

Table des matières

Introduction générale	9
Chapitre I.....	10
Fondements théoriques.....	10
I. Introduction	11
II. E-business	11
III. E-commerce.....	11
III.1 Définition	11
III.2 M-commerce	12
IV. Marketplace	12
IV-1. Définition.....	12
IV-2. Acteurs dans un Marketplace	13
IV -3. Typologie des relations commerciales	15
IV-4 Les cinq piliers du marketplace.....	15
IV-5. Avantages et inconvénients du marketplace	16
V. Relation entre marketplace, e-business & e-commerce.....	17
VI. Conclusion.....	17
Chapitre II	18
Etat de l'art	18
I. Introduction	19
II. Comparatifs de marketplace	19
III. Critiques :	20
IV. Suggestion	21
V. Conclusion	21
Chapitre III	22
Conception de l'application	22
I. Introduction	23
II. Processus de Développement.....	23
II.1. Cycle de vie en V	23
II.2 Diagramme de gant	24
II.3 la méthode agile scrum.....	26
III Patron de conception « Design Patterns »	29
III.1 Patron de création « Patterns de Création »	29

III.2 Patron de fabrication « Design Pattern factory »	29
III.3 Patron de conception « Pattern de conception »	30
IV Analyse	31
IV.1 Diagramme de cas d'utilisation	31
IV.1.2. Cas d'utilisation marchand	33
IV.1.3 Cas d'utilisation administrateur	35
IV.2 Diagramme de séquence système	37
V. Conception	42
V.1. Diagramme de séquence	42
V.2. Modèle physique de donnée	46
V. Conclusion	47
Chapitre IV	48
Réalisation de l'application	48
I. Introduction	49
II. Environnement de logiciel	49
III. Environnement de développements	50
III.1 Plateformes	50
III.2 Logiciels	52
III.3 Technologies utilisées	53
IV. Pourquoi IONIC et pourquoi la programmation Hybride ?	54
V. Prototypes	54
V.1 Architecture	54
VI. Caractéristiques	59
VI.1 Caractéristiques fonctionnelles	59
VI.2 Caractéristiques non fonctionnelles	60
VII. Conclusion	60
Chapitre V	61
Manuel de l'application	61
I. Démonstration des interfaces	62
Conclusion générale	82
References biographiques	84

Table des figures

Figure 1: Les acteurs de marketplace. [4]	13
Figure 2: Relation entre e-commerce, e-business et marketplace.....	17
Figure 3:Interface Ebay.....	20
Figure 4:Interface Cdiscount.....	21
Figure 5: Schéma cycle de vie en V. [11]	23
Figure 6: Diagramme de gant.....	25
Figure 7: Requête SQL complexe conditionnelle.....	27
Figure 8: Patron de fabrication "factory".....	29
Figure 9: Patron MVC. [16]	30
Figure 10: Architecture trois tiers MVC.....	31
Figure 11: Diagramme de cas d'utilisation acheteur.....	32
Figure 12: Diagramme de cas d'utilisation marchand.....	33
Figure 13: Diagramme cas d'utilisation administrateur.....	35
Figure 14: Diagramme de séquence système login.....	37
Figure 15: Diagramme de séquence système inscription.....	38
Figure 16: Diagramme de séquence système ajouté article.....	39
Figure 17: Diagramme de séquence système géré adresse.....	40
Figure 18: Diagramme de séquence système consulter achat.....	41
Figure 19: Diagramme de séquence ajouté article.....	42
Figure 20: Diagramme de séquence acheté article.....	44
Figure 21: Modèle physique de données.....	46
Figure 22: Certification Apple.....	51
Figure 23: Plateforme Apple Developer.....	51
Figure 24: la vue de haut niveau de l'application.[34]	54
Figure 25: Schéma des fonctionnalités de l'application.....	55
Figure 26: Architecture de système.....	57
Figure 27: Emplacement boutiques avec DiscountDice.....	58
Figure 28: Création d'entrepôts donnés.....	58
Figure 29: Interface inscription IOS.....	62
Figure 30: Interface inscription Android.....	62
Figure 31: Interface authentification IOS.....	63
Figure 32: Interface authentification Android.....	63
Figure 33: Interface accueil "sidebar" IOS.....	64
Figure 34: Interface accueil "sidebar" Android.....	64
Figure 35: Interface accueil Android.....	65
Figure 36: Interface accueil client IOS.....	65
Figure 37: Interface détail article IOS.....	65
Figure 38: Interface détail article Android.....	65
Figure 39: Interface validation achat Android.....	66
Figure 40: Interface validation achat IOS.....	66
Figure 41: Interface facture IOS.....	67
Figure 42: Interface facture Android.....	67
Figure 43: Interface ticket Android.....	68
Figure 44: Interface ticket IOS.....	68

Figure 45: Interface accueil marchand IOS.....	69
Figure 46: Interface accueil marchand Android.....	69
Figure 47: Interface article IOS.....	70
Figure 48: Interface article Android.....	70
Figure 49: interface historique IOS.....	71
Figure 50: Interface historique Android.....	71
Figure 51: Interface accueil administrateur "sidebar" Android.....	72
Figure 52: Interface accueil administrateur "sidebar" IOS.....	72
Figure 53: Interface tableau de bord Android.....	73
Figure 54: Interface tableau de bord IOS.....	73
Figure 55: Interface notification article IOS.....	74
Figure 56: Interface notification utilisateur.....	74
Figure 57: Interface ressource humaine.....	75
Figure 58: Interface ressource humaine IOS.....	75
Figure 59: interface article marchand Android.....	76
Figure 60: interface article marchand IOS.....	76
Figure 61: Interface informations par fournisseur Android.....	76
Figure 62: Interface informations par fournisseur IOS.....	76
Figure 63: Interface contrat Android.....	77
Figure 64: Interface de traçabilité des articles et tickets validés IOS.....	77
Figure 65: Interface catégorie Android.....	78
Figure 66: Interface catégorie IOS.....	78
Figure 67: Interface ajouté sous-catégorie Android.....	78
Figure 68: Interface ajouté sous-catégorie IOS.....	78
Figure 69: Interface algorithme IOS.....	79
Figure 70: Interface algorithme Android.....	79
Figure 71: Interface configuration des pays Android.....	80
Figure 72: Interface configuration des pays IOS.....	80
Figure 73: Interface tester gain d'article IOS.....	80
Figure 74: Interface tester gain d'article Android.....	80

Table des tableaux

Tableau 1: Tableau des différents acteurs. [4]	13
Tableau 2: comparatif dans marketplace.....	19
Tableau 3: les taches de diagramme de gant.	25
Tableau 4: Description de cas d'utilisation acheteur.	32
Tableau 5: description cas d'utilisation marchand.....	33
Tableau 6: Description de cas d'utilisation administrateur.....	36
Tableau 7: Configuration de téléphone et serveur.	49

Introduction générale

Le secteur des applications mobiles est actuellement en plein croissance grâce aux multiples fonctionnalités natives qu'il offre, ainsi que l'ergonomie spéciale qui respecte parfaitement la résolution des Smartphones. Contrairement même aux sites web responsifs où l'utilisateur reste avec les mêmes possibilités que les sites web classiques. Cette tendance offre aux entreprises de formidables opportunités pour créer des nouveaux services et élargir ces supports de communications puisque les Smartphones et les tablettes sont prêts à détrôner les ordinateurs à travers le monde.

Le marché des applications mobiles touche les différents domaines et plus particulièrement le domaine de commerce. Ce dernier a connu un succès très remarquable, en passant de la vente de main à main vers la vente virtuelle, par l'intermédiaire d'un ensemble d'opérations successives. La vente en ligne fournit aux clients un pont de passage à l'ensemble des informations, des produits et des services à partir d'une vaste foire virtuelle des magasins.

Chaque magasin souhaite améliorer la visibilité de ces produits afin d'accueillir de nouveaux clients et développer une meilleure relation avec eux. En leurs facilite la communication et ils peuvent être en contact avec les magasins à tout moment pour un bon service client et un commerce plus efficace. Et pour un rendu plus élevé, il faut respecter les besoins des acheteurs, qui cherchent toujours de nouveaux produits, avec un renseignement de leurs qualités, leurs caractéristiques et surtout de leur réduction de prix. Les clients veulent aussi acheter sans penser au problème de distance géographique ni des horaires de travail.

Le nombre des utilisateurs de Smartphones augmente du jour à un autre. Par conséquent l'application mobile Marketplace permettra aux magasins de mieux se positionner sur le marché et être une vitrine pour leurs clients.

Notre projet, qui est réalisé dans le cadre d'un mémoire de master, a comme objectif la conception et la réalisation d'un marketplace sous forme d'une application mobile avec des nouvelles technologies en utilisant la programmation hybride. Nous allons détailler le projet dans ce rapport à travers quatre chapitres :

- Le chapitre 01 va présenter la notion de Marketplace et ses bases : e-commerce et e-business, ses intervenants ainsi que les typologies des relations commerciales, ensuite quelques avantages et inconvénients.
- Le chapitre 02 va présenter une étude comparative de quelques marketplaces existantes.
- Le chapitre 03 va présenter la conception et la réalisation de l'application.
- Le chapitre 04 va présenter l'implémentation de l'application en décrivant l'environnement logiciel et de développement, et donnera un aperçu sur les interfaces réalisées.

Et nous terminerons ce document par une conclusion générale qui résume notre travail avec des perspectives d'avenir.

Rapport-Gratuit.com

Chapitre I

Fondements théoriques

I. Introduction

Les innovations dans le domaine de développement informatique se multiplient et s'évaluent sans cesse. Ce qui nous pousse toujours à trouver des moyens plus efficaces pour répondre à plusieurs besoins afin de faciliter notre vie.

La technologie récente est plus en plus dans le domaine de commerce en ligne, De vente de main à main vers des ventes virtuelles.

II. E-business

E-business est une stratégie basée sur des technologies de l'information et de communication en ligne pour mettre en réseau et habilitier les processus d'affaires, commerce électronique, communication organisationnelle et collaboration dans une entreprise et avec ses clients, ses fournisseurs et tous les partis prenantes dans le but d'augmenter la valeur de l'entreprise grâce à la motivation du personnel et l'utilisation des différents actions pour concrétiser une vente, par la suite une assurance de la fidélisation client de la vente en ligne. [1]

E-business met sous contrôle les différents processus internes et externes de l'entreprise y compris les relations efficaces avec les fournisseurs, les clients et les partenaires (ressource humaines, vente, achat, stock et production), afin de l'aider à travailler plus intelligemment et satisfaire le besoin des utilisateurs d'une manière plus facile. [2]

III. E-commerce

III.1 Définition

E-commerce (commerce électronique) est une sous activité d'e-business qui ne se concentre que sur la procédure d'achat et non pas sur la fidélisation clientèle c'est-à-dire un ensemble des transactions commerciales entre deux entités sur des réseaux informatiques. Ces entités peuvent être des entreprises, entreprise et un particulier ou deux particuliers.

Le commerce électronique s'appuie sur plusieurs technologies basé sur le marketing Internet, les systèmes de gestion des stocks, le traitement des transactions en ligne et beaucoup d'autres systèmes. [2]

Les entreprises de commerce peuvent être une partie ou même tout élément des sites d'achat en ligne pour les particuliers ou entre entreprise pour échanger les données et les services, elles peuvent même participer et être un élément de marketplace.

III.2 M-commerce

Le M-commerce est un sous ensemble d'e-commerce. Il englobe l'achat et la vente de produits et services effectués à partir d'un Smartphone ou d'une tablette, le commerce mobile permet de mieux localiser les positions des utilisateurs et faire des affaires à tout moment en toute sécurité grâce à la combinaison des différentes technologies telle que les systèmes embarqués, la mise en réseau et l'envoi des données plus rapidement. [3]

IV. Marketplace

IV-1. Définition

Les marketplaces sont l'avenir de e-commerce, principalement un canal de vente en ligne sur lequel se rencontrent les acheteurs et les vendeurs pour échanger leurs biens et leurs services en s'assurant que les transactions se dérouleront dans les meilleures conditions en toute transparence et sécurité, géré par l'opérateur marketplace.

L'opérateur marketplace met en relation les acheteurs et les vendeurs en leur mettant à disposition des outils et des services pour faciliter la mise en relation et les échanges comme les vitrines, la gestion de stock et le service client. Dans une marketplace on peut trouver deux types d'opérateurs :

- **Hybride** : c'est un opérateur et commerçant en même temps, il peut poser ces articles et les vendre sur la plateforme
- **Pure Player** : c'est juste un opérateur intermédiaire, il n'est pas un concurrent des vendeurs sur la plateforme [4]

Des espaces de vente sont réservés pour chaque marchand afin de profiter des fonctionnalités de la plateforme d'e-commerce, on peut distinguer deux approches de marketplace : [4]

- **Généraliste** : offre une grande gamme de produits dans plusieurs catégories.
- **Verticale** : se concentre sur un secteur d'activité particulier.

IV-2. Acteurs dans un Marketplace

Dans un marketplace on distingue trois acteurs principaux acteurs et d'autres secondaires, qui ont des rôles spécifiques et complémentaires afin de concrétiser une vente. La figure 01 représente les différents acteurs :

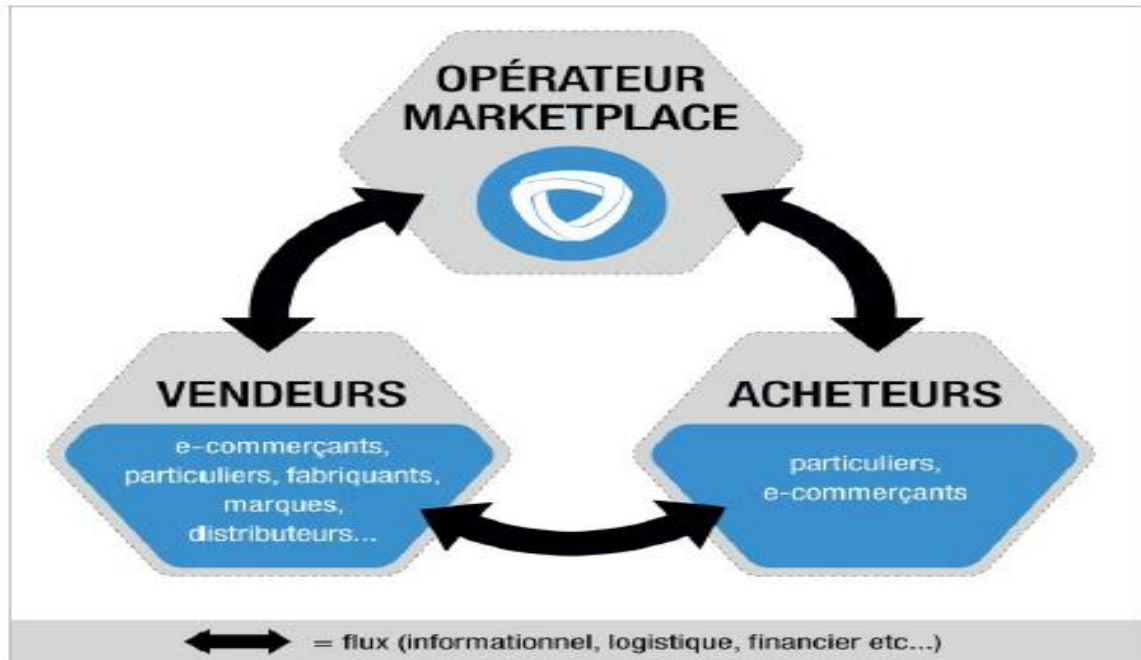


Figure 1: Les acteurs de marketplace. [4]

Le tableau 1 représente la relation entre les acteurs ainsi que leurs rôles : [4]

Tableau 1: Tableau des différents acteurs. [4]

Acteurs principaux	Rôles
Opérateurs	Sont des acteurs qui gèrent la plateforme et garantissent une confiance aux utilisateurs. Ils mettent les vendeurs et les acheteurs en relation, et garantissent la fluidité des échanges afin d'offrir une meilleure transaction.
Vendeur	Sont des personnes présentes dans Marketplace pour vendre leurs produits et leurs services, ils peuvent être des vendeurs professionnels ou non professionnels. Leur intérêt est d'accueillir des nouveaux clients en présentant leurs produits aux meilleurs prix avec le bon service.
Acheteur	Peuvent être des particuliers ou des entreprises qui prennent en charge la politique d'achat en consultant une vaste sélection de produits proposés par une diversité de vendeurs.
Acteurs secondaires	Rôles
Editeurs de solution	Sont des acteurs qui construisent Marketplace et proposent des solutions techniques. Pour ajouter des fonctionnalités techniques, il existe des plugins CMS

	proposé par des logiciels de gestion de contenu d'e-commerce afin de transformer un site de vente en ligne en marketplace.
Intégrateurs de flux	Leurs solutions assurent d'exporter les catalogues des marchands sur des différents marketplace. Ils se chargent aussi d'adapter les informations de chaque produit (article, prix, quantité, disponibilité...) sur chaque support e-commerce.
Logisticiens	Sont responsables d'un bon acheminement des commandes afin de satisfaire le client.

Marketplace met en disposition des espaces de vente virtuelles à louer pour le marchand (via internet), ces derniers sont les clients de fondateurs de cette plateforme. Le marchand loue un espace afin de présenter ses articles en respectant les lois imposés par les éditeurs de solutions (créateurs de marketplace).

Et comme on a présenté précédemment à la définition que marketplace est une plateforme donc obligatoirement elle a des manipulateurs tels que les opérateurs qui font la gestion des ressources (des articles) de la plateforme.

Marketplace a aussi une relation avec les intégrateurs de flux qui ont comme rôle la publication des catalogues des marchands associés, ainsi une relation avec les logisticiens tels que les responsables de la livraison des achats.

IV -3. Typologie des relations commerciales

Les entreprises et les consommateurs sont les deux acteurs plus courants au commerce électronique. Il existe quatre types principaux d'e-commerce, qu'ils ont cités dans la source suivante [5]:

B2C (Business to Consumer) : commerce électronique entre entreprise et un grand public grâce à des catalogues en ligne. C'est la forme de e-commerce qui connaît la plus forte progression.

B2B (Business to Business) : commerce électronique entre des entreprises, elle est plus ancienne que B2C

C2B (Consumer to Business) : commerce électronique dont un consommateur est au service de l'entreprise, comme appels d'offres d'emplois d'un projet.

C2C (Consumer to Consumer) : commerce électronique entre deux consommateurs sur des sites qui offrent des annonces sur internet, ou peut-être même deux particuliers peuvent vendre et acheter en ligne.

IV-4 Les cinq piliers du marketplace

Une marketplace efficace repose sur cinq piliers nécessaires : [4]

- **le trafic:** est le pilier le plus important dans une marketplace, car cette dernière est en relation directe avec le trafic, plus il y a des visiteurs plus les opportunités de transactions seront nombreuses et la visibilité des articles des vendeurs augmente.
- **Le mix produit :** on peut distinguer deux types de marketplace :
 - Des marketplaces avec une faible limite imposée aux vendeurs et la restriction des quasi-inexistants, le vendeur peut poser tous types d'articles.
 - Des marketplaces où les opérateurs définissent précisément quels produits mis en vente avec une maîtrise de catalogue, chaque article doit respecter le catalogue qui lui convient.
- **Le cadre légal :** les marketplaces n'imposent pas le même niveau d'exigence au niveau juridique. Pour certaines marketplaces les contrats sont bien contrôlés pour les échanges des articles sensibles. Et pour d'autres cette exigence est moindre, l'identification des intervenants n'est pas trop poussée.
- **Les outils de communication et de la gestion de cycle de vente :**
 - Les outils de communication entre les acheteurs et les vendeurs sont le facteur le plus importants de la réussite de concrétisation d'une vente et de l'après-vente, ça peut être des messageries, chat, Skype et beaucoup d'autres outils.
 - Pour la gestion du cycle de vente, les marketplaces affichent aux vendeurs la quantité de stocks de ses produits mise à jour au temps réel. Ainsi que les opérateurs suivent les transactions d'achat avec tous les états d'avancement de la commande jusqu'à la réception d'article par le client.

- **Le paiement** : cette partie est très sensible, elle doit être bien Contrôlée. pour avoir la confiance du client il faut leur assurer que les opérateurs enregistrent la transaction dans un environnement sécurisé, et que leurs données sont bien protégées.

IV-5. Avantages et inconvénients du marketplace

Il faut bien connaître les avantages et inconvénients de marketplace par apport aux deux principaux acteurs, l'entreprise et le client : [2]

IV-4-1. Avantages

Pour l'entreprise :

- Permet d'envisager la politique de la fidélisation du client à travers une offre des services.
- Nouveau canal de distribution qui permet de se développer sur le marché mieux qu'un magasin physique.
- Meilleure économie en évitant les coûts des locaux des magasins.
- Améliorer la visibilité des produits, donc c'est un marketing très efficace.
- Permet de toucher des nouveaux clients et développer une relation entre eux et l'entreprise.

Pour le client :

- Plus de choix des produits et avec des prix moins chers.
- Economie du temps puisque le client peut acheter sans se déplacer.
- Un renseignement sur la qualité de produit grâce aux descriptions bien détaillées.
- Des économies grâce à la possibilité de comparer les prix.
- Possibilité d'acheter n'importe où et n'importe quelle heure (24h/24 et 7j/7).
- Pas de file d'attente.
- Shopping de la maison.

IV.4.2 Inconvénients

Pour l'entreprise :

- L'insécurité des transactions et le manque de confiance des moyens de paiement face aux pirates.
- Des frais supplémentaires ou la nécessité d'une bonne formation du personnel.
- -l'anonymat du client.
- Coût logistique plus élevé que le commerce classique comme les frais des transports.

