le développement d'applications temps réel, etc.

# Webographie

- ❖ [1] Wikipédia 06 mai 2020 “Application mobile”  
[https://fr.wikipedia.org/wiki/Application\\_mobile](https://fr.wikipedia.org/wiki/Application_mobile) [consulté le 11/05/2020].
- ❖ [2] Matthew Panzarino 13 février 2015, “A History of Mobile Apps By AVG Technologies”:<https://www.theguardian.com/media-network/2015/feb/13/history-mobile-apps-future-interactive-timeline>, [consulté le 11/05/2020].
- ❖ [3] Lucas Dominic, “How to Define E-Payments”:  
<https://securionpay.com/blog/how-to-define-e-payments/>, [consulté le 11/05/2020].
- ❖ [4] Dr Nikolas Beutin and Maximilian Harmsen, Mars 2019 “Mobile payment report 2019”<https://www.pwc.de/de/digitale-transformation/pwc-studie-mobile-payment-2019.pdf>, [consulté le 29/05/2020]
- ❖ [5] Luke Vu, 15 January 2020: “Main Advantages of Mobile Payment Systems”:  
<https://www.simicart.com/blog/mobile-payment-systems-advantages/>, [consulté le 24/05/2020]
- ❖ [6] Julie le jeu, 02/08/2018 - 09:35: “Paiement mobile bonne ou mauvaise idée?”:  
<https://www.echosdunet.net/dossiers/paiement-mobile-bonne-ou-mauvaise-idee>,  
consulté le 23/05/2020.
- ❖ [7] Jamie Roberts, April 7th, 2013 “How do SMS payments work for users?”  
<https://www.mobiletransaction.org/how-do-sms-payments-work-for-users/>, [consulté le 28/05/2020].
- ❖ [8] [Robert Gibb](#) 31 mai 2016 “What is a Web Application?”:  
<https://blog.stackpath.com/web-application/> [consulté le 23/05/2020]

- ❖ [9] Techopedia 05 février 2018 “Client/Server Architecture”:<https://www.techopedia.com/definition/438/clientserver-architecture> [consulté le 23/05/2020].
  
- ❖ [10] Kony “Mobile Application Platform”  
<https://www.kony.com/resources/glossary/mobile-application-platform/> [consulté le 11/05/2020]
  
- ❖ [11] Maximilian Schwarzmüller 20 décembre 2017: “React Native vs Ionic vs NativeScript vs Android/ iOS Native Apps”:[https://www.youtube.com/watch?v=rb8smP\\_xTTY](https://www.youtube.com/watch?v=rb8smP_xTTY) [consultée le 12/05/2020].
  
- ❖ [12] Brad Travesty 04 Août 2017: “Mobile Apps - Web vs. Native vs. Hybrid”:  
<https://www.youtube.com/watch?v=ZikVtdopsfY> [consultée le 12/05/2020].
  
- ❖ [13] Martin Aiguinis 17 Septembre 2019 “How is Flutter different for app development” <https://www.youtube.com/watch?v=I-YO9CmaSUM&t=149s> [consultée le 12/05/2020]
  
- ❖ [14] Squareup “What Are Mobile Payments? And How to Use Them”:  
<https://squareup.com/guides/mobile-payments> [consulté le 14/05/2020]
  
- ❖ [15] emerchantpay:” Everything you need to know about mobile payments”:  
<https://www.erchantpay.com/insights/everything-you-need-to-know-about-mobile-payments/> [consulté le 13/05/2020]
  
- ❖ [16] Taneleo “Les avantages et inconvénients du paiement par smartphone”  
<http://taneleo.fr/paiement-par-smartphone-avantage-inconvenients/> [consulté le 13/05/2020].
  
- ❖ [17] Chris Griffith “What is Apache Cordova?”  
<https://ionicframework.com/resources/articles/what-is-apache-cordova>, [consulté le 12/05/2020].

- ❖ [18] William Craig “Native App vs. Mobile Web App: A Quick Comparison”  
<https://www.webfx.com/blog/web-design/native-app-vs-mobile-web-app-comparison/>  
 [consulté le 13/05/2020].
  
- ❖ [19] Laurence Georges 15 octobre 2015 “Quelle plateforme pour le développement de votre application mobile?”:  
<https://www.ideematic.com/actualites/2015/10/quelle-plateforme-pour-le-developpement-de-votre-application-mobile/> [consulté le 13/05/2015]
  
- ❖ [20] Wikipédia 30 Avril 2020 “iOS”: <https://en.wikipedia.org/wiki/IOS> [consulté le 11/05/2020]
  
- ❖ [21] Wikipédia 10 mai 2020 “Android (operating system)”  
[https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)) [consulté le 11/05/2020]
  
- ❖ [22] Marc Knoll 4 décembre 2012 “Mobile Payment Overview: Definition, Trends And Payment Systems”  
<https://trendblog.net/mobile-payment-overview-definition-trends-and-payment-systems/> [consulté le 23/05/2020].
  
- ❖ [23] Margaret Rouse Août 2019 “Application architecture”:  
<https://searcharchitecture.techtarget.com/definition/application-architecture>  
 [consulté le 31/05/2020].
  
- ❖ [24] Margaret Rouse Mars 2020 “Middleware”:  
<https://searcharchitecture.techtarget.com/definition/middleware>, [consulté le 31/05/2020].
  
- ❖ [25] Sylvie SERVIGNE, « SYSTÈMES INFORMATIQUES - Conception, architecture et urbanisation des systèmes d'information », Encyclopædia Universalis [en ligne]:  
<http://www.universalis.fr/encyclopedie/systemes-informatiques-conception-architecture-et-urbanisation-des-systemes-d-information/> [consulté le 2 juin 2020]



- ❖ [26] Franck Ledoux 2003 -2004 “DESS Bio -Informatique”:  
<https://www.cours-gratuit.com/cours-uml/cours-genie-logiciel-et-uml> [consulté le 02/06/2020.]
  
- ❖ [27] Pr F. Bouarab-Dahmani département informatique UMMTO 2018/2019 cour  
:”Assurance et Contrôle Qualité du Logiciel”, [consulté le 02/06/2020].
  
- ❖ [28] Stéphane Jorge,” Qu'est-ce que le langage UML (langage de modélisation unifié)  
?”: <https://www.lucidchart.com/pages/fr/langage-uml> [consulté le 05/06/2020]
  
- ❖ [29] Pierre Gérard IUT de Villeteuse DUT informatique, S2  
2013:<https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html> [consulté le 05/06/2020]
  
- ❖ [30] Mathias Péron, cours “Analyse, Conception et Validation de Logiciels” INP  
Grenoble 2006-2007:” <http://mathias.xn--pron-bpa.eu/docs/teaching/acvl/cours.pdf>”,  
[consulté le 06/06/2020].
  
- ❖ [31] Yann-Gaël Guéhéneuc © Guéhéneuc, 2012 Département de génie informatique  
et de génie logiciel École Polytechnique de Montréal “Architectures – Partie 1/2 “:  
“[http://www.ptidej.net/courses/log4430/winter12/slides/Cours%201%20-%20Concepti  
on%20architecturale%201.pdf](http://www.ptidej.net/courses/log4430/winter12/slides/Cours%201%20-%20Concepti%20on%20architecturale%201.pdf)”, [consulté le 10/06/2020].
  
- ❖ [32] Lydie dBousquet cour: GL - 2 Architecture logicielle”  
“[http://membres-lig.imag.fr/dubousquet/docs/2.4\\_Architecture.pdf](http://membres-lig.imag.fr/dubousquet/docs/2.4_Architecture.pdf)”, [consulté le 12/06/2020].
  
- ❖ [33] Rémy Manu “UML : Langage de modélisation objet unifié Cours n°3 :  
Diagramme des classes “: <http://remy-manu.no-ip.biz/UML/Cours/coursUML3.pdf>”  
[consulté le 12/06/2020].
  
- ❖ [34] Pierre Gérard IUT de Villeteuse DUT informatique, S2 2013 : “UML Cours 5 :  
Diagramme de séquences”:  
<https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours05.html>. [consulté le 12/06/2020].

- ❖ [35] Laurent Dufour, le 13/09/2019, “Le Cahier Des Charges”:  
<https://www.leblogdudirigeant.com/le-cahier-des-charges>, [consulté le 06/06/2020].
- ❖ [36] Bonnie Eisenman “Learning React Native”:  
[“https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html”](https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html)  
[consulté le 15/06/2020].
- ❖ [37] MDN web docs “JavaScript”:<https://developer.mozilla.org/fr/docs/Web/JavaScript>  
[consulté le 15/06/2020].
- ❖ [38] Domantas G, 25/11/2019 “What is HTML? The Basics of Hypertext Markup Language Explained ”: [“https://www.hostinger.com/tutorials/what-is-html”](https://www.hostinger.com/tutorials/what-is-html) [consulté le 17/06/2020].
- ❖ [39] Łukasz Mądrzak 11/12/2019: “What is Laravel? Explain it like I'm five”:  
[https://dev.to/creativetim\\_official/what-is-laravel-explain-it-like-i-m-five-19eb](https://dev.to/creativetim_official/what-is-laravel-explain-it-like-i-m-five-19eb) [consulté le 15/06/2020].
- ❖ [40] Lyna Griffin 25/11/2017 “What is PHP Used For? - Uses Advantages”:<https://study.com/academy/lesson/what-is-php-used-for-uses-advantages.html> [consulté le 15/06/2020].
- ❖ [41] C.SOUTOU. Apprendre SQL avec MySQL Avec 40 exercice corrigés. Paris, groupe Eyrolles, 2006.
- ❖ [42] Ansgar Becker “What's this?”: <https://www.heidisql.com>. [consulté le 15/06/2020]
- ❖ [43] Antoine Crochet-Damais, 01/02/19: “Visual Studio Code, l'IDE open source que les développeur s'arrachent”:  
<https://www.journaldunet.com/web-tech/developpeur/1204957-visual-studio-code-l-ide-open-source-que-les-devs-s-arrachent/> [consulté le 15/06/2020].
- ❖ [44] [Aditya Sridhar](#) 12/08/2018 “An introduction to Git: what it is, and how to use it”:  
<https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/>  
[consulté le 15/06/2020].