Pour le client :

- L'insécurité de paiement et la possibilité de tomber sur des sites malveillants.
- En cas de problème c'est difficile de contacter le service après-vente.
- le manque de contact avec le produit et par la suite le client ne peut pas bien se rendre compte sur ses qualités et ses défauts.
- Crainte sur la qualité de service (la livraison, la qualité,...).
- Absence de contact avec le vendeur et par la suite manque de ses conseils.

V. Relation entre marketplace, e-business & e-commerce

La réalisation de **marketplace** se base principalement sur l'**e-business** et l'**e-commerce**, comment ça ? La figure 02 représente la relation entre ces derniers.



Figure 2: Relation entre e-commerce, e-business et marketplace.

Pour une bonne planification des techniques technologiques et de nouvelles règles **e-business** qui mènent à la fidélisation des clients d'une manière différente et plus efficace ainsi une manière plus intelligente pour organiser les opérations et les ressources internes de marketplace (analyse de donnée, motivation de personelles et la gestion des ressources humaine ...etc.) Afin de mettre notre client (marchand) à l'aise et gagner sa confiance, vu que ces règles vont être présentées et publiées sur le marché, et par la suite attirent le maximum de clients.

Ensuite mettre en disposition de marchand sa propre plateforme qui va être sa propre boutique en ligne en lui garantissant une puissante logistique telle que les prestataires de paiement sécurisés afin de réaliser une bonne solution **e-commerce** qui convient à l'utilisateur finale.

Et pour conclure la marketplace est l'avenir d'e-commerce car ce modèle est aujourd'hui le meilleur moyen de faire de l'**e-commerce** de manière rentable et avec une forte croissance.

VI. Conclusion

Aujourd'hui, de nouvelles technologies émergent, de nouveaux outils, de nouveau concepts qui permettent d'enrichir le commerce électronique tel que l'e-business et les marketplaces.

Nous avons présenté chaque concept (e-commerce, e-business et marketplace) ainsi la relation entre eux, nous avons aussi constaté quelques inconvénients malgré les nombreuses avantages qu'ils offrent.

Chapitre II

Etat de l'art

I. Introduction

L'étude de l'existant est une étape très importante. Elle se définit par la recherche des applications similaires à notre projet, pour but de traiter et améliorer les fonctionnalités qui existent et chercher d'autres qui ne sont pas déjà intégrés.

II. Comparatifs de marketplace

Dans le tableau 2 on présente un comparatif de quelques concurrents marketplace : **Amazon**, **Cdiscount** et **Ebay**, suite à notre recherche et analyse des références suivantes [6] [7] [8] [9] [10].

Tableau 2: comparatif dans marketplace.

	Amazon	EBay	Cdiscount
Transaction	Prix fixe	Prix fixe	Prix fixe
Occasion	Oui	Oui	Oui
Vente à l'international	Oui	Oui	Non
Livraison	Oui	Oui	Oui
Abonnement mensuel	39 €	17€ à 260€	39.99 €
Unique visiteurs par jour	1934000	1207000	836000
Unique visiteurs par mois	16 544 000	9172000	10099000

Le tableau 2 montre une comparaison entre les principaux marketplace, cette comparaison fait suite à des critères. :

On remarque qu'**Amazon** couvre le plus grand nombre de visiteurs mensuel. Le critère occasion se trouve à tous les marketplaces ainsi que la vente à l'international sauf **Cdiscount** qu'elle n'a pas.

Les transactions de tous les marketplace sont avec des prix fixes. La livraison est disponible chez tous les marketplaces et Les frais d'abonnement se change d'une marketplace à une autre.

III. Critiques :

On a pris deux marketplaces EBay et Cdiscount afin d'étudier leurs points faibles :

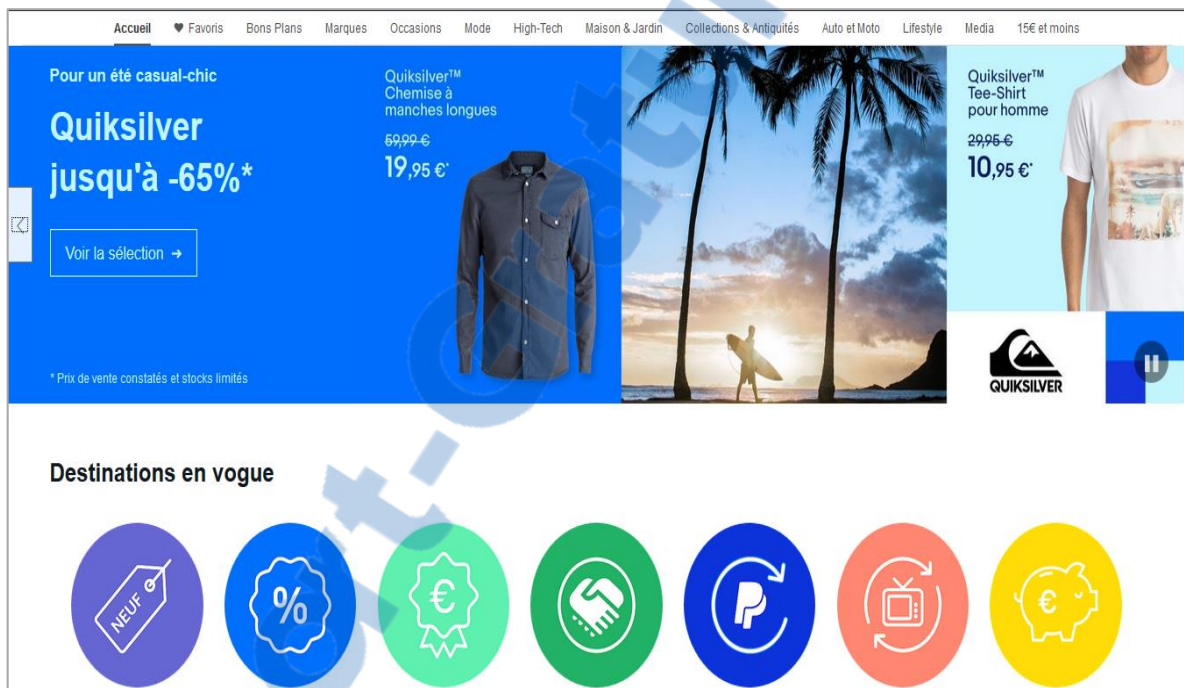


Figure 3: Interface Ebay.

Pour Ebay, les vendeurs sont soumis des frais mensuel ce qui est un peu dure pour les nouveaux dans l'e-commerce pour attirer les acheteurs. Ainsi des espaces de vente impersonnel ce qui est difficile de différencier entre les boutiques des vendeurs car elles se ressemblent.

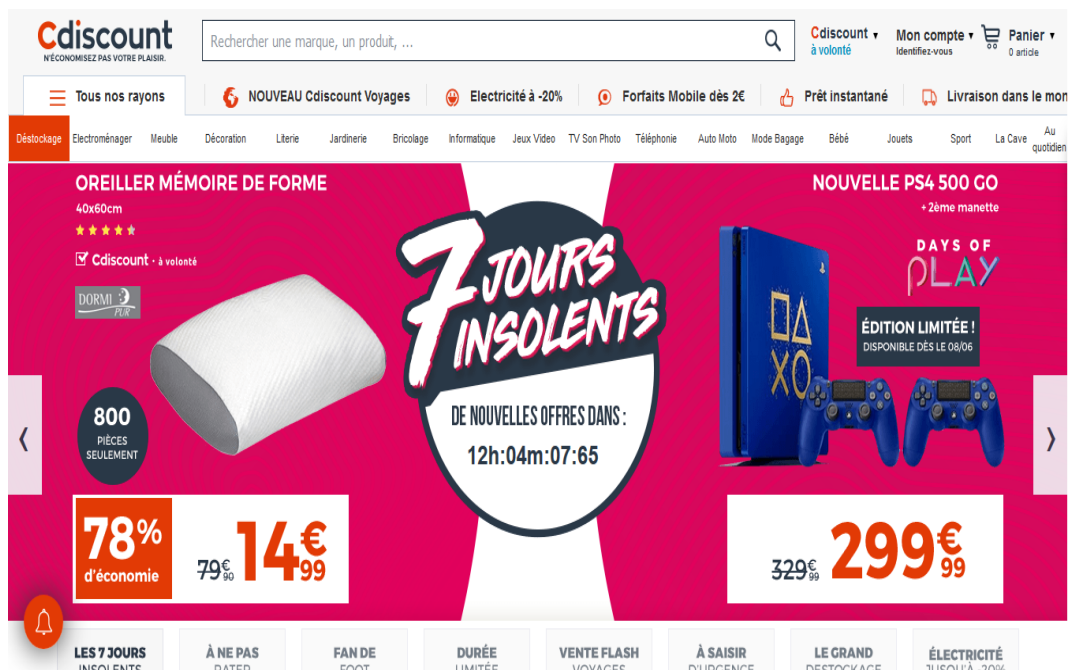


Figure 4: Interface Cdiscount.

Pour Cdiscount, L'application est trop lente au chargement. Les vendeurs sont soumis à des frais mensuels ce qui est un peu dure pour les nouveaux dans l'e-commerce pour attirer les acheteurs. Ainsi que les annonces publiées ne sont pas ciblé au grand public à l'international donc y'a pas des acheteurs étrangers.

IV. Suggestion

Nous proposons une application qui a pour but de créer des boutiques de vente sur la plateforme marketplace, et respecter l'image de chaque magasin en ajoutant leurs noms et leurs logos sur leurs articles afin que l'acheteur puisse différencier entre les boutiques et de laquelle il va acheter. Et surtout faire des économies pour les vendeurs (pas de frais mensuel). Et des prix dynamiques des articles et non pas fixe pour les vendeurs.

V. Conclusion

L'étude des systèmes existants nous permet d'analyser les projets similaires afin d'améliorer les fonctionnalités et savoir ce qui manque pour les ajouter à notre projet pour donner un maximum de services nécessaires.

Chapitre III

Conception de l'application

I. Introduction

Afin d'une bonne gestion de projet et de mieux structurer notre travail, nous avons décidé à faire une conception basée sur les diagrammes afin de mieux comprendre et schématiser notre système pour but de faciliter le développement et avoir un travail efficace.

II. Processus de Développement

II.1. Cycle de vie en V

Le model de cycle de vie en V est actuellement l'outil de gestion de projet le plus utilisé. Deux types de taches sont réalisés en même temps : verticalement, on prépare l'étape suivante et horizontalement, on prépare la vérification de la tâche en cours. Les étapes de test sont nombreuses ce qui rend le risque de l'erreur moins élevé et c'est pour cela que nous avons choisis cet outil durant notre processus de développement. [11]

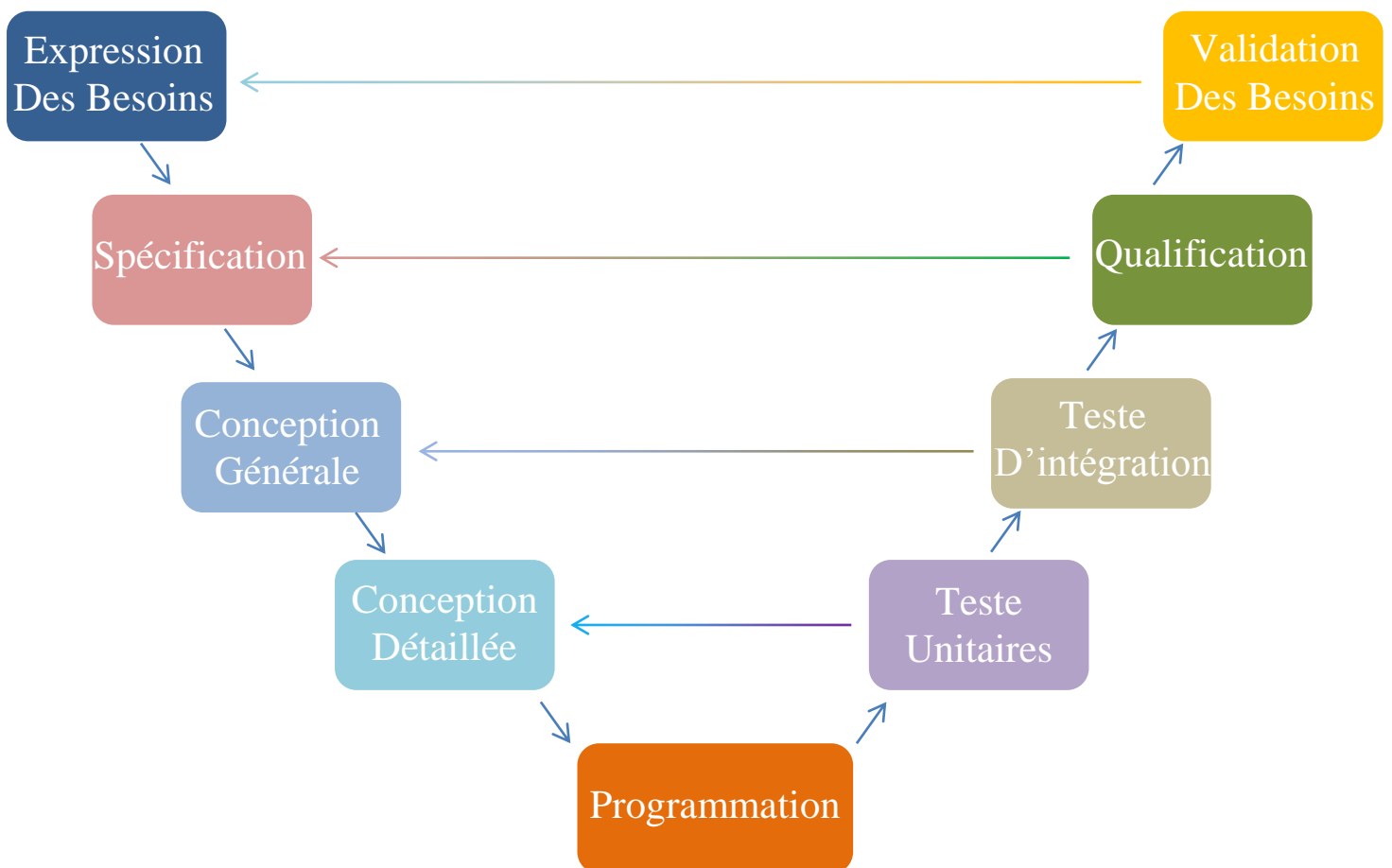
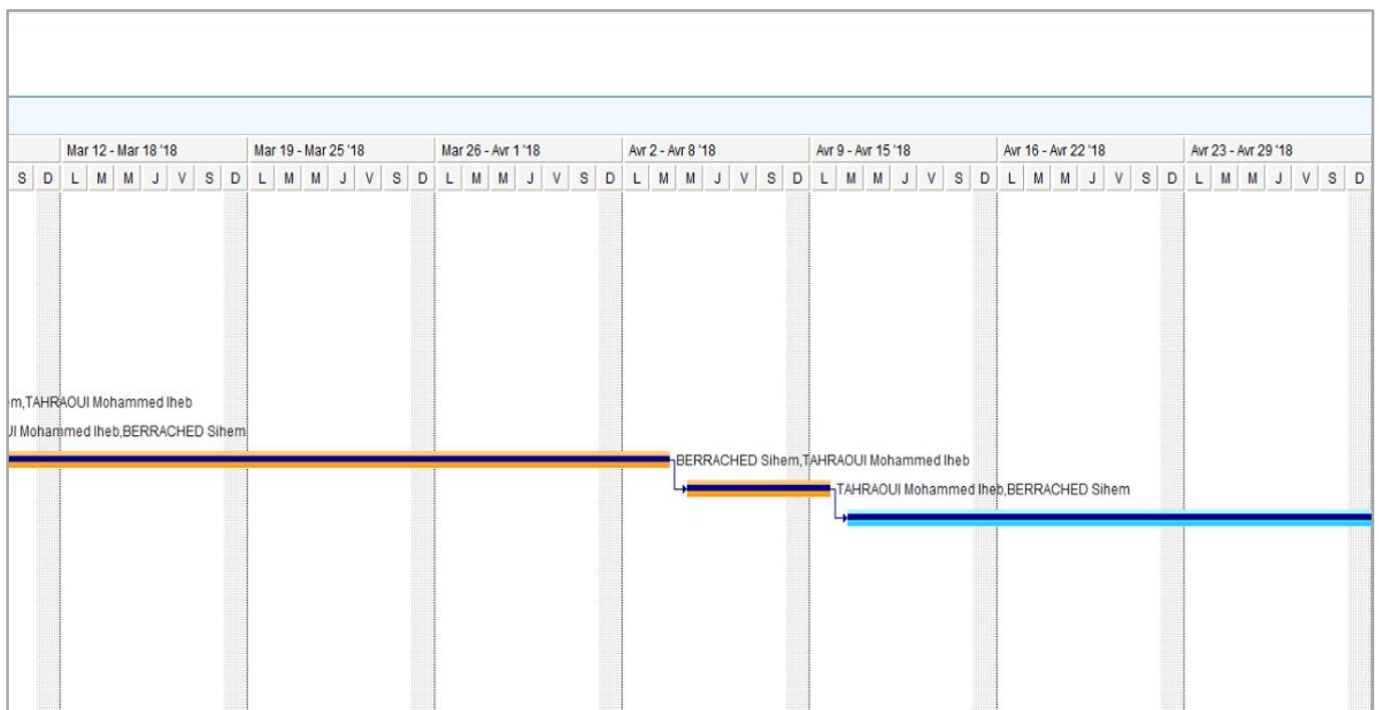
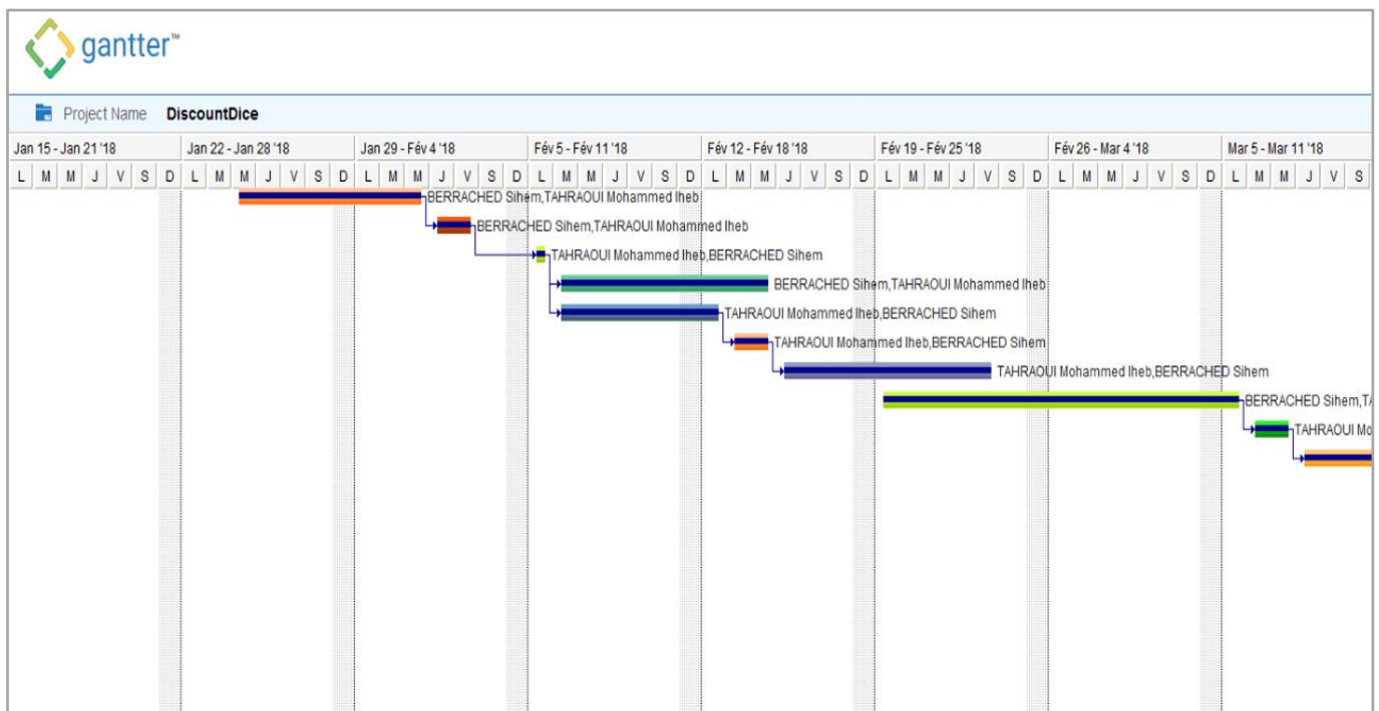


Figure 5: Schéma cycle de vie en V. [11]

II.2 Diagramme de gant

Le diagramme de Gant est un outil de planification pour gérer les projets quel que soit le secteur d'activité, et partage l'état d'avancement des taches et des sous taches au temps réel. [12]

Nous avons utilisé ce diagramme pour la réalisation de notre projet qui est représenté dans la [figure 06](Nous l'avons divisé pour une bonne visibilité).



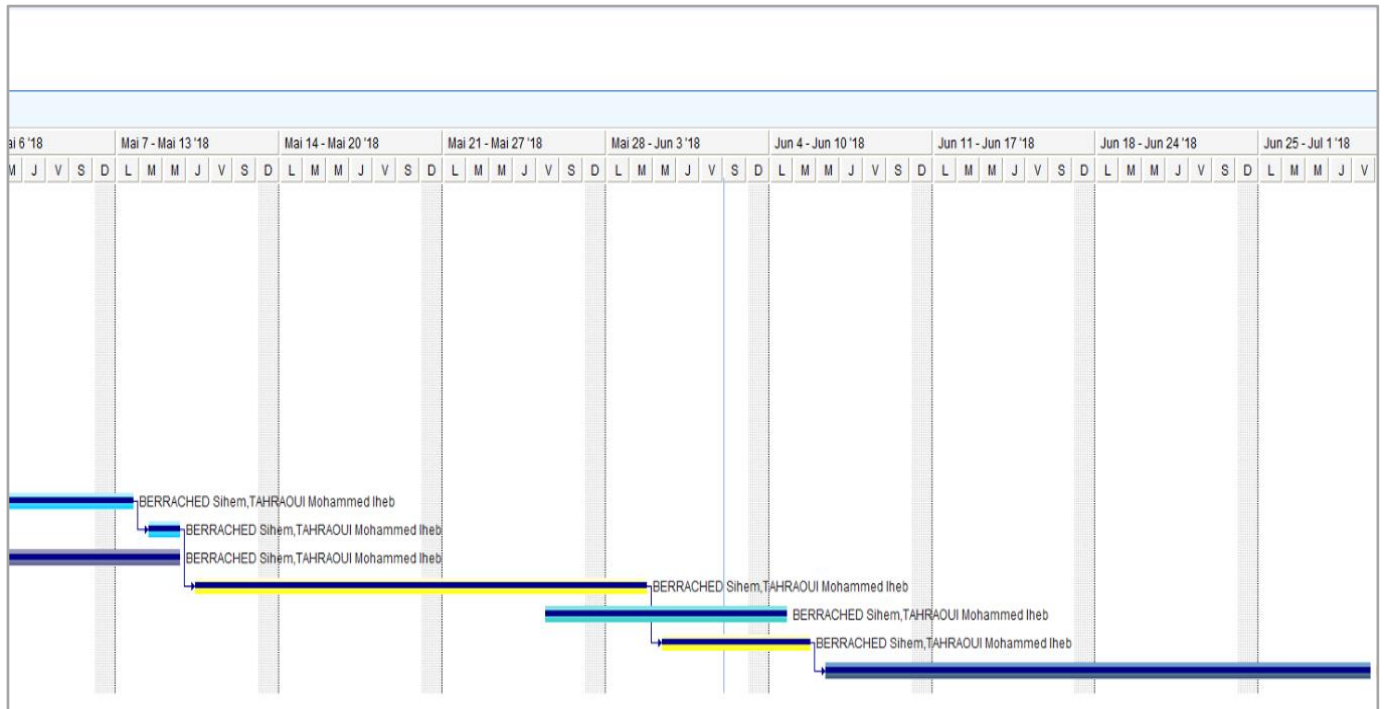



Figure 6: Diagramme de gant.

Le tableau 3 représente les descriptions des taches mentionné dans le diagramme en respectant leurs couleurs :

Tableau 3: les taches de diagramme de gant.

							
Project Name DiscountDice							
		Nom	Durée	Début	Fin	Prédécesseurs	Ressources
1		Etudier les besoins	6journées	24/01/2018	31/01/2018		BERRACHED Sihem,TAHRAOUI Mohammed Iheb
2		Prioriser les besoins	2journées	01/02/2018	02/02/2018	1	BERRACHED Sihem,TAHRAOUI Mohammed Iheb
3		Préparer l'environnement de développement	1journée	05/02/2018	05/02/2018	2	TAHRAOUI Mohammed Iheb,BERRACHED Sihem
4		Apprentissage des langages de développement	7journées?	06/02/2018	14/02/2018	3	BERRACHED Sihem,TAHRAOUI Mohammed Iheb
5		Réaliser le premier chapitre du rapport	5journées?	06/02/2018	12/02/2018	3	TAHRAOUI Mohammed Iheb,BERRACHED Sihem
6		Réaliser la maquette des applications	2journées?	13/02/2018	14/02/2018	5	TAHRAOUI Mohammed Iheb,BERRACHED Sihem
7		Compléter le second chapitre de rapport	7journées?	15/02/2018	23/02/2018	6	TAHRAOUI Mohammed Iheb,BERRACHED Sihem
8		Développer les fonctionnalité communes	11journée...	19/02/2018	05/03/2018		BERRACHED Sihem,TAHRAOUI Mohammed Iheb
9		Séparer l'application en 6 parties	2journées?	06/03/2018	07/03/2018	8	TAHRAOUI Mohammed Iheb,BERRACHED Sihem
10		Développer les fonctionnalité Spécifique au Client (Plateforme Android)	19journée...	08/03/2018	03/04/2018	9	BERRACHED Sihem,TAHRAOUI Mohammed Iheb
11		Générer et adapter l'application coté client dans la Plateforme IOS	4journées?	04/04/2018	09/04/2018	10	TAHRAOUI Mohammed Iheb,BERRACHED Sihem
12		Développer les fonctionnalité Spécifique au Marchand(Plateforme Android)	20journée...	10/04/2018	07/05/2018	11	BERRACHED Sihem,TAHRAOUI Mohammed Iheb
13		Générer et adapter l'application coté Marchand dans la Plateforme IOS	2journées?	08/05/2018	09/05/2018	12	BERRACHED Sihem,TAHRAOUI Mohammed Iheb
14		Complété le rapport	8journées?	30/04/2018	09/05/2018		BERRACHED Sihem,TAHRAOUI Mohammed Iheb
15		Développer les fonctionnalité Spécifique au Administrateur(Plateforme Andro...	14journée...	10/05/2018	29/05/2018	13	BERRACHED Sihem,TAHRAOUI Mohammed Iheb
16		Valider le Rapport	7journées?	25/05/2018	04/06/2018		BERRACHED Sihem,TAHRAOUI Mohammed Iheb
17		Générer et adapter l'application coté Administrateur dans la Plateforme IOS	5journées?	30/05/2018	05/06/2018	15	BERRACHED Sihem,TAHRAOUI Mohammed Iheb
18		Tester L'application	18journée...	06/06/2018	29/06/2018	17	BERRACHED Sihem,TAHRAOUI Mohammed Iheb

II.3 la méthode agile scrum

La méthode scrum sert à gérer les projets d'une façon agile, utilisé pour développer un logiciel complet ou quelques parties d'un grand système avec une approche itérative incrémentale qui consiste à découper le projet en plusieurs étapes appelé « sprint » ou « itération ». Les sprints sont toujours d'une courte durée entre 2 à 4 semaines constituent d'un ensemble de besoins appelé « user story ».

Scrum commence par un « **Product backlog** » qui sélectionnons toutes les choses qui doivent être faites dans le projet. A partir de cet ensemble nous sélectionnons les exigences priorisés qui seront développées « **user story** ». Ce sous ensemble des éléments forment « **sprint backlog** » qui sera mis à jour sur une base quotidienne. Pendant les sprints chaque jour des réunions « **Daily scrum meeting** » seront organisés avec tous les membres de l'équipe pour discuter le déroulement du travail et les obstacles rencontrés.

Scrum se base sur une équipe de différents rôles : Product down, scrum master et team,

Product down : il s'agit d'avoir une vision de ce que nous voulons construire et transmettre cette vision à l'équipe de développement.

Scrum master : il guide l'équipe de développement pendant les itérations, il assure la communication efficace entre les scrums team et il gère aussi les obstacles pour l'équipe.

Scrum team : il s'agit d'un ensemble de personnes travaillant ensemble pour livrer un produit utilisable à la fin de chaque sprint. [13]

Nous avons choisis scrum comme méthode de gestion de notre projet pour développer notre application, car elle permet d'offrir une meilleurs visibilité par rapport aux autres méthodes, nous avons utilisé quatre itérations pour la réalisation de notre application qu'on va les définir :

Itération 1 (du 13/02/2018 au 05/03/2018)

{**BismiLlah**} est notre 1^{ère} itération qui a duré 17 jours. On a détecté un bug bloquant qu'il s'agit d'une erreur de génération d'application, 4 bugs majeurs et 8 bugs mineurs.

Quelques fonctionnalités de ce sprint **Backlog** :

- La Réalisation de la charte graphique en intégrant les différentes interfaces en commun entre le Client, Marchand et le l'Administrateur tel que la page de profile, le menu latérale, la page articles et catalogues.
- L'implémentation du coté fonctionnelle et interactif entre les différents pages communes.

Itération 2 (du 06/03/2018 au 09/04/2018)

{Percevoir} est notre 2eme itération qui a duré 33 jours, on a commencé la séparation des espaces utilisateurs car on a entamé les différentes fonctionnalités spécifiques pour chaque utilisateur. Durant ce sprint on a eu un bug majeur et 12 bugs mineurs.

Quelques fonctionnalités de ce sprint Backlog :

- Intégration de la partie qui détaille des articles, ainsi que l'implémentation du GPS afin de localiser la disponibilité du produit.
- Préparation de prestataire de paiement en utilisant **Pin Payment** comme plateforme de Test afin de valider les transactions d'achats.
- Implémentation de la partie Achat.
- Installation et intégration de la partie stockage interne native en utilisant le plugin **SQLite** afin de garantir la bonne gérance des tickets de réduction et garantir l'utilisation de l'application même en mode déconnecté.
- Génération des applications **Android** et **IOS**.
- Adaptation de l'application sur la plateforme **IOS**.

Dans cette itération nous avons fréquenté des requêtes **SQL** complexes conditionnelles comme s'est montré dans la figure 7 qui se charge de sélectionner le nombre de ticket valide, non valide et en attente ainsi que le nombre d'article restant.

```
1 • SELECT
2   article.id idArticle,
3   article.reference ReferenceArticle,
4   CASE achat.dateAchat WHEN TRUE THEN achat.dateAchat ELSE achat_algo2.dateAchat END AS dateAchat,
5   article.quantite,
6   article.quantiteSave,
7   COUNT(achat.id) + COUNT(achat_algo2.id) AS SommeTicket,
8   SUM(if(achat.isVendu='false',1,0)) + SUM(if(achat_algo2.isVendu='false',1,0)) AS SommeTicketNoVendu,
9   SUM(if(achat.isVendu='true',1,0)) + SUM(if(achat_algo2.isVendu='true',1,0)) AS Vendu
10  FROM
11  (article
12   LEFT JOIN achat ON achat.article_id = article.id)
13   LEFT JOIN achat_algo2 ON achat_algo2.article_id = article.id
14   WHERE article.marchand_id = 16
15   GROUP BY article.id
16   HAVING
17   SommeTicket >=2
```

Figure 7: Requête SQL complexe conditionnelle.

Itération 3 (du 10/04/2018 au 09/05/2018)

{Dégarnir} est la 3eme itération qui a duré presque 30 jours. Dans ce sprint on s'est focalisé sur la partie Marchand et toutes ses fonctionnalités qui lui mènent à la bonne gérance de sa boutique virtuelle. Dans cette partie on s'est mis face à un bug bloquant dont il s'agit de la taille monstrueuse de la requête d'ajout d'article lors de sa transmission au serveur, ainsi que 28 bugs mineurs.

Quelques fonctionnalités de ce sprint Backlog :

- Intégration de la fonctionnalité ajouter article dont elle est caractérisée par la spécification des images descriptives de produit les adresses et les descriptions.
- Implémentation de la partie scanner le **QR Code** et son déchiffrement.
- L'ajout de la partie Suivre interactions.
- Génération des applications **Android** et **IOS**.
- Adaptation de l'application sur la plateforme **IOS**.

Itération 4 (du 10/05/2018 au 30/05/2018)

{**Kudos**} est la 4^{ème} itération qui a duré 20 jours. La partie la plus importante dans notre projet qui consiste à tous types de gérance de Marketplace, cette partie est dédiée aux Administrateurs. Dans ce sprint on a eu un problème bloquant et un bug majeur.

Quelques fonctionnalités de ce sprint Backlog :

- Implémentation de la fonction notification.
- La gérance des utilisateurs.
- L'analyse des données.
- Génération des applications **Android** et **IOS**.
- Adaptation de l'application sur la plateforme **IOS**.

III Patron de conception « Design Patterns »

III.1 Patron de création « Patterns de Création »

Les modèles de création visent à séparer un système de la façon dont ses objets sont créés, composés, et représentés. Ils augmentent la flexibilité du système en termes de quoi, qui, comment, et quand de la création d'objet. Les modèles créatifs encapsulent la connaissance sur les classes d'un système, mais ils cachent les détails de la façon dont les instances de ces classes sont créées et assemblées. [14]

III.2 Patron de fabrication « Design Pattern factory »

Le design Pattern **factory** est un moyen de créer des objets, mais en laissant des sous-classes décider exactement quelle classe instancier. Diverses sous-classes pourraient mettre en œuvre l'interface, design Pattern factory instancie la sous-classe appropriée en fonction des informations fournis par le client ou extrait de l'état actuel. [14]

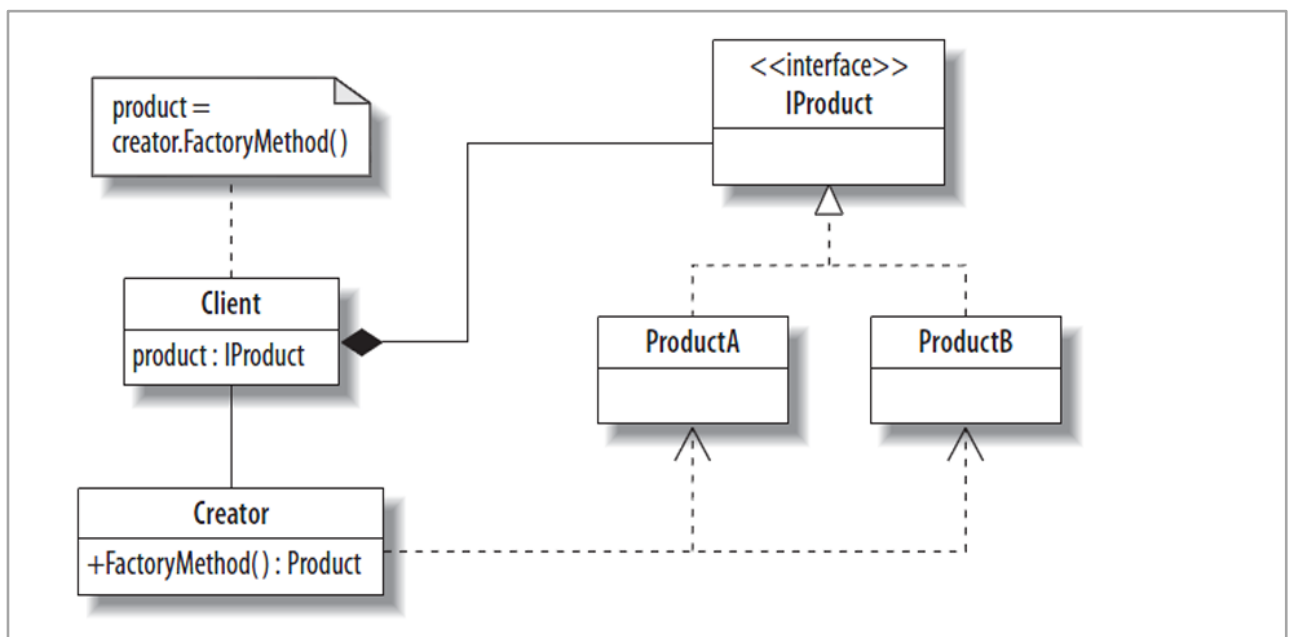


Figure 8: Patron de fabrication "factory".

Nous avons utilisé ce design pattern suite au niveau élevé de la flexibilité désiré sur nos applications mobile, car les performances matérielles de l'utilisateur final ça se varie du Smartphones le moins performant au Smartphones moderne. Donc on a visé le minimum (en t performance) pour qu'on puisse garantir le maximum des utilisateurs avec n'importe quelle configuration de leurs hardwares (matériels) et software (version d'OS).

III3 Patron de conception « Pattern de conception »

II.3.1 Modèle vue contrôleur

Modèle Vue Contrôleur qui signifie qu'une application sera divisée en au moins trois parties qui réagissent entre eux avec une manière homogène.

Pourquoi MVC ?

M : Modèles qui contiennent ou représentent les données avec lesquelles les utilisateurs travaillent, il existe deux type de modèles : modèle de vue simples et modèles de domaine.

Modèle de vue simple : représentent juste les données étant transférées entre les vues et les contrôleurs.

Modèle de domaine : contiennent les données dans une entreprise ainsi que les opérations, les transformations et les règles de manipulation de ces données.

V : Vues sont utilisées pour rendre une partie du modèle en tant qu'interface utilisateur.

C : Contrôleurs traitent les demandes entrantes, effectuent des opérations sur le modèle et sélectionnent les vues à rendre à l'utilisateur. [15]

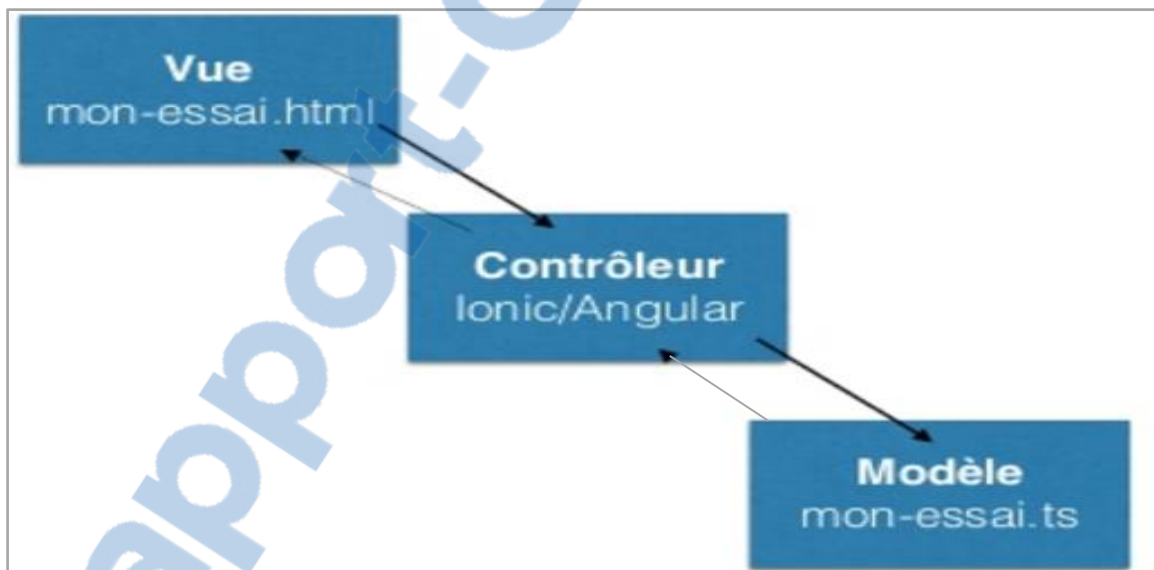


Figure 9: Patron MVC. [16]

La figure 09 représente les **trois tiers** fondamentaux de l'architecture MVC du Framework IONIC.

La figure 10 représente l'architecture notre application sous forme de modèle MVC :

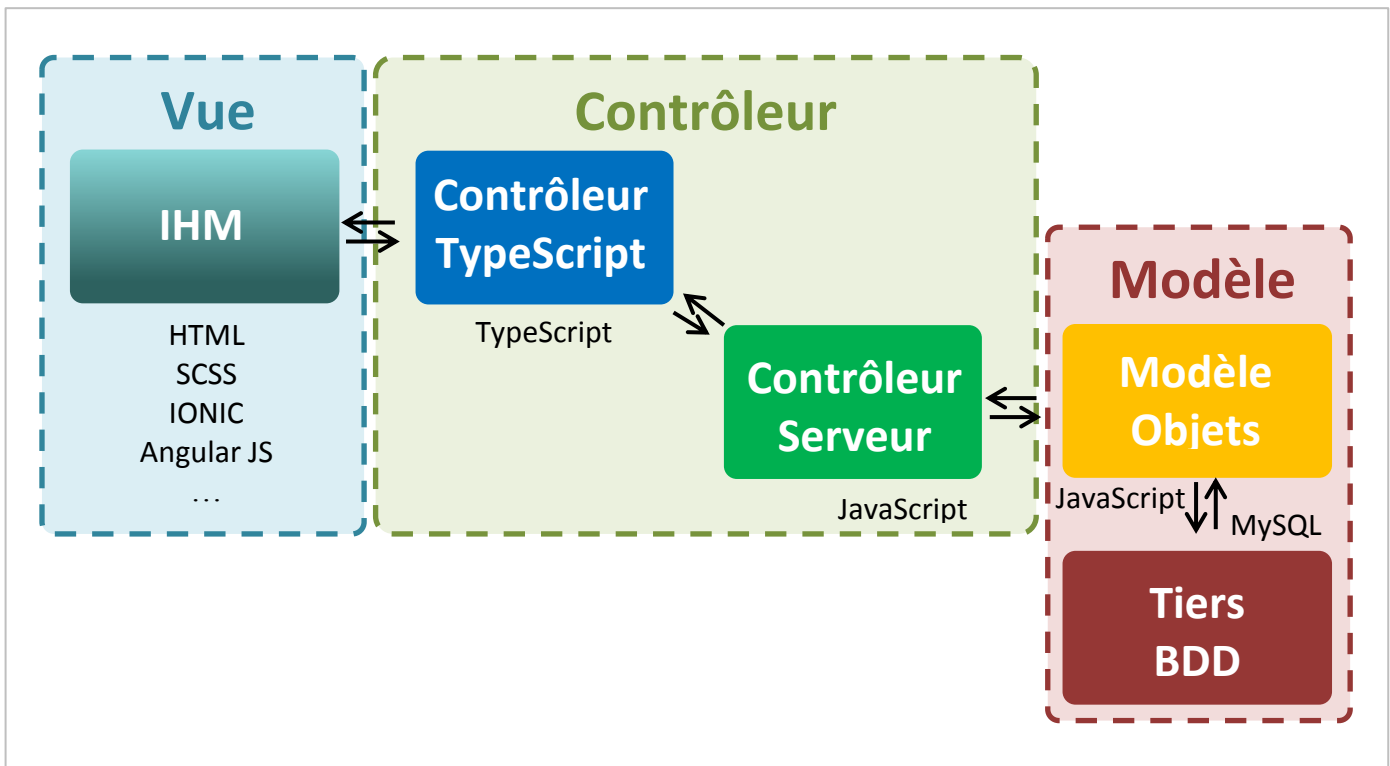


Figure 10: Architecture trois tiers MVC.