- ❖ [45] Parikshit Hooda “Introduction to Postman for API Development”:  
<https://www.geeksforgeeks.org/introduction-postman-api-development/>. [consulté le 15/06/2020].
  
- ❖ [46] Margaret Rouse, Juin 2012 “Heroku”:  
“<https://searchcustomerexperience.techtarget.com/definition/Heroku>” , [consulté le 15/06/2020].
  
- ❖ [47] Phill Legget, 19 Juillet 2012, “Pusher Overview - Pusher Realtime Web Workshop” [https://www.youtube.com/watch?v=\\_3S9Z3NZKTU](https://www.youtube.com/watch?v=_3S9Z3NZKTU) , [consulté le 15/06/2020].
  
- ❖ [48] Samia Bouzefrane, “Introduction aux systèmes temps réels”.  
“<https://cedric.cnam.fr/~bouzeфра/cours/Introduction.pdf>” ,[consulté le 15/06/2020].
  
- ❖ [49] débitoor, Paiement par carte bancaire - Qu'est-ce qu'un paiement par carte bancaire:  
“[https://debitoor.fr/termes-comptables/paiement-carte-bancaire#:~:text=Paiement%20par%20carte%20bancaire%20%2D%20Qu.ses%20achats%20chez%20un%20comm er%C3%A7ant.&text=Une%20CB%20\(carte%20bancaire\)%20peut.%2C%20t%C3%A9l%C3%A9phone%2C%20par%20correspondance](https://debitoor.fr/termes-comptables/paiement-carte-bancaire#:~:text=Paiement%20par%20carte%20bancaire%20%2D%20Qu.ses%20achats%20chez%20un%20comm er%C3%A7ant.&text=Une%20CB%20(carte%20bancaire)%20peut.%2C%20t%C3%A9l%C3%A9phone%2C%20par%20correspondance)” , [consulté le 05/09/2020]
  
- ❖ [50] journaldunet.fr, [Micropaiement : définition, traduction](#), 30.04.2018  
“<https://www.journaldunet.fr/business/dictionnaire-economique-et-financier/1199329-micropaiement-definition-traduction/>” , [consulté le 05/09/2020].
  
- ❖ [51] Bernard ESPINASSE Professeur à l'Université d'Aix-Marseille, 2009  
“ Introduction à UML (Unified Modelling language) “:  
<http://www.lsis.org/espinasseb/Supports/MR-TC/IntroULM-sept09-4p.pdf>, [consulté le 05/09/2020].
  
- ❖ [52] Paymill, “Systèmes de paiement électronique”  
<https://www.paymill.com/fr/glossaire/e-payment-system/> [consulté le 05/09/2020].
  
- ❖ [53] Margaret Rouse, “Déploiement continu (continuous deployment, CD)” :  
<https://www.lemagit.fr/definition/Deploiement-continu> , [consulté le 05/09/2020].

- ❖ [54] adnews by Galit, 05.08.2014 “CopSonic présente sa solution de paiement mobile par ondes sonores” <https://adn-co.news/actualites/details/6961> [consulté le 05/09/2020].
- ❖ [55] BFM, 05/04/2019 à 11h23 “Le virement instantané par mobile se généralise”:  
<https://rmc.bfmtv.com/emission/le-virement-instantane-par-mobile-se-generalise-1667108.html> ,[Consulté le 05/09/2020].
- ❖ [56] Paylib, “Virement par mobile ou virement bancaire entre particuliers”:  
<https://www.paylib.fr/paiement-entre-amis/virement-par-mobile-ou-virement-bancaire-entre-particuliers/> ,[Consulté le 05/09/2020].