IV Analyse

IV.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation décrit les utilisateurs d'un système, et ce que ces derniers sont supposés faire. Chaque usage que les acteurs font est représenté par un cas d'utilisation. Dans notre application on distingue principalement trois acteurs : l'acheteur, le marchand et l'administrateur :

IV.1.1 Cas d'utilisation acheteur

L'acheteur est un client indirect (client de notre client direct). La figure 11 représente les cas d'utilisation d'acheteur :

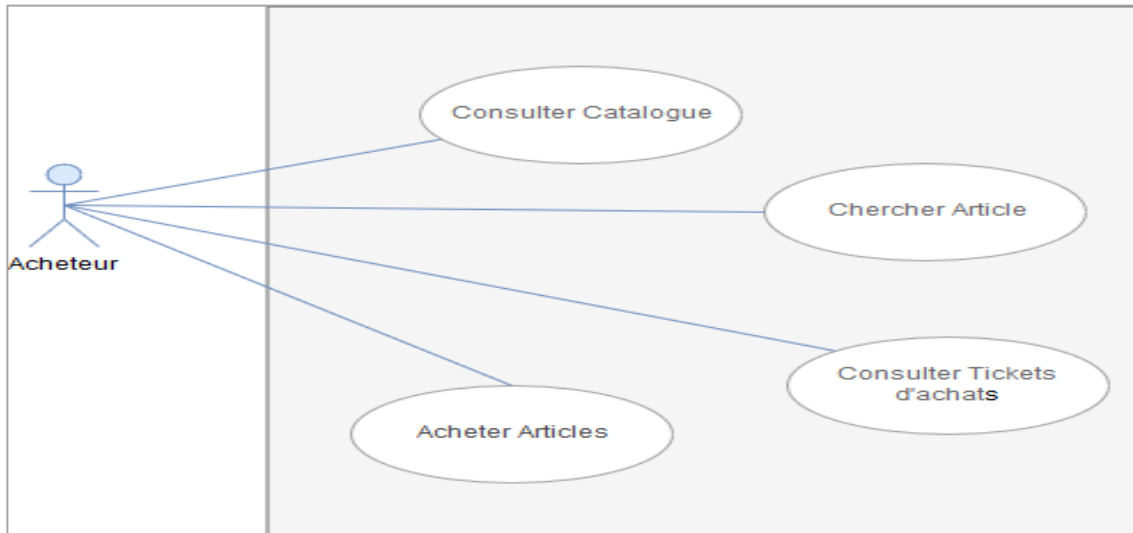


Figure 11: Diagramme de cas d'utilisation acheteur.

Le tableau 4 présente les deux scénarios (nominal et alternatif) de chaque cas d'utilisation acheteur, ainsi que les pré-conditions et les poste-conditions :

Tableau 4: Description de cas d'utilisation acheteur.

Acteur	Acheteur
Pré-condition	Il doit avoir une connexion internet. Il doit être authentifié.
Scénario nominal	<p>Consulter catalogue :</p> <ol style="list-style-type: none"> 1- l'acheteur demande de consulter catalogue. 2- le système affiche catalogue. <p>Chercher article :</p> <ol style="list-style-type: none"> 1 : l'acheteur saisit un article à chercher. 2 : le système affiche l'article désiré. 3 : l'acheteur demande détail. 4 : le système affiche détail de l'article. <p>Acheter article :</p> <ol style="list-style-type: none"> 1 : l'acheteur clique sur acheter. 2 : le système génère une réduction de prix. 3 : l'acheteur remplit les données de paiement. 4 : le système vérifie les informations. 5 : le système génère un ticket de réduction. <p>Consulter tickets achats:</p> <ol style="list-style-type: none"> 1 : l'acheteur clique sur consulter les tickets d'achats. 2 : le système affiche la liste des achats de cet acheteur.
Scénario alternatif	<p>Chercher article :</p> <p>A.2 : Le système affiche article non disponible.</p> <p>Acheter article :</p> <p>A.5 : le système répond : données invalides.</p>
Post condition	Un produit s'est vendu. La quantité de ce produit se diminue.

IV.1.2. Cas d'utilisation marchand

Le marchand est un fournisseur (client direct) qui a un compte sur notre application. , La figure 12 représente les cas d'utilisation de marchand:

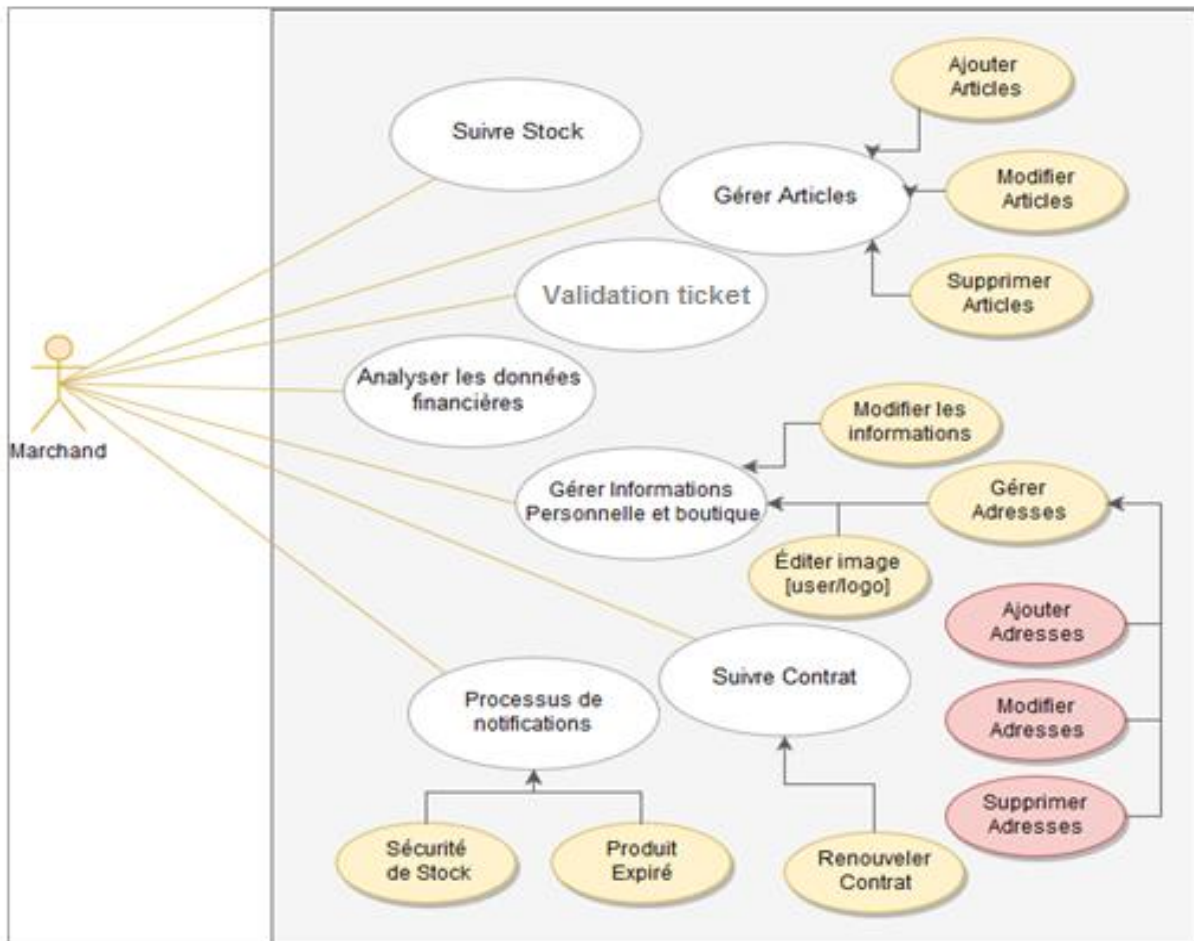


Figure 12: Diagramme de cas d'utilisation marchand.

Le tableau 5 présente les deux scénarios (nominal et alternatif) de chaque cas d'utilisation marchand, ainsi que les pré-conditions et les poste-conditions :

Tableau 5: description cas d'utilisation marchand.

Acteur	Marchand
Pré-condition	Il doit avoir une connexion internet. Il doit être authentifié. Il doit avoir un contrat valide.
Scénario nominal	<p>Gérer stock :</p> <ol style="list-style-type: none"> 1 : le marchand demande de consulter le stock. 2 : le système affiche le stock des produits. 3 le marchand peut ajouter la quantité de produit précis. <p>Ajouter article :</p> <ol style="list-style-type: none"> 1 : le marchand demande l'ajout d'un nouvel article. 2 : le système affiche un formulaire à remplir. 3 : le marchand remplit le formulaire. 4 : le système envoie le formulaire à l'administrateur.

	<p>5 : l'administrateur accepte l'article. 6 : le système envoie une notification d'acceptation au marchand et publie l'article.</p> <p>Supprimer article : 1 : le marchand demande une suppression d'article. 2 : le système affiche un message de confirmation. 3 : le marchand confirme la suppression. 4 : le système supprime l'article.</p> <p>Validation ticket : 1 : le marchand demande l'ouverture de scanner QR Code. 2 : le système ouvre l'appareil photo de Smartphone. 3 : le marchand scanne le QR Code. 4 : le système affiche les informations de scanne.</p> <p>Modifier les informations: 1 : le marchand demande de modifier les informations. 2 : le système autorise le marchand à modifier. 3 : le marchand modifie l'information voulu.</p> <p>Supprimer adresse: 1 : le marchand demande de supprimer une adresse. 2 : le système vérifié si sa reste aucune récupération d'article de cet Adresse. 3 : le système affiche une alerte de confirmation. 4 : le marchand confirme la suppression.</p> <p>Suivre contrat (renouveler contrat) : 1 : le marchand demande de renouveler contrat. 2 : le système affiche un message de confirmation. 3 : le marchand confirme le renouvellement. 4 : le système envoie la demande à l'administrateur. 5 : l'administrateur accepte le renouvellement. 6 : le système renouvèle le contrat.</p>
<p>Scenari alternatif</p>	<p>Supprimer article : A.3 : le marchand annule la suppression.</p> <p>Ajouter article : A.5 : l'administrateur refuse l'article et justifie la refusions. A.6 : le système envoie la justification au marchand.</p> <p>Supprimer adresse: A.3 : le système affiche un message de refusions en cas des articles non récupérés de cette adresse.</p> <p>Suivre contrat (renouveler contrat) : A.3 : le marchand annule le renouvellement. A.5 : l'administrateur refuse le renouvellement.</p>
<p>Post condition</p>	<p>La quantité d'article augmente (ajout). La quantité d'article diminue (suppression). Le nombre de jour de contrat augmente dans le cas de renouvellement. Les informations de marchand peut être seront modifiées. Une adresse peut être supprimée.</p>

IV.1.3 Cas d'utilisation administrateur

L'administrateur est un responsable général de l'application, la figure 13 représente les cas d'utilisation d'administrateur:

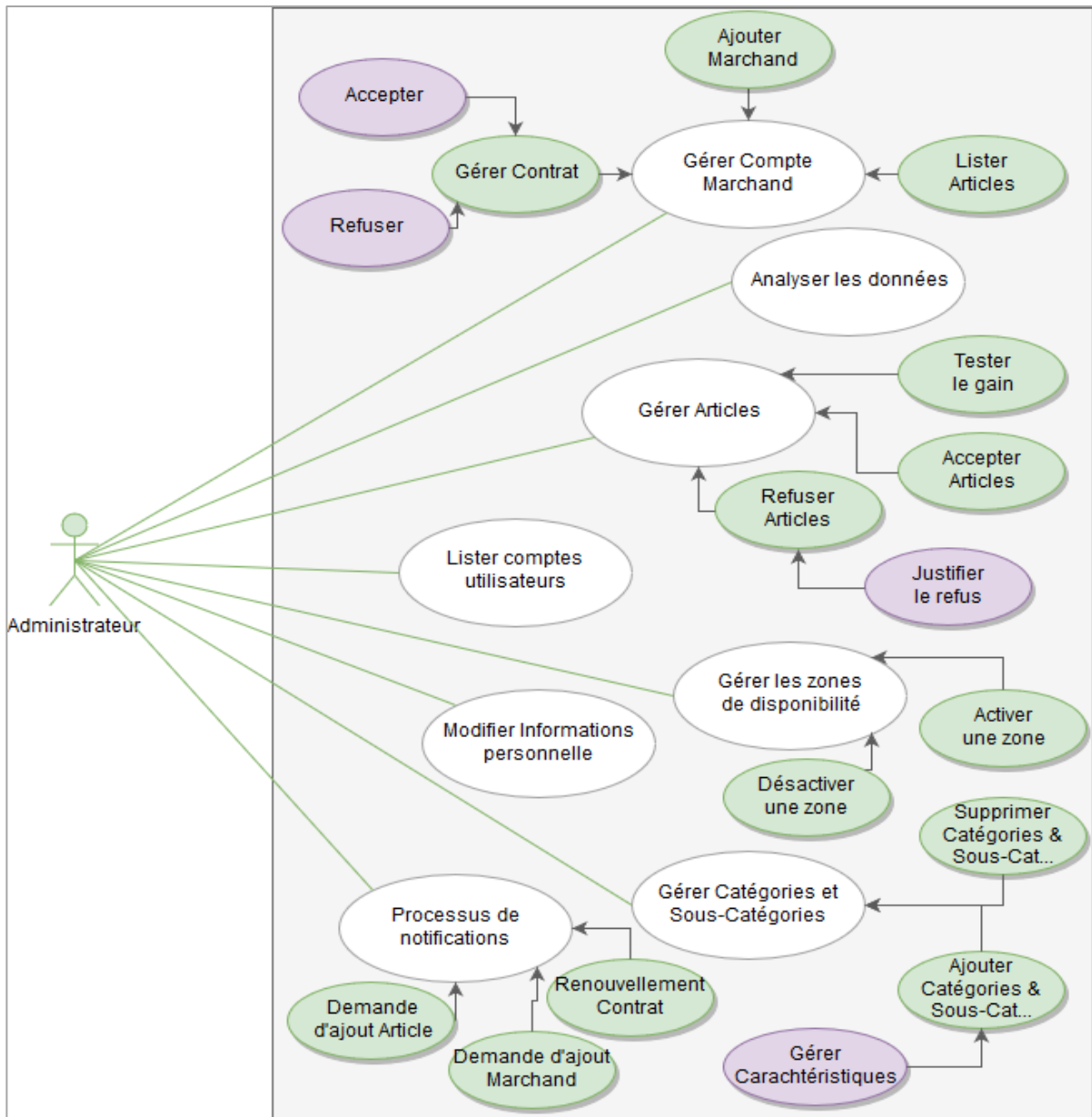


Figure 13: Diagramme cas d'utilisation administrateur.

Le tableau 6 présente les deux scénarios (nominal et alternatif) de chaque cas d'utilisation administrateur, ainsi que les pré-conditions et les poste-conditions :

Tableau 6: Description de cas d'utilisation administrateur.

Utilisateur(s)	Administrateur
Pré-condition	Il doit avoir une connexion internet. Il doit être authentifié.
Scenario nominal	<p>Ajouter compte :</p> <ol style="list-style-type: none"> 1 : l'administrateur demande ajout d'un nouveau compte. 2 : le système affiche un formulaire à remplir. 3 : l'administrateur remplit le formulaire. 4 : le système affiche un message : ajout avec succès. <p>Gérer contrat :</p> <ol style="list-style-type: none"> 1 : l'administrateur reçoit une demande de contrat. 2 : le système affiche les informations. 3 : l'administrateur accepte le contrat. <p>Analyser données:</p> <ol style="list-style-type: none"> 1 : l'administrateur demande d'analyser les données. 2 : le système affiche des graphes d'analyse de données. <p>Parcourir les comptes utilisateurs:</p> <ol style="list-style-type: none"> 1 : l'administrateur demande de parcourir les comptes Utilisateurs. 2 : le système affiche la liste des comptes. <p>Ajouter catégorie et sous-catégorie:</p> <ol style="list-style-type: none"> 1 : l'administrateur demande d'ajouter une catégorie. 2 : le système demande l'administrateur de donner les sous-catégories de cette catégorie avec les caractéristiques. 3 : l'administrateur ajoute les sous-catégories et les caractéristiques de chaque une. <p>Refuser article:</p> <ol style="list-style-type: none"> 1 : l'administrateur refuse un article. 2 : le système demande l'administrateur de justifier le refus. 3 : l'administrateur remplit une justification. 4 : le système envoie la justification de refus au marchand.
Scenario alternatif	<p>Gérer contrat :</p> <p>A.3 : l'administrateur refuse le contrat.</p>
Post condition	<p>Un nouveau compte s'ajoute.</p> <p>Un nouveau contrat s'est ajouté.</p> <p>Une nouvelle catégorie s'est ajoutée avec ses sous-catégories.</p>

IV.2 Diagramme de séquence système

Diagramme de système est un sous type de diagramme de séquence qui décrit les interactions entre les acteurs et le système avec un ordre séquentiel, on présente par la suite quelques diagrammes de notre application :

IV.2.1. Diagramme de séquence système « login »

La figure 14 décrit les interactions possibles entre l'internaute et le système lors d'une authentification d'utilisateur

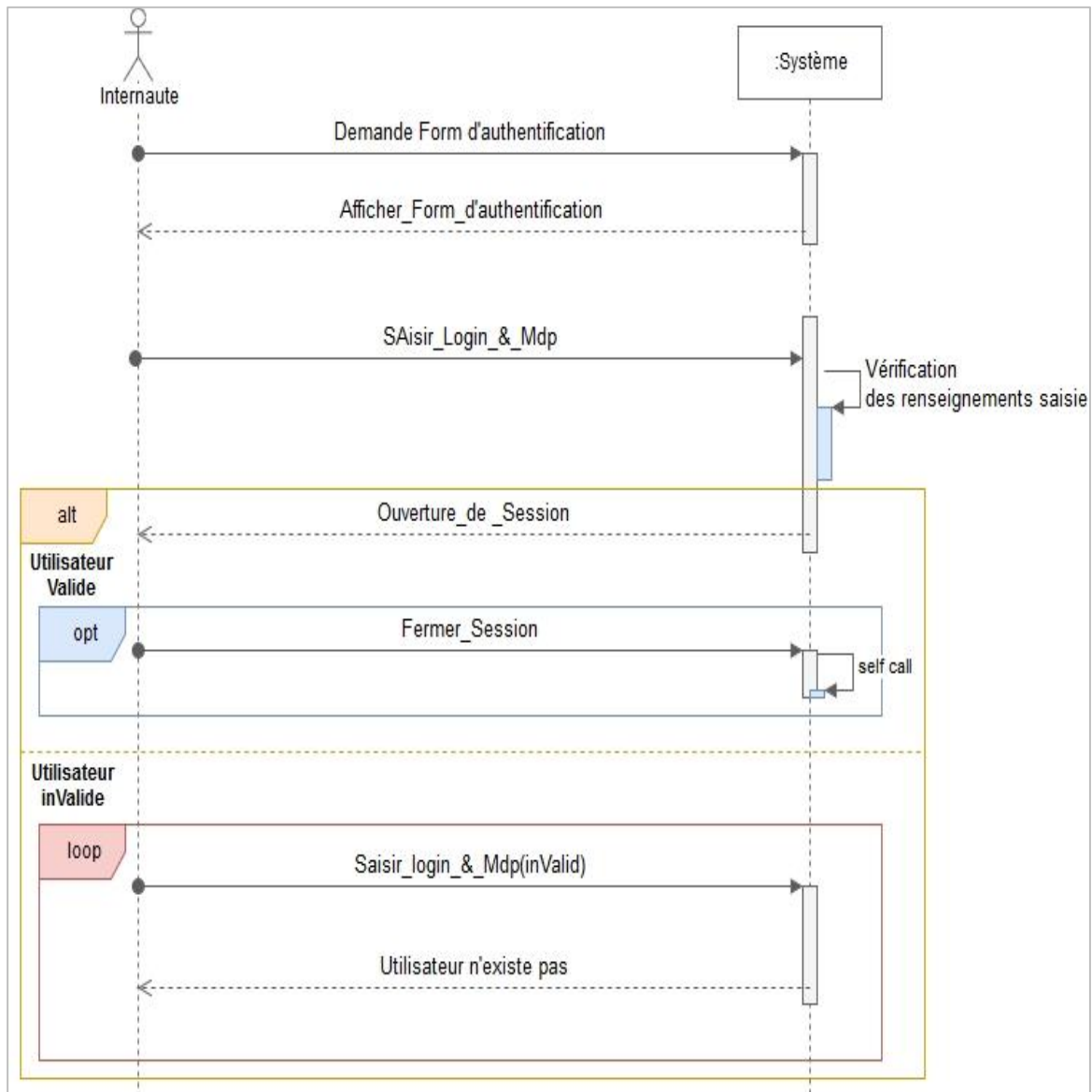


Figure 14: Diagramme de séquence système login.

Acteur : Administrateur, marchand, acheteur.

Résumé : il est indispensable de renseigner les informations d'authentification (login et mot de passe), le système vérifie ces données lorsqu'elles sont valides il lui permet d'accéder aux quelques services et pages protégées, autoriser selon le type d'utilisateur (administrateur, marchand, acheteur).

IV.2.2 Diagramme de séquence système « inscription »

La figure 15 décrit les interactions possibles entre l'utilisateur (le visiteur) et le système lors d'une inscription :

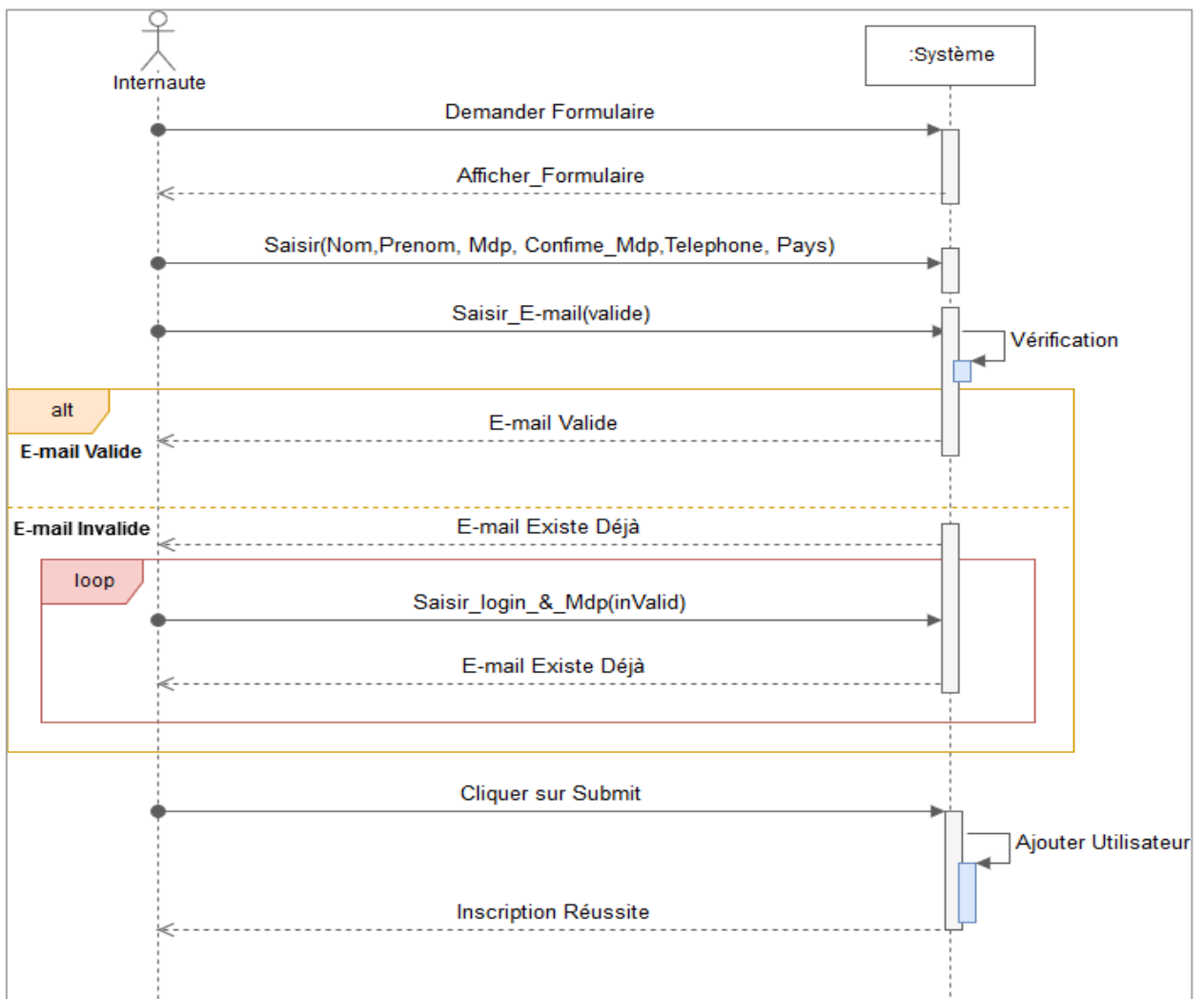


Figure 15: Diagramme de séquence système inscription.

Acteur : Acheteur.

Résumé : pour profiter de tous les privilèges dédiés à un acheteur, il doit d'abord entamer la phase d'inscription en renseignant ces informations. Le système vérifie ces données si c'étaient valides il aura un compte, sinon l'inscription se termine par échec.

IV.2.3 Diagramme de séquence système « Ajouté article »

La figure 16 décrit les interactions entre le marchand et le système pour ajouter un article :

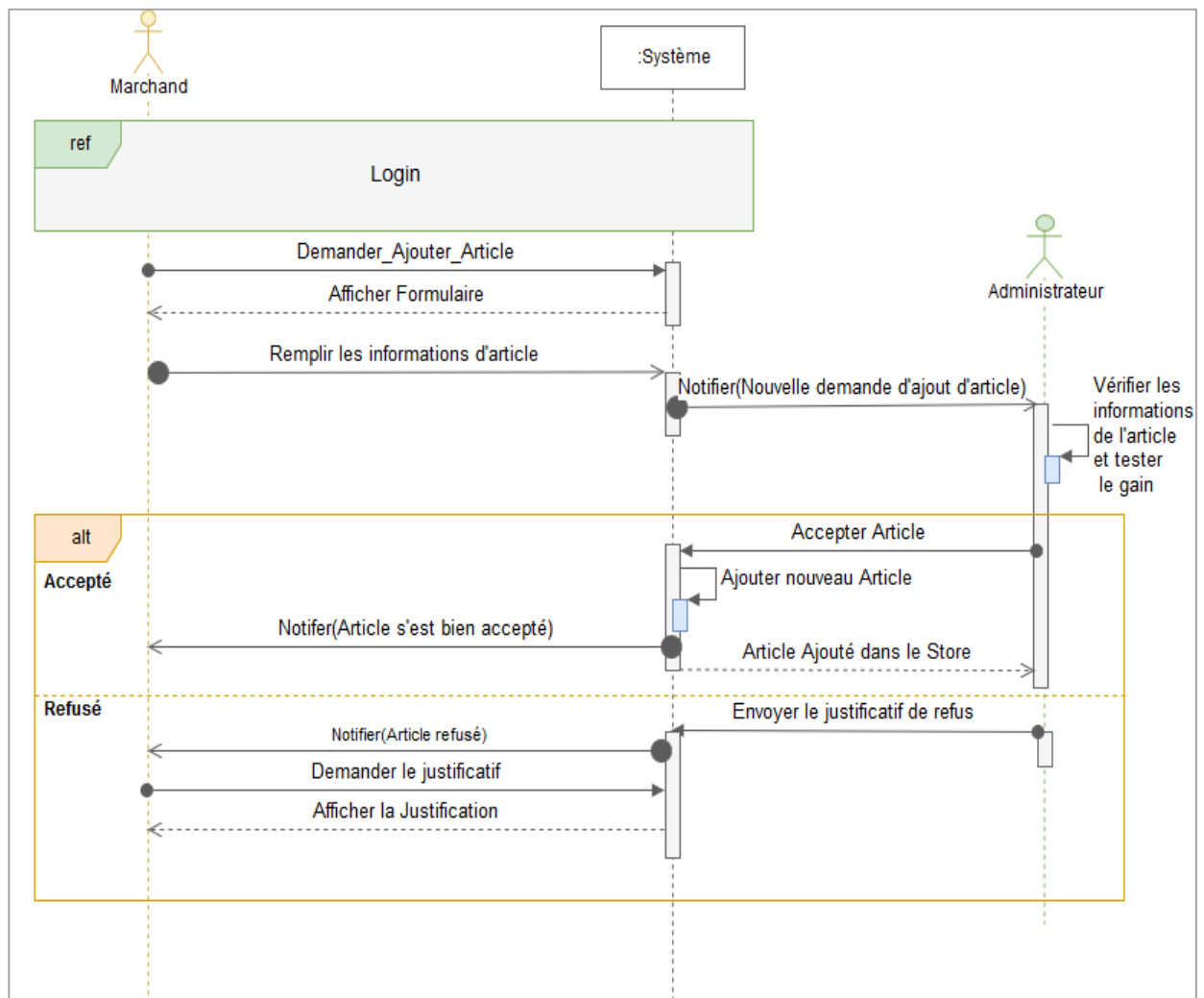


Figure 16: Diagramme de séquence système ajouté article.

Acteur : marchand.

Résumé : parmi les scénarios les plus importants dans notre application c'est l'ajout d'un nouvel article. Le marchand remplit les informations de l'article dans un formulaire, le système notifie l'administrateur qu'il y a un nouveau produit à valider, ce dernier vérifie les informations et teste le gain, s'il répond aux critères il l'accepte sinon il refuse et envoi une justification au marchand en passant par le système.

IV.2.4 Diagramme de séquence système « Gérer adresse»

La figure 17 décrit les interactions possibles entre l'utilisateur et le système lors d'une gérance des adresses:

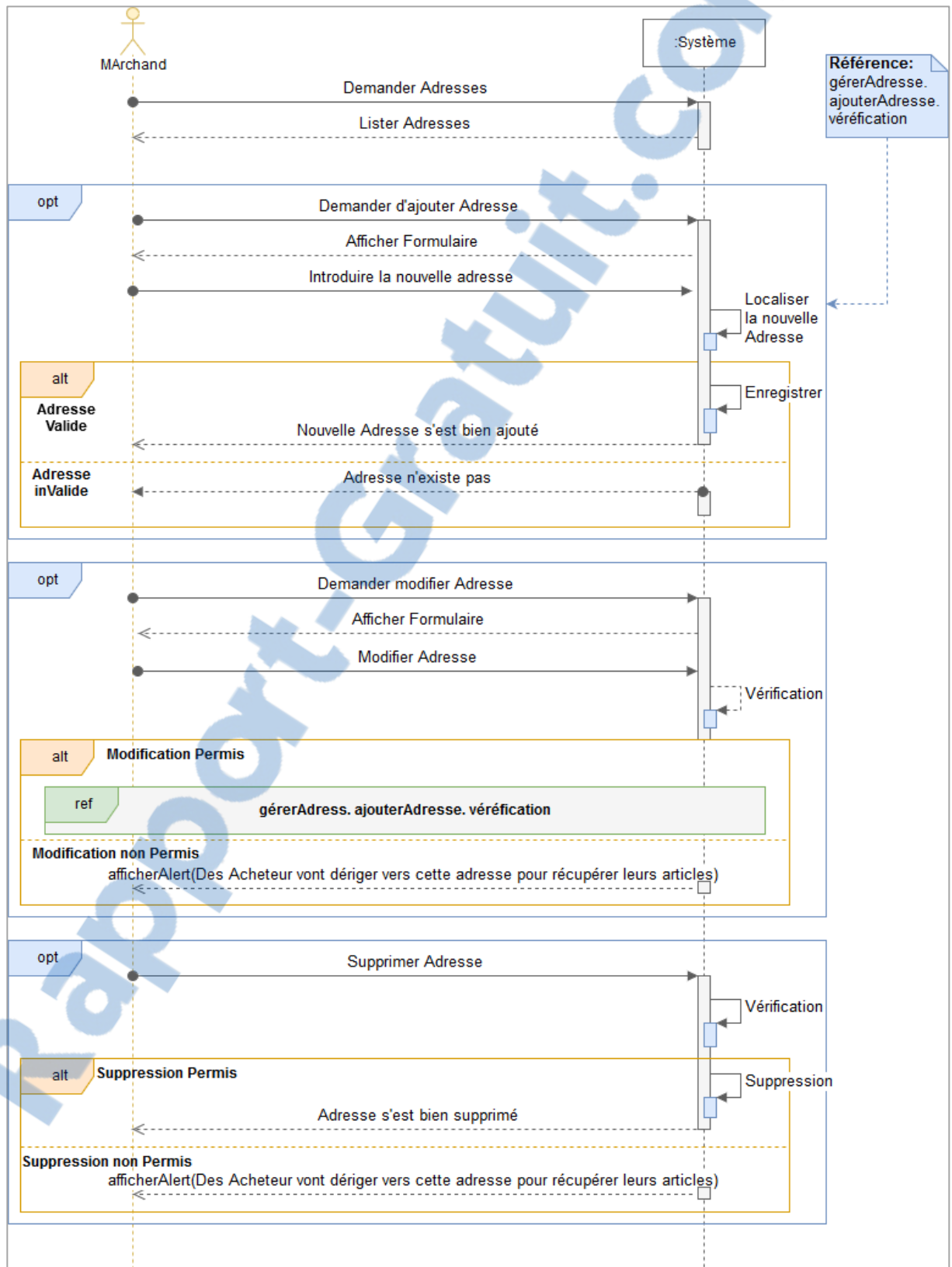


Figure 17: Diagramme de séquence système géré adresse.

Acteur : marchand.

Résumé : la gérance des adresses est l'un des scénarios les plus nécessaires, le marchand a la possibilité d'ajouter une adresse en remplissant un formulaire, le système localise cette nouvelle adresse si elle est valide il l'accepte et l'enregistre sinon il refuse.

Le marchand peut aussi modifier l'adresse de sa boutique. Il remplit le formulaire de modification, le système vérifie s'il y a des articles achetés de cette adresse et non pas encore récupérés. S'il y a alors il refuse la modification, sinon le système vérifie la localisation si c'est valide il accepte cette modification.

Le marchand a aussi la possibilité de supprimer une adresse. Le système vérifie s'il y a des achats encore non récupérés alors la suppression s'est refusée sinon s'est acceptée.

IV.2.5. Diagramme de séquence système « Consulter achats »

La figure 18 décrit les interactions possibles entre le client et le système lors d'une consultation des achats:

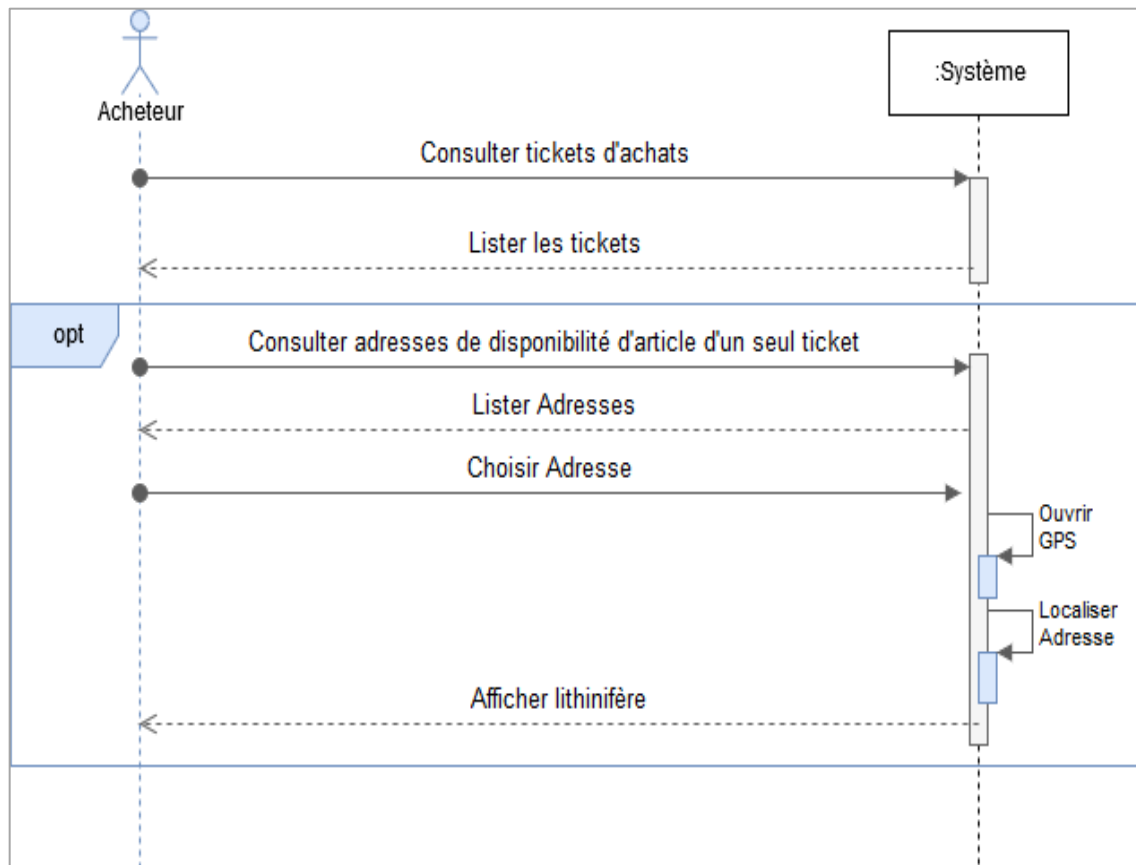


Figure 18: Diagramme de séquence système consulter achat.

Acteur : acheteur.

Résumé : l'acheteur peut consulter les tickets et les informations des articles achetés. Parmi ces informations c'est les adresses des magasins des produits achetés. Il choisit une, le système ouvre le GPS pour localiser cette dernière puis il affiche lithinifère à l'acheteur.

V. Conception

V.1. Diagramme de séquence

Diagramme de séquence est une représentation graphique qui décrit les interactions entre les acteurs et les objets de système avec un ordre séquentiel, on présente quelques diagrammes de notre application :

V.1.1. Diagramme de séquence « Ajouter article »

La figure 19 décrit les interactions entre le marchand et les différents objets de système lors d'un ajout d'article:

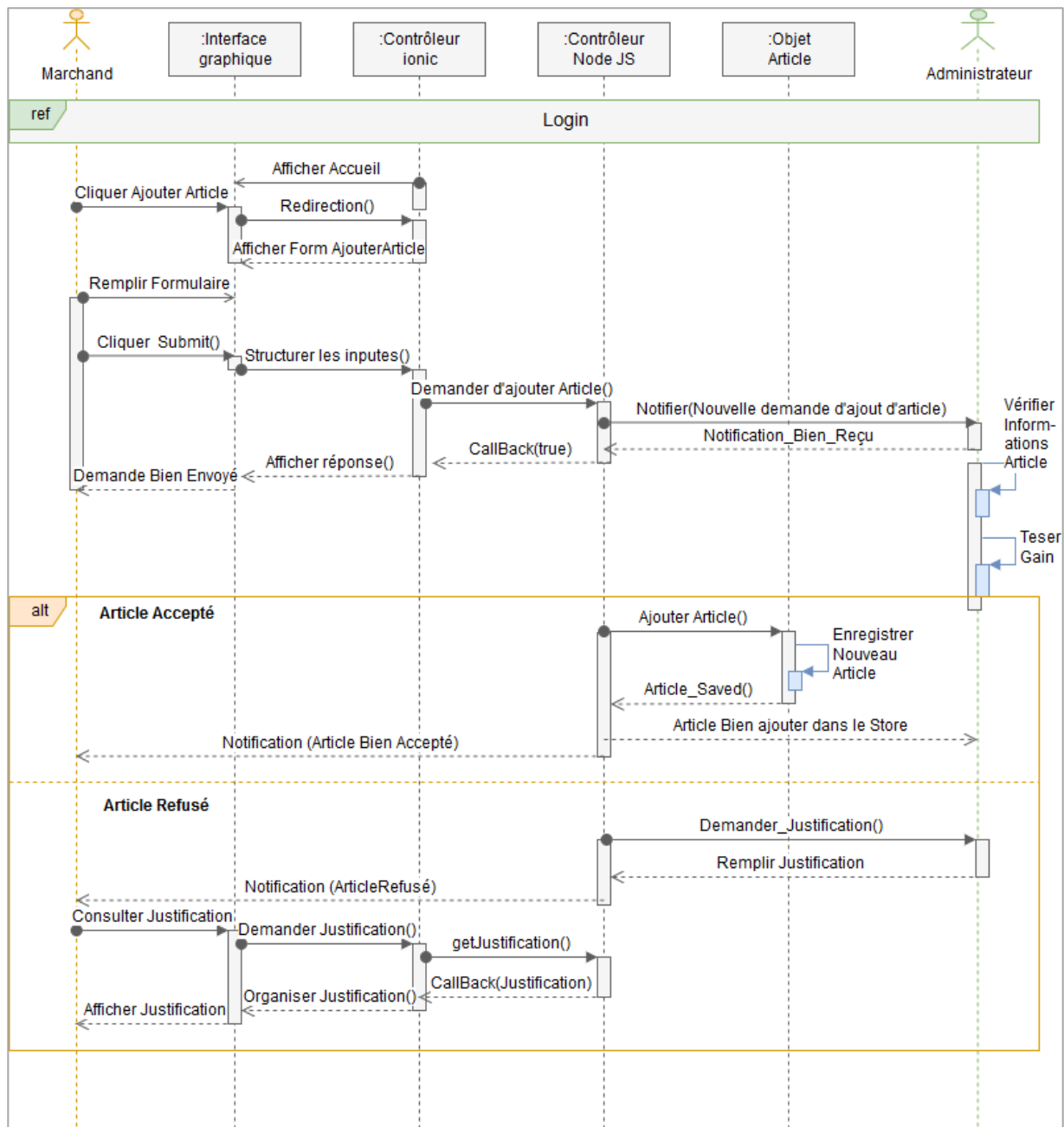


Figure 19: Diagramme de séquence ajouté article.

Acteur : marchand.

Résumé : la mise à jour des articles est un des scénarios les plus nécessaires pour le marchand. Pour cette procédure, il doit d'abord s'authentifier afin de pouvoir remplir le formulaire d'ajout avec les informations de l'article puis l'envoyer au contrôleur Node JS pour notifier l'administrateur d'avoir reçu une demande d'ajout de produit.

L'administrateur vérifie les données de formulaire puis teste le gain de cet article. Si tout est valide il accepte le produit et sera ajouté et enregistré dans l'objet article, **Node Js** envoie une notification au marchand que son produit est bien publié. Sinon l'administrateur refuse l'ajout et doit justifier cette refusion en remplissant un formulaire qui sera envoyé au marchand.



V.1.2. Diagramme de séquence « acheté article »

La figure 20 décrit les interactions possibles entre l'acheteur et les objets de système lors d'un achat d'article:

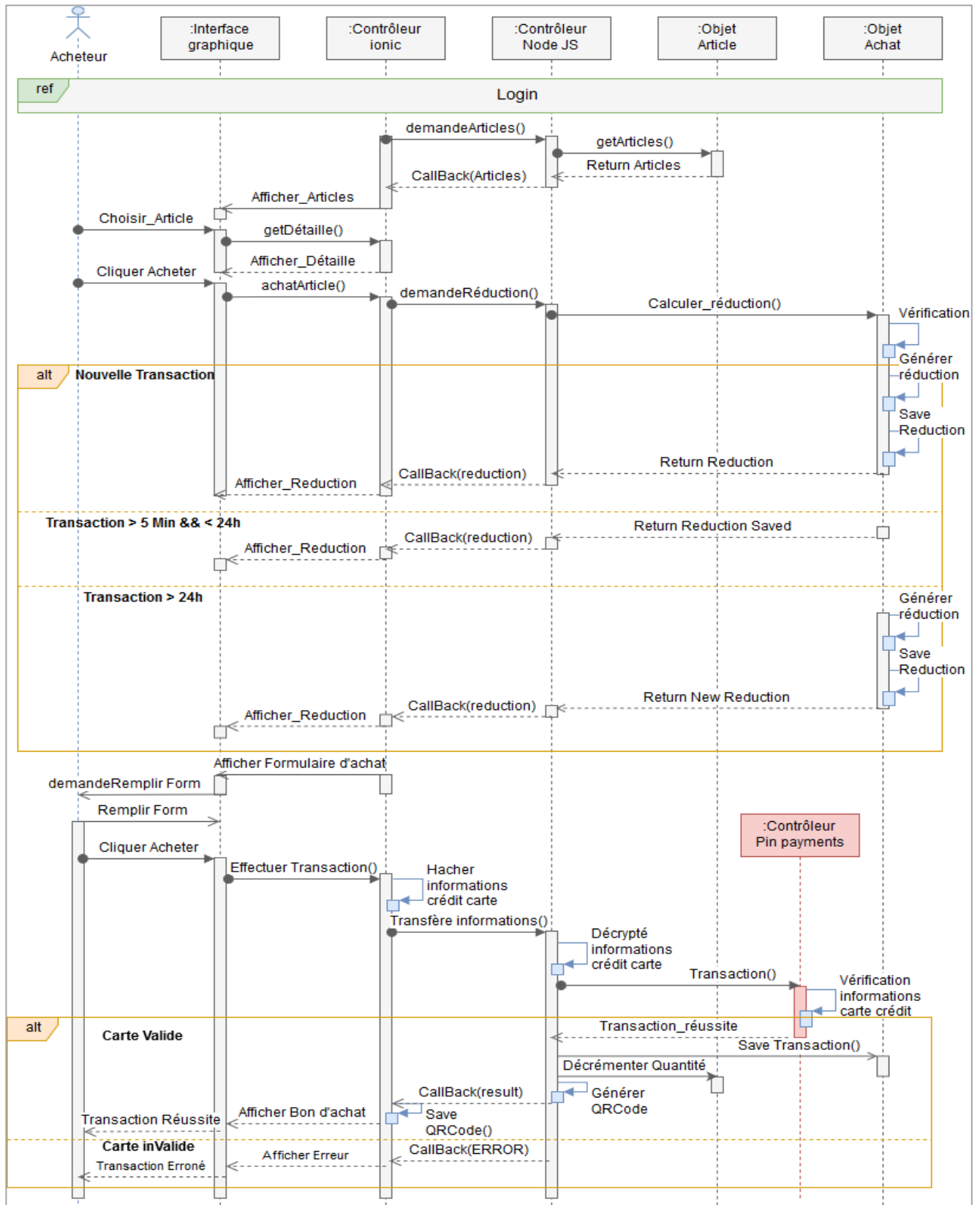


Figure 20: Diagramme de séquence acheté article.

Acteur : acheteur.

Résumé : le noyau de notre application est l'achat d'un article, l'acheteur se connecte à son compte pour consulter les produits, puis il sélectionne un pour l'acheter, le contrôleur **Ionic** demande le contrôleur Node JS de calculer un pourcentage de réduction, ce dernier la transmette à l'objet achat pour le générer.

L'objet achat vérifie si cette transaction est faite pour la première fois ou ce client a déjà fait une avant pour le même article et il a annulé la procédure après le calcul et l'affichage de pourcentage. Si c'était une nouvelle transaction, il génère une réduction et l'enregistre puis l'envoyer afin d'être afficher a l'acheteur. Sinon si l'acheteur a lancé la procédure d'achat puis il a annulé après la génération de pourcentage, ensuite avant les 24h qui suivent il a refait la transaction sur le même article, alors l'objet achat affiche toujours l'ancien pourcentage généré la première fois. Sinon si c'était après les 24h alors il génère un nouveau pourcentage de réduction.

Après l'affichage de pourcentage, l'acheteur remplit le formulaire d'achat avec les informations nécessaires, parmi ces derniers c'est le numéro de la carte crédit, l'objet **Ionic** le crypté puisque c'est une donnée trop critique donc il fait la protégé puis la transfère à l'objet contrôleur Node JS qui va la décryptée et l'envoyer au Controller pin payment afin de vérifier si cette carte est valide ou pas. Si c'était valide la procédure d'achat se termine avec succès et la quantité de cet article décrémente par un. Sinon se termine par un échec.

V.2. Modèle physique de donnée

Le modèle physique de données représente les classes intervenant dans le système avec ces attributs et les relations entre eux, la figure 21 représente un modèle de données de notre application :

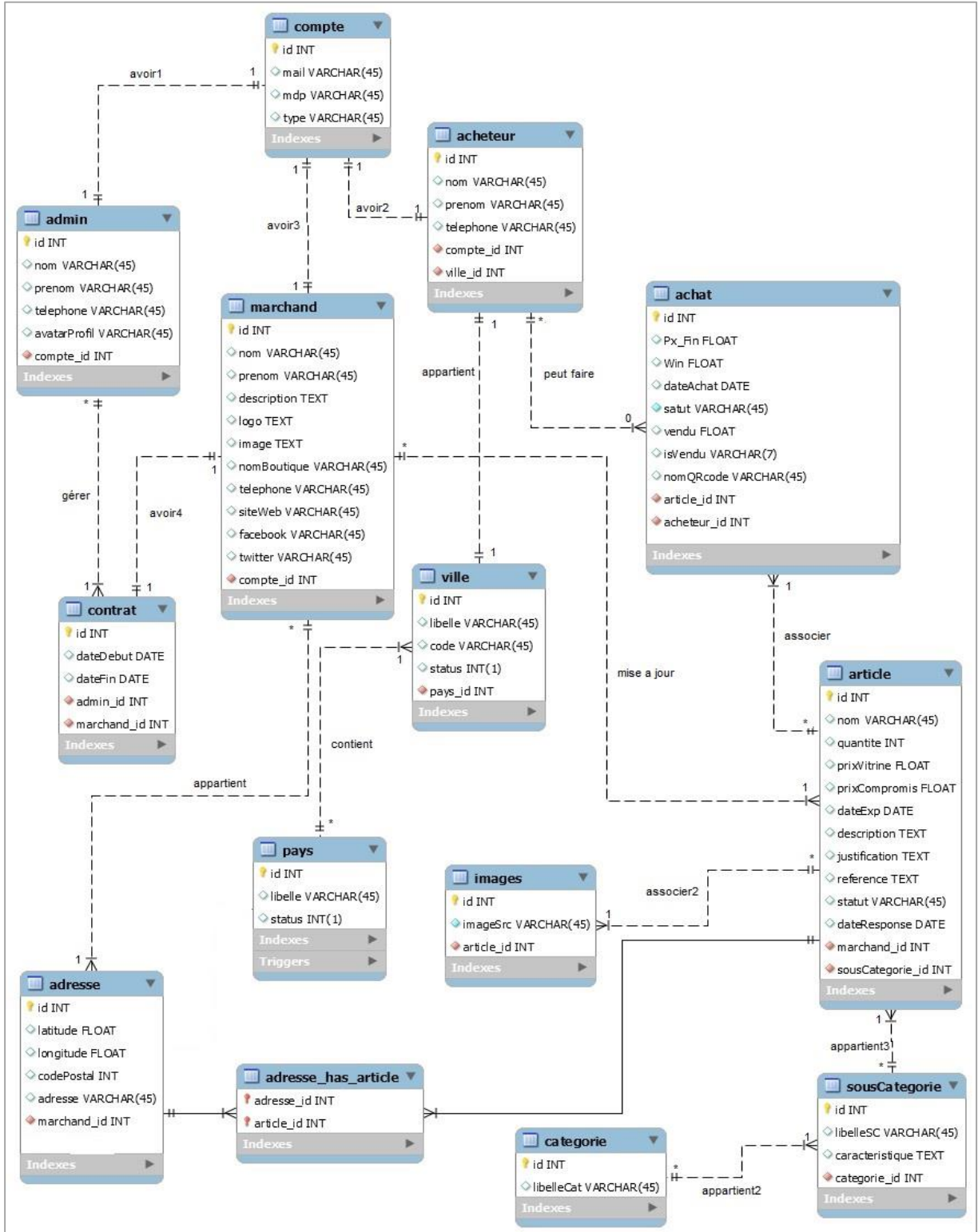


Figure 21: Modèle physique de données.

Le diagramme ci-dessus représente les données de notre projet, la table « article » est très importante puisque elle est une parmi les noyaux de notre application, elle a une relation avec « adresse » qui contient les coordonnées GPS, c'est une table nécessaire pour une bonne redirection vers les magasins des produits, « article » a aussi une relation avec la table « image » ou les photos des produits sont stockés.

Nous avons distingué trois utilisateurs : la table «Amin», gère les publications et toutes données du marketplace et les « contrats »

Pour le « marchand » chacun a un contrat, et peut avoir une ou plusieurs « adresses » de son magasin.

Et pour la table « acheteur » est une relation avec « achat » qui correspond à un seul « article».

V. Conclusion

Ce chapitre a donné une vision sur notre application « **DiscountDice** », on a commencé par la gestion de projet qui définit la planification qui est une étape primordiale dans un projet pour un bon déroulement et enchaînement d'activité, puis on a passé aux différents diagrammes UML:

- **Cas d'utilisation** : a décrit les acteurs de notre système ainsi que leurs rôles.
- **Diagramme de séquence système** : a décrit les interactions entre les acteurs et le système comme boîte noire.
- **Diagramme de séquence** : a présenté les interactions entre les acteurs et les différents objets de notre système.
- **Modèle conceptuel de données** : a présenté les classes intervenantes dans notre application.

Chapitre IV

Réalisation de l'application

I. Introduction

Dans ce chapitre nous abordons la dernière partie qui représente la réalisation de notre application, en se basant sur les mécanismes et les solutions déterminés dans la phase de conception.

II. Environnement de logiciel

Le tableau 7 présente les configurations des téléphones et des serveurs où notre application peut être utilisée :

Tableau 7: Configuration de téléphone et serveur.

Recommandations Smartphones		
OS	Android	IOS
Version Minimale	4.5.0	8.0
RAM Minimum	512	512
Stockage Libre	40Mb	40Mb
Connexion internet	Recommandé	Recommandé
GPS	Recommandé	Recommandé
Cordova-Plugins	Whitelist, device, splashscreen, ionic-webview, ionic-keyboard, camera, networkinterface, statusbar, sqlite-storage, local-notification, cszbar.	
Recommandations Serveur		
OS	Windows, Linux, MacOS	
RAM Minimum	4GB	
Stockage Minimum	30GB	
Processeur	I5	
Connexion Internet	Nécessaire	
Google Map API Key	Recommandé	
NPM dépendances	body-parse, express, nodemailer, sha1, qr-image, randomstring, fs, path, multer, base64-img, pinjs, moment, mysql, node-geocoder, email-existence, cors	

III. Environnement de développements

Lors du développement de cette application, nous avons utilisé les plateformes, les logiciels et les technologies suivantes:

III.1 Plateformes

Google Map : un service web qui fournit des informations détaillées sur les régions géographiques.



Pin Payments : Est une plate-forme pour accepter les paiements en ligne par carte crédit. [17]



Draw.io: Est une application création de diagrammes compatible avec Google Drive, permet de dessiner : des organigrammes, des diagrammes UML et des maquettages d'interface. [18]



Trello: Est outil de gestion de projet en ligne, basé sur une organisation d'une liste des cartes, ces derniers sont assignables à des utilisateurs et sont mobile d'une planche à une autre, Dans notre projet on a trois listes :



- Les tâches à faire (TO DO).
- Les tâches en cours de réalisation (DOING).
- Les tâches réalisées (DONE). [19]

Drive: Est un service de stockage et de partage de fichiers (vidéos, photos, documents, etc.) dans le Cloud.



FireBase : Est un ensemble de service de haut niveau pour développer des applications mobiles et web, offre des services pour la gestion des utilisateurs, des notifications, Stockage des fichiers et de base de données. [20]



Apple Developer : Une plateforme qui offre la possibilité de développé des applications iPhone, iPad, Mac, Apple Watch, Apple TV et Message et de les partagé sur **App Store** après avoir eu un certificat de développement. Un certificat qui nous a couté **100 \$** et une paperasse spécifique.



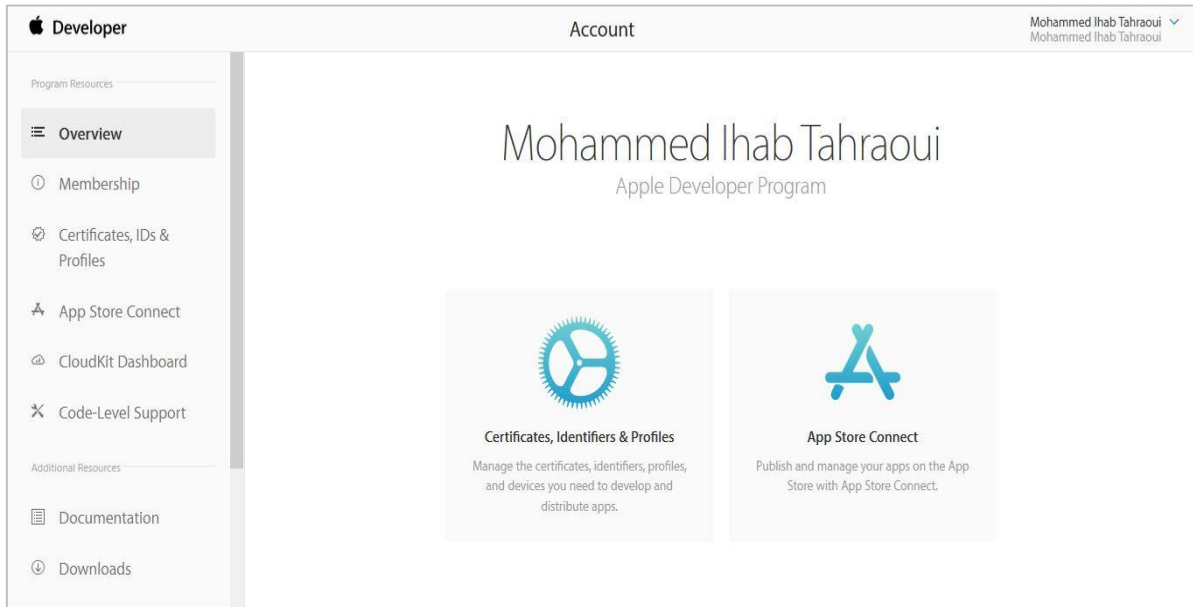


Figure 23: Plateforme Apple Developer.

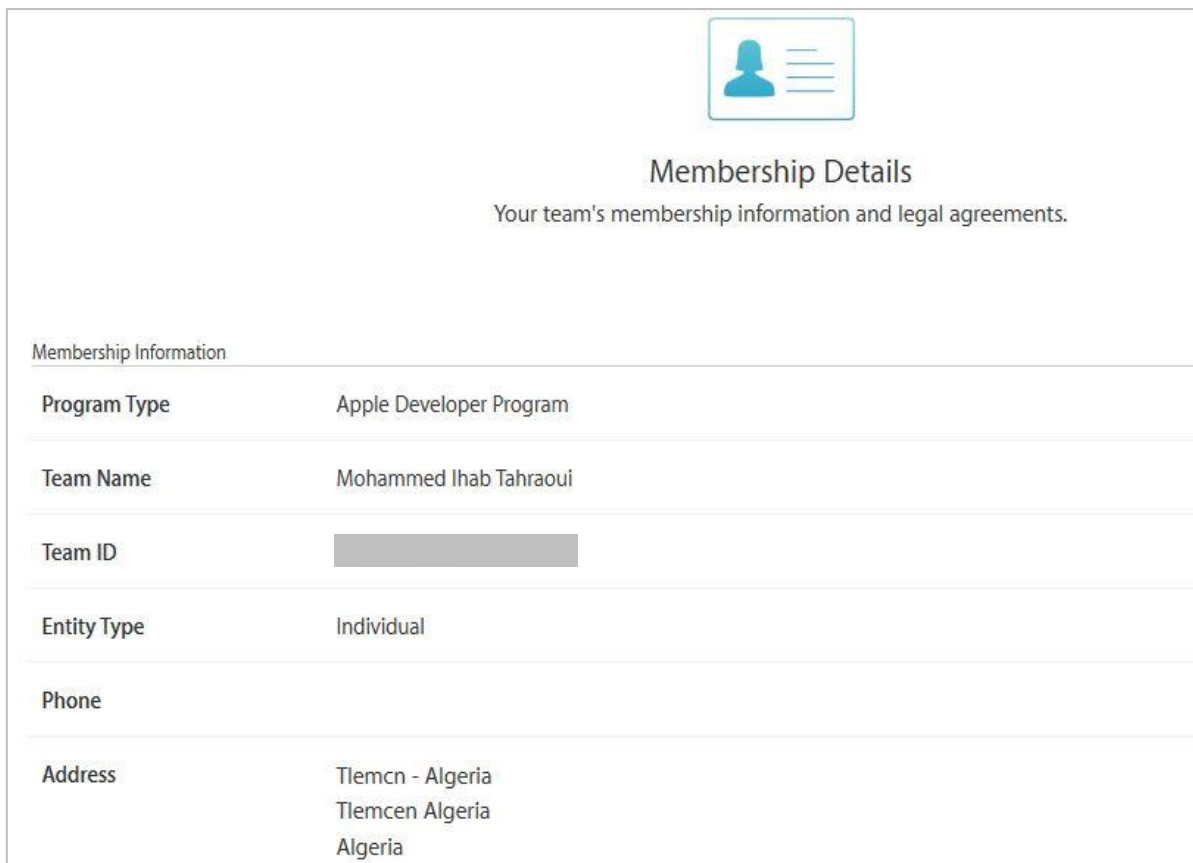


Figure 22: Certification Apple.

III.2 Logiciels

III.2.1 Logiciels de développement

Visual studio Code: Est un éditeur de code puissant, prend en charge les opérations de développement tel que le débogage et l'exécution des tâches, il fonctionne sur Windows, MacOS et linux. [21]



Workbench: est un outil qui permet de concevoir, modéliser, générer, et gérer visuellement des bases de données, il est disponible sur Windows, linux, MacOS.[22]



Xcode: Est un éditeur de texte ultra rapide pour la création des applications pour Mac et iPhone, dispose d'une interface qui unifie le design, la programmation, le test et le débogage.



Android Studio: Est un environnement de développement, en abrégé IDE, permet de développer et tester des applications sous Android. [23]



DB Browser for SQLite: Est un outil visuel, open source de haute qualité pour créer, concevoir et éditer des fichiers de base de données compatibles avec SQLite.[24]



JetBrains PhpStorm 2016: Est un environnement de développement adapté au langage de programmation PHP, pour créer et modifier le code source, quel que soit le langage de programmation utilisé.



III.2.2 Logiciels de Modélisation

Modelio : Est un environnement de modélisation open source, offre plusieurs fonctionnalités pour les développeurs de logiciel, les analystes et les concepteurs. [25]



III.2.3 Logiciels d'édition de texte

Adobe Acrobat XI Pro : Est outil qui permet de créer, fusionner et modifier des documents de format PDF, il dispose d'une fonctionnalité spécifique dédiés à la création des formulaires web.



III.2.4 Logiciels de traitement d'images

Photoshop CS6 : Est un logiciel de retouche, de traitement de création de photo avec un excellent mode de qualité.



III.2.6 Autres Logiciels

Baidu Wifi HotSpot : Est un outil qui permet de partager la connexion, il peut aussi recevoir des fichiers provenant de téléphone portable connecté au réseau. [26]



III.3 Technologies utilisées

IONIC : Est un Framework coté client open source de création des applications mobile multi plateformes à la façon Hybride, avec nos propres connaissances des technologies Web. [27]



APACHE CORDOVA : Est un Framework gratuit et open source, sert à la création des applications Hybride (Application mobile multiplateforme) en utilisant une combinaison de technologies d'applications natives et Web. [28]



TypeScript : Est un contrôleur coté client avec un langage de programmation qui a mené une couche supplémentaire à JavaScript tel que l'orienté objet et beaucoup d'autres fonctionnalité qui rendent le développement des applications web ou d'autre projet plus élégant [29]



AngularJs : Est un Framework JavaScript coté client trop puissant pour développer Une application complexe, peut être ajoutée au HTML pour déclarer des pages dynamiques, Il suit le model d'architecture MVC. [30]



NodeJs : Est un langage interpréter coté serveur écrit en JavaScript, il mit en disposition des bibliothèques JavaScript selon le besoin grâce au gestionnaire de baquet NPM.



JSON : Est l'acronyme de JavaScript Object Notation. C'est un langage JavaScript pour échanger et structurer les données textuelles. il ajoute un niveau supplémentaire vers la programmation orienté objet. [31]



SQLite : Est une base de données locale très appréciée, fournit une interface SQL, elle est très rapide en traitement et offre une empreinte mémoire très réduite. [32]



SQL : Est l'acronyme de Structured Query Language. C'est un langage qui permet de communiquer des instructions à la base de données, pour stocker et lire les données.



Socket : Sont des mécanismes E/S qui gèrent la liaison entre client et serveur en se basant sur des couches réseaux de communication spécifiques.

Swift : Est un langage de programmation multi paradigme, il est développé pour faciliter la réalisation des applications sure IOS et Mac. [33]



IV. Pourquoi IONIC et pourquoi la programmation Hybride ?

Il y'a 2.5 milliards de téléphones intelligents dans le monde, mais ces Smartphones ne sont pas homogènes au niveau de système d'exploitation (IOS et Android les plus populaires) ainsi qu'ils sont disponibles avec des tailles différentes. Ces deux critères représentent un problème pour les développeurs des applications mobiles, car pour utiliser des outils de débogages et des instruments pour chacune de plateformes mobiles c'est un travail difficile. Et c'est pour cela que les plateformes de développement hybride apportent aux développeurs un soulagement face à des difficultés rencontrées. IONIC a émergé comme un cadre multiplateforme populaires, il est open source avec des documentations bien détaillées, il fournit une large gamme d'outils et de services qui facilite le développement.

Ionic se repose sur HTML, CSS3, cordova et angular js, avec des SDK natifs des plateformes IOS et Android. La figure 24 représente la vue de haut niveau de l'application ionique. [34]



Figure 24: la vue de haut niveau de l'application.[34]

V. Prototypes

V.1 Architecture

V.1.1 Architecture de fonctionnement d'application

La figure 25 résume le scénario de notre application en montrant quelques fonctionnalités avec chacun de nos utilisateurs.

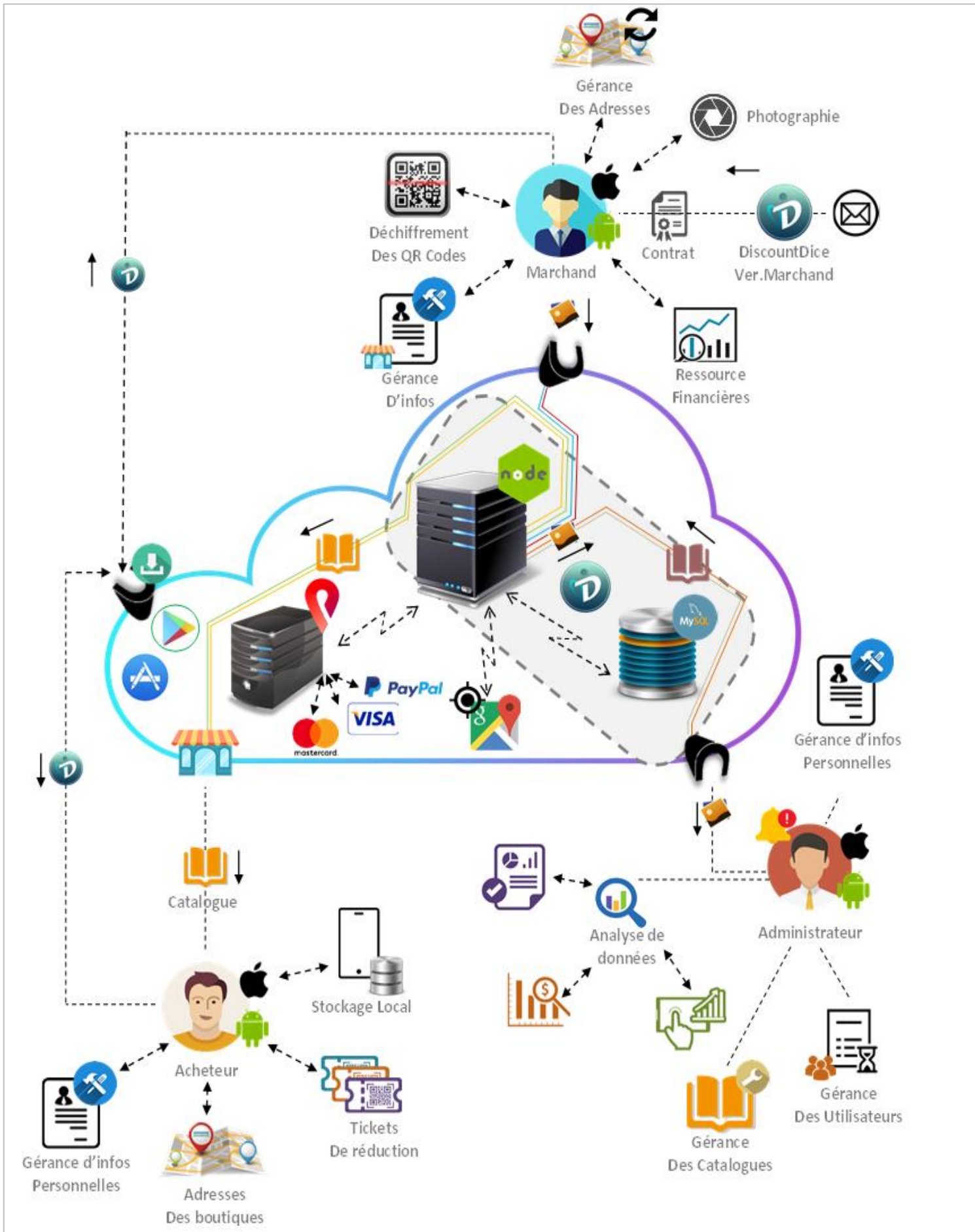


Figure 25: Schéma des fonctionnalités de l'application.

La figure 25 résume le scénario de notre application en montrant quelques fonctionnalités avec chacun de nos utilisateurs.

Expliquant notre système en le divisant en quatre parties (plateformes distante, Application Administrateur, espace Marchand et la partie acheteur) sachant que les applications sont toutes disponibles sous version **Android** et **IOS**.

Commençant par la partie plateforme, là où se retrouve toutes les parties prenantes de notre système ainsi notre partie serveur. On trouve comme plateforme les boutiques d'applications **AppStore** et **PlayStore** là où nous allons mettre l'application dédiée aux acheteurs en disposition, aussi la plateforme **Pin payments** comme prestataire responsable sur les transactions de paiement en ligne en gérant tous types de cartes de crédit, nous avons aussi comme plateformes la partie **Google Map** là où nous avons pu enregistrer une **API Key** afin que nous puissions localiser les boutiques de nos marchands en récupérant leurs longitude et latitude qui vont faciliter le chemin aux acheteurs lors la récupération de leurs articles.

Passant aux espaces utilisateur, nous commençons par l'administrateur qui a comme possibilité une large analyse de données, une gérance de catalogues ainsi que les utilisateurs (**Marchands et Administrateurs**). Nous avons aussi l'espace Marchand qui ne peut pas être sauf si le contrat est signé en satisfaisant certains critères spécifiques tel que l'obligation de disposition d'une boutique agréée, ensuite notre nouveau fournisseur aura une application de boutique **DiscountDice Version Marchand**, Là où il peut gérer ses articles qui vont être visible dans le monde d'internet qui garantit une grande publicité, sachant que les articles ne vont pas être mis en ligne sans l'accord de l'administrateur, aussi il a comme fonctionnalités la gérance des ressources financières tel que la gestion de stocke et le revenu constaté avec l'entreprise **DiscountDice**, déchiffrement et validation des tickets de réduction présenter par les récupérateurs (**acheteurs**) des articles vendus en ligne, ainsi qu'un historique de validation de ces derniers.

Finalement nous avons l'internaute le plus important dans notre système qui est l'acheteur, il peut exploiter nos articles après avoir eu DiscountDice de près des stores des applications et s'inscrire et puis aura en disposition une large gamme de fonctionnalités. Tel que l'exploitation efficace des articles, la recherche rapide, la consultation des tickets de réduction stockés dans le portable et la gérance des informations privés.

Passant au cœur de notre Marketplace, qui est le **prix dynamique** de l'article qui peut varier de 0.01£ jusqu'à un prix inférieur sur le prix vitrine de la boutique physique en respectant des règles de gestions ainsi de sécurités.

V.1.2 Architecture des technologies utilisées

La figure 26 représente l'architecture de notre système ainsi que la relation entre les différentes technologies utilisées :

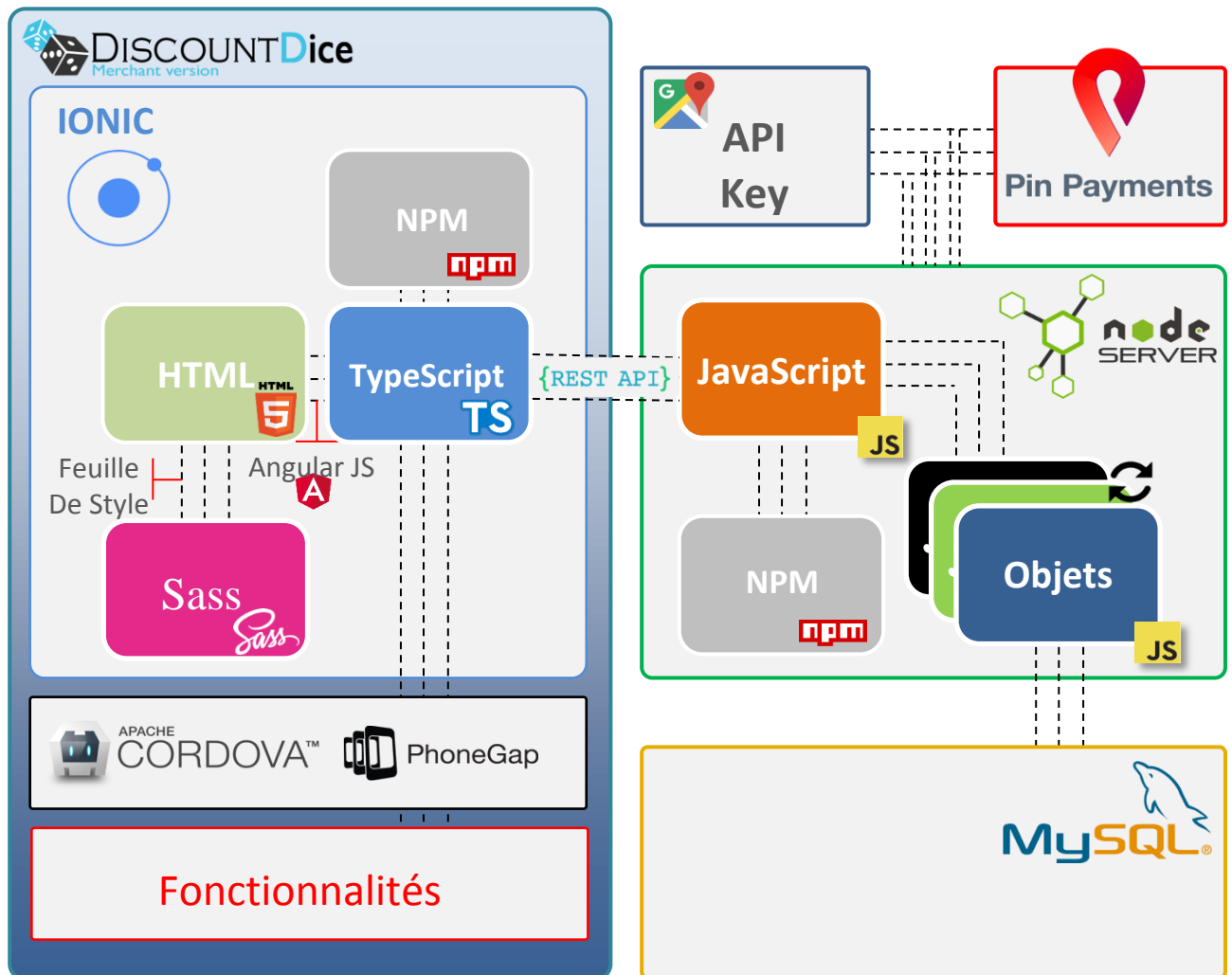


Figure 26: Architecture de système.

La partie **DiscountDice** représente l'architecture dans le Smartphone où se trouve les différentes technologies utilisées du Framework IONIC et la relation entre les fonctionnalités natives et ce dernier qui est géré par le Framework **Apache Cordova/Phone Gap**. Ensuite on a une relation entre l'application et le serveur distant géré par **L'API REST**.

La partie serveur a comme environnement la technologie **Node Js** qui se base sur des Framework **javascript** géré par le gestionnaire de paquets **NPM**, concernant le code métier est aussi en javascript structuré en orienté objet. Notre serveur a un SGBD **MySQL** interconnecter via le connecteur **JDBC**, il a des relations externe avec la plateforme **Pin Payment** qui est responsable sur tout type de transaction de paiement ainsi qu'une relation avec **Google Map** afin de localiser les adresses des nouveaux boutiques rapidement.

La figure 27 représente les boutiques qui possèdent l'option **DiscountDice** :



Figure 27: Emplacement boutiques avec DiscountDice.

V.1.3 les entrepôts de données

Le processus de réduction dynamique nous l'avons testé et analysé en effectuant 500 tests, nous avons pu les analyser en réalisant un entrepôt de Données (Data WareHowse) en respectant l'architecture et les technologies représentés dans la figure 28:

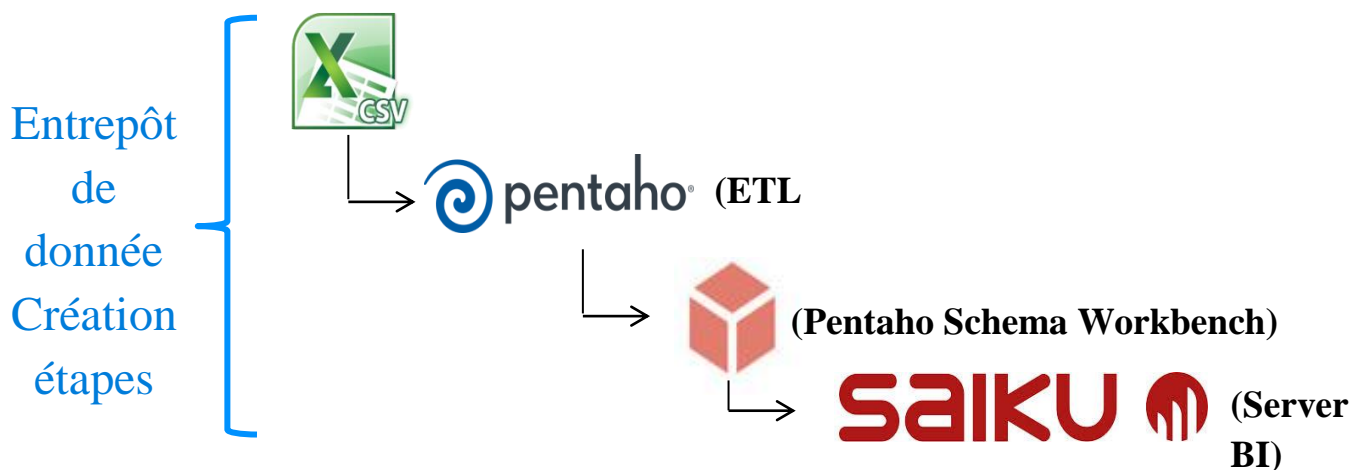


Figure 28: Création d'entrepôts donnés.

VI. Caractéristiques

VI.1 Caractéristiques fonctionnelles

Dans cette partie nous allons spécifier l'ensemble de caractéristiques fonctionnelles de notre application :

Coté client

L'inscription : Un compte est attribué pour chaque client inscrit pour avoir plus de visibilité sur le contenu de l'application et d'avoir plus d'accès sur les ressources qu'il contient.

La consultation des articles : l'application dispose d'une vitrine virtuelle à travers laquelle le client peut consulter une grande variété des produits et il a la possibilité de les lister par des catégories et des sous catégories.

La recherche personnalisée : le client peut effectuer une recherche d'article avec son nom ou un de ses caractéristiques.

La réservation d'un produit : Le système doit permettre au client de réserver un produit dans les catalogues.

Le paiement : C'est une phase très sensible, pour cela il faut qu'elle soit très sécuriser, pour terminer la procédure de paiement avec succès le client doit indiquer le numéro de sa carte et sa valeur de vérification CVV.

Consultation d'historique d'achat : le client peut lister des transactions d'achat et consulter les tickets.

Coté marchand

La gérance des produits : le marchand ajoute un nouveau article, modifier leurs caractéristiques et augmenter sa quantité, il peut aussi le supprimer.

L'exposition des produits: Il sera donc indispensable d'y présenter les prix et les caractéristiques techniques de chaque produit pour faciliter la sélection du produit à acheter.

La gestion des stocks : le marchand gère son stock et illustre ses articles par plusieurs photos.

La gestion des accès aux statistiques : le marchand consulte les statistiques de son finance.

Coté administrateur

La gestion des catalogues : l'administrateur mis à jour les catalogues il peut ajouter, modifier et supprimer des catégories ou des sous catégories.

La gérance des catégories : Affichage de l'ensemble des caractéristiques de chaque produit dans une organisation homogène.

La validation des articles : l'administrateur valide la publication des articles des fournisseurs.

La gestion des statistiques : l'administrateur consulte les statistiques de différentes données de l'application sous formes des diagrammes.

Le test de gain des articles : l'administrateur teste le gain de l'article avant de décider la validation.

La gestion des utilisateurs : la possibilité d'ajouter des profils (administrateurs / fournisseur).

VI.2 Caractéristiques non fonctionnelles

Dans cette section nous allons spécifier l'ensemble de caractéristiques non fonctionnelles de notre application :

La fiabilité: L'application doit fonctionner de façon cohérente sans erreurs et doit être satisfaisante.

L'ergonomie: L'application doit être adaptée à l'utilisateur et à tous types de Smartphones.

La sécurité : L'application doit la confidentialité des données personnelles des clients et surtout le cryptage des cartes crédits.

La réutilisation : L'application est conforme à une architecture standard permettant sa réutilisation.

La confidentialité : Il s'agit tout d'abord d'interdire l'accès en lecture et en écriture aux informations considérées comme privées à chaque utilisateur.

Les droits d'accès : L'accès sera contrôlé selon le profil et le type d'utilisateur.

VII. Conclusion

Au cours de ce chapitre nous avons présenté la phase de réalisation avec les différents technologies utilisées ainsi que l'environnement logiciel et de développement de notre travail, nous avons présenté aussi une architecture globale de l'application.

Chapitre V

Manuel de l'application

I. Démonstration des interfaces

Notre application n'est pas basée sur une plateforme existante déjà, nous l'avons développée du début avec nos propres contraintes qui suivent les règles d'e-business. C'est une application avec une ergonomie hiérarchique, et utilisation du filtrage des données dans la majorité des pages par la recherche indexée avec des mots clés pour un usage facile.

Notre marketplace a comme type commerciale B2B (entre entreprises) et B2C (entre entreprise et un grand public). Elle est d'une approche généraliste ou elle offre une grande gamme de produits dans plusieurs catégories et ne se concentre pas que sur un secteur d'activité précis. Cette mixité d'article doit être bien maîtrisée, chaque produit correspond à une sous-catégorie et cette dernière dans sa catégorie qui la convienne. Cette hiérarchie est suivie par un opérateur de type pure Player.

Le vendeur n'a pas l'accès à l'application comme un marchand qu'après la validation et l'acceptation du contrat.

***N.B :** Notre application s'est développée en une version anglaise.*

Dans cette section on va présenter quelques principales interfaces de notre application sur les deux plateformes (Android & IOS) en indiquant les différentes itérations plateforme **IOS** s'est indiqué dans le côté gauche, et **Android** qui est dans le côté droit.

Partie Client

Interface inscription

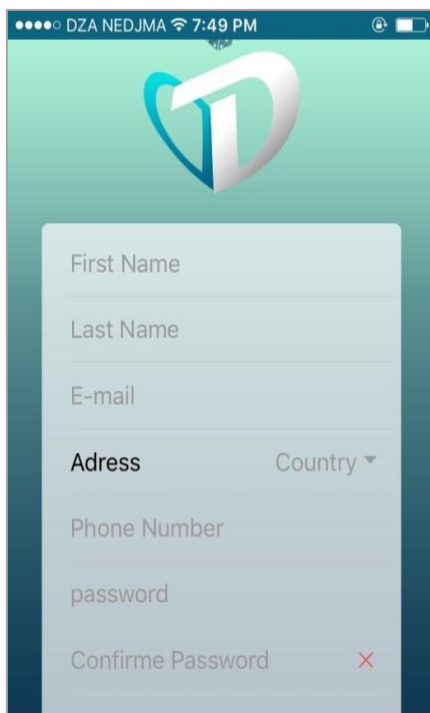


Figure 29: Interface inscription IOS.

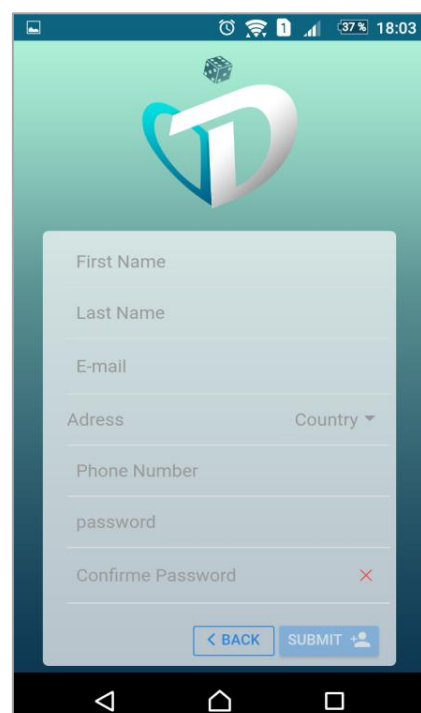


Figure 30: Interface inscription Android.

Les figures 29 et 30 présentent l'interface d'inscription, Ce formulaire s'affiche lorsqu'un internaute veut s'inscrire à notre application. Le système vérifie l'existence de l'email et la confirmation de mot de passe au temps réel, si tout est valide, il sera un nouveau client sur notre plateforme.

Interface authentication

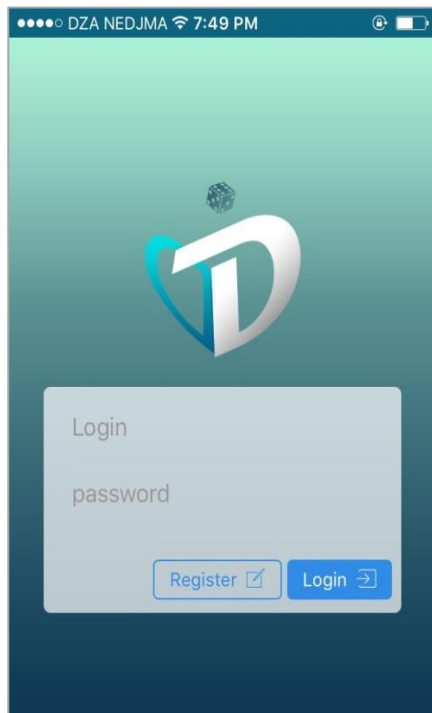


Figure 31: Interface authentication IOS.

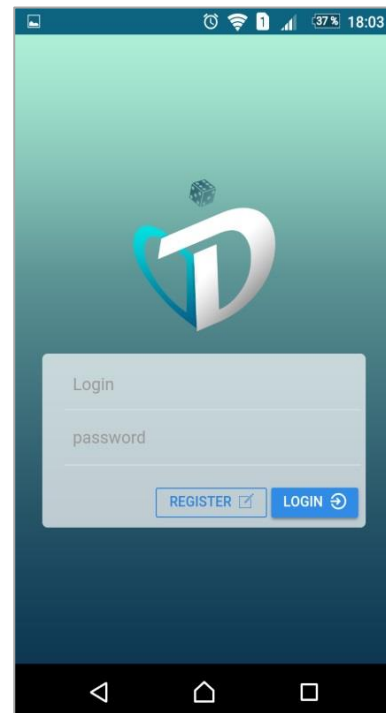


Figure 32: Interface authentication Android.

Les interfaces 31 et 32 s'affichent lorsque le client veut se connecter pour accéder à son compte, il introduit son login et mot de passe puis il valide en cliquant sur le bouton login, sinon s'il n'a pas encore un compte il clique sur « **register** » pour s'inscrire.

Interface accueil « sidebar »

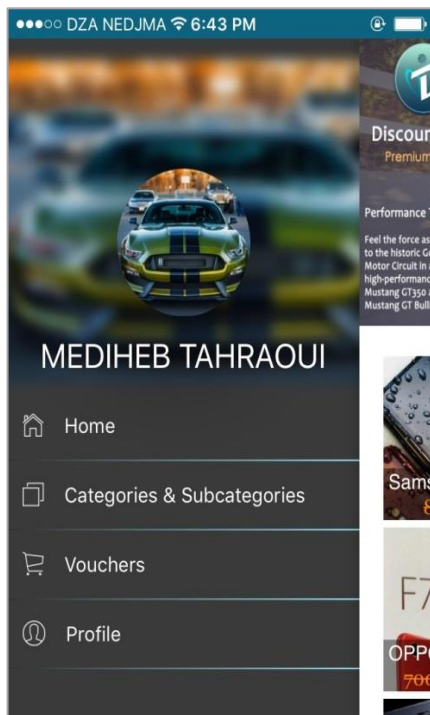


Figure 33: Interface accueil "sidebar" IOS.

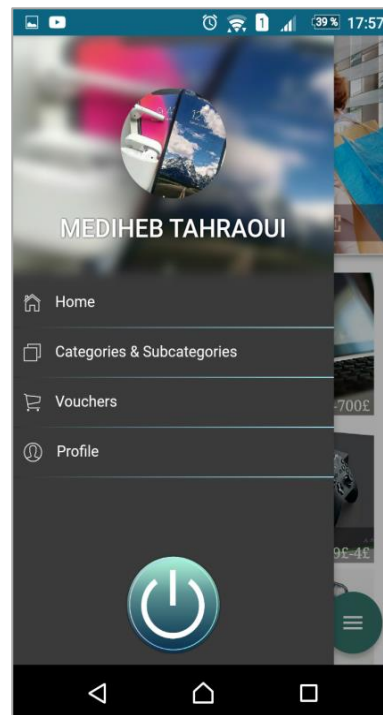


Figure 34: Interface accueil "sidebar" Android.

Les figures 33,34 présentent l'interface d'accueil et plus précisément la section sidebar qui contient quatre roots:

- **Accueil** : la page vitrine qui affiche les derniers articles publiés.
- **Catalogues & sous catalogues** : la page qui contient la répartition des articles par catégories ainsi le filtrage dynamique afin de faciliter la navigation.
- **Achats** : la page qui contient la liste de tous les transactions de paiement qui sont représenté par des tickets de réduction qui vont servir à la récupération des articles.
- **Profil** : la page qui contient les informations privées du client avec la possibilité de les modifier.

Partie Accueil Client

Passant au root Accueil :

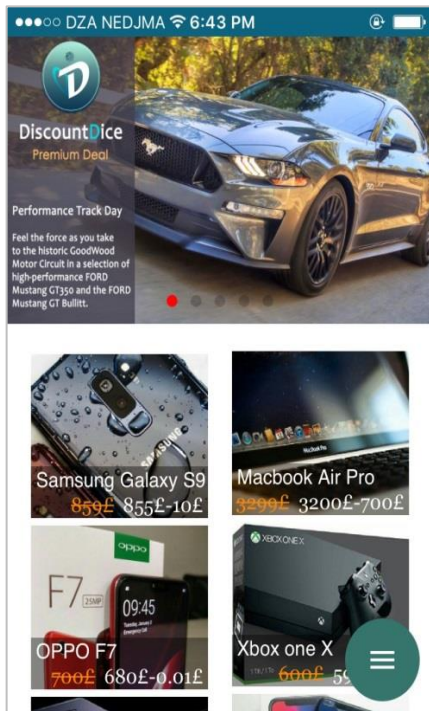


Figure 36: Interface accueil client IOS.

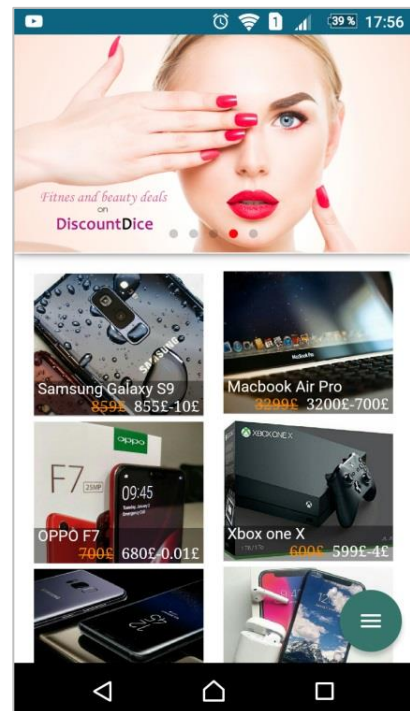


Figure 35: Interface accueil Android.

Les figures 35 et 36 représentent la page d'accueil qui s'affiche dès qu'un client se connecte, elle affiche les articles en mode **LIFO** avec leurs prix minimum et maximum.

Interface détail d'article

Représentant la procédure d'achat :



Figure 37: Interface détail article IOS.

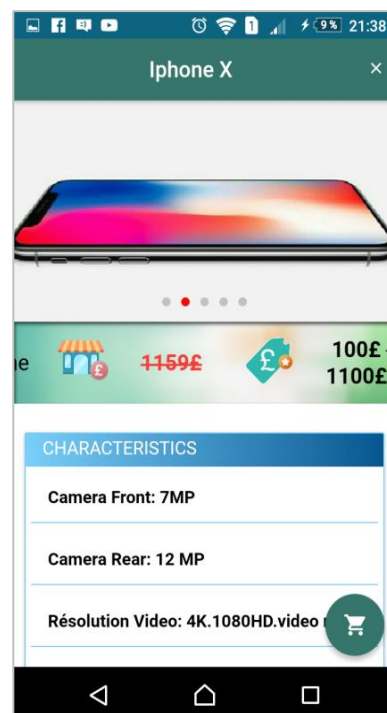


Figure 38: Interface détail article Android.

Les figures 37,38 représentent l'interface des informations d'un article choisi par le client. Elle a des photos descriptives du produit, les détails de prix, les caractéristiques, une description et les coordonnées nécessaires sur les lieux de disponibilité de produit telle que le nom de la boutique ainsi que l'adresse exacte associée à la fonctionnalité native GPS afin de faciliter la localisation de la boutique.

Interface validation d'achat

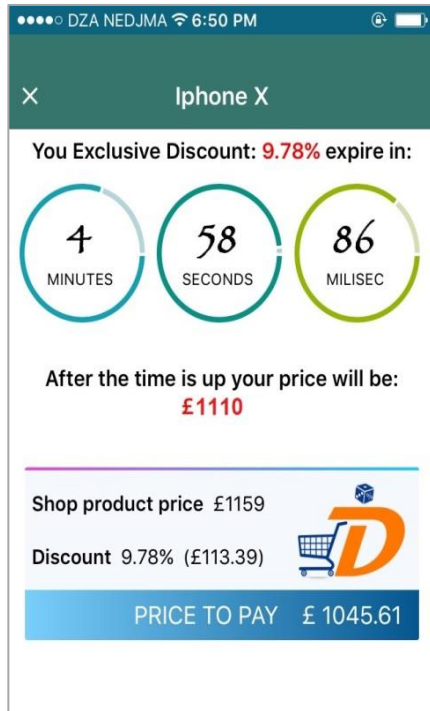


Figure 40: Interface validation achat IOS.

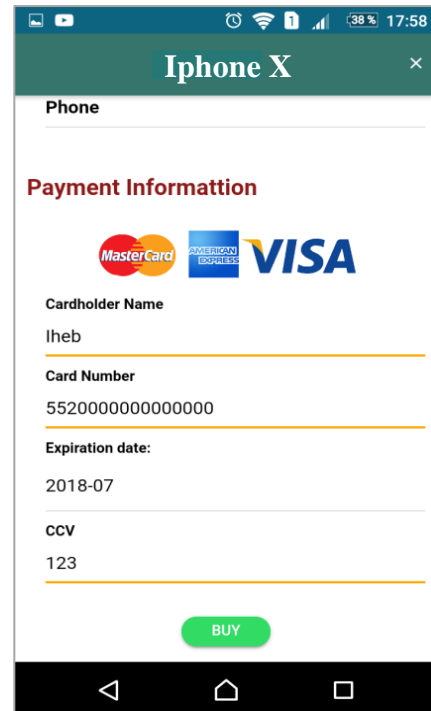


Figure 39: Interface validation achat Android.

Les figures 39, 40 représentent la partie validation de la transaction d'achat, la première étape est la génération d'un pourcentage de réduction de prix. Si le client valide l'achat avant les 5 minutes il paye le prix affiché après la réduction, sinon il aura une marge de prix additionnelle.

Interface facture

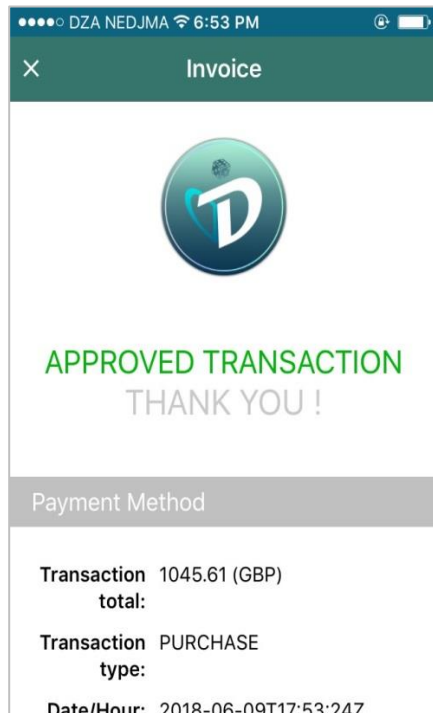


Figure 41: Interface facture IOS.



Figure 42: Interface facture Android.

Les Figures 41 et 42 représentent la facture qui s'affiche après une bonne validation d'achat avec un **QR Code unique** associé à cette transaction comme justificatif d'achat. Sachant que la procédure d'achat est la plus sécurisée car c'est l'étape la plus critique et ciblée de la part des hackers. Pour ce faire nous avons haché les données de la carte de crédit de l'utilisateur en utilisant le cryptage de **JULES CESAR** en générant des clés aléatoires lors de la transmission des données entre le client et notre serveur.

Interface Tickets

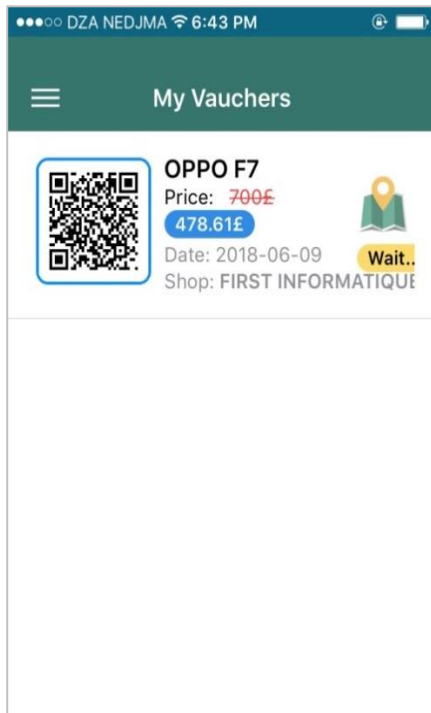


Figure 44: Interface ticket IOS.

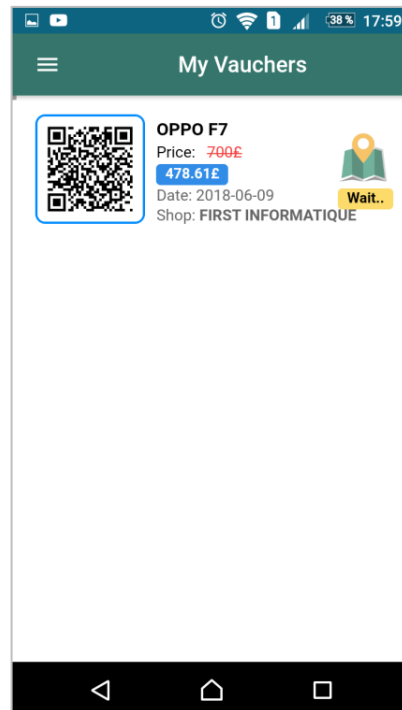


Figure 43: Interface ticket Android.

Les Figures 43 et 44 représentent la liste des tickets de réduction générée après la validation d'achat. Ces derniers sont caractérisés par un **QR Code unique** qui contient le nom du produit ainsi que sa référence dans la boutique physique afin de rendre la récupération de produit plus rapide, aussi la localisation de la boutique en utilisant le **GPS**. Sachant que tous les tickets sont stockés dans une base de données interne du téléphone afin de garantir l'utilité de l'application même s'elle est en mode déconnecté, ce qui facilite l'usage auprès des utilisateurs.

Partie Marchand

Interface d'accueil

Représentant la partie de publication d'article :

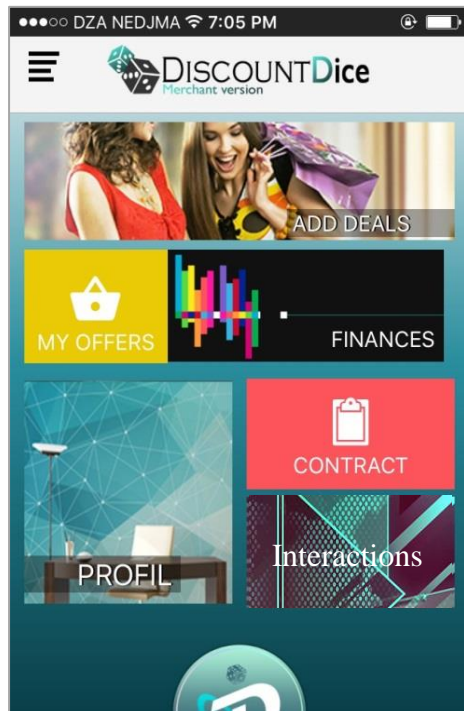


Figure 45: Interface accueil marchand IOS.

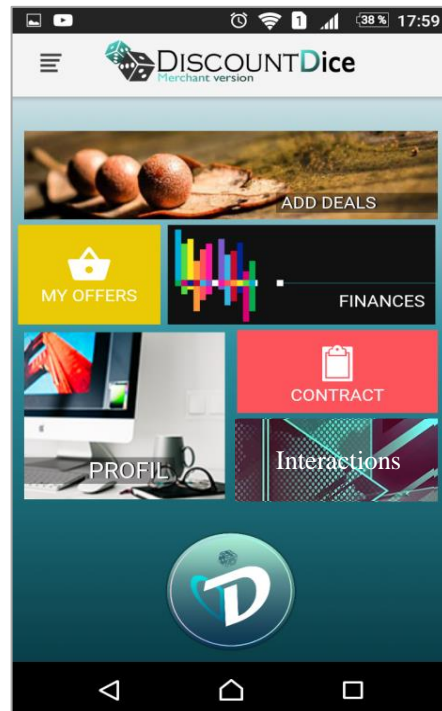


Figure 46: Interface accueil marchand Android.

Les figures 45, 46 représentent la page d'accueil qui s'affiche dès l'authentification de marchand après l'activation de compte de la part de l'administrateur, elle contient sept boutons :

- **Ajouter article** : La page primordiale qui sert à la publication d'article après avoir mentionner tous informations nécessaires et un accord auprès de l'administrateur.
- **Mes offres** : La page qui contient tous articles introduit avec ses informations.
- **Finances** : La page qui contient les informations de finance telle que la somme d'argent qu'il va gagner chaque mois et le type article le plus vendu.
- **Contrat** : La page qui contient les informations de contrat et le nombre de jours restants.
- **Interactions** : La page qui contient tous type d'interactions telle que la gérance de stocke ainsi qu'une traçabilité sur les produits récupérés et les tickets validés.
- **Profile** : Là où se trouvent les informations privées de marchand ainsi que les informations de sa boutique avec la possibilité de les modifiées.

Interface article

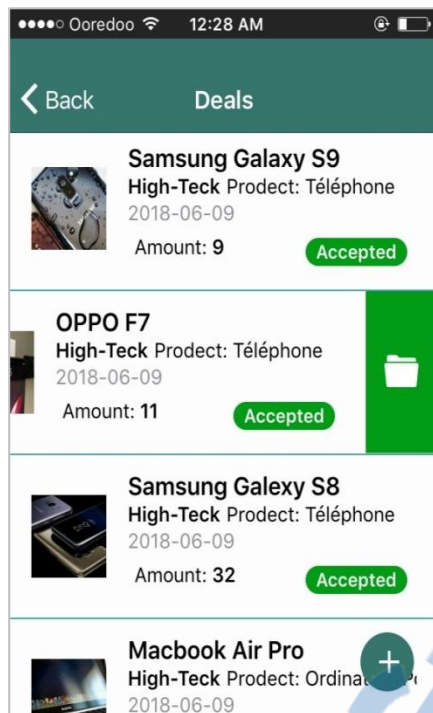


Figure 47: Interface article IOS.

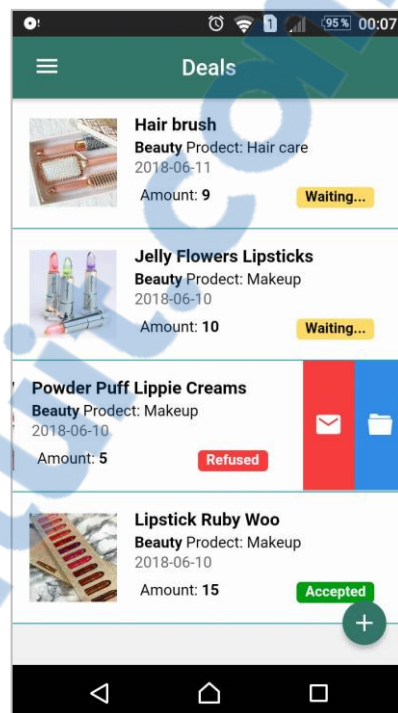


Figure 48: Interface article Android.

Les figures 47 et 48 représentent la liste de tous les produits du marchand avec les différents statuts :

- **Accepted** : article accepté par l'administrateur, il sera publié directement sur le store.
- **Waiting** : article en attente à la prise de décision de la part des administrateurs.
- **Refused** : article s'est refusé en joignant une lettre justificative indiquant les raisons dont l'offre n'est pas être acceptée.

Pour chaque item de la liste il y'a un bouton pour accéder aux détails d'articles désirer et un autre pour supprimer les offres.

Interface historique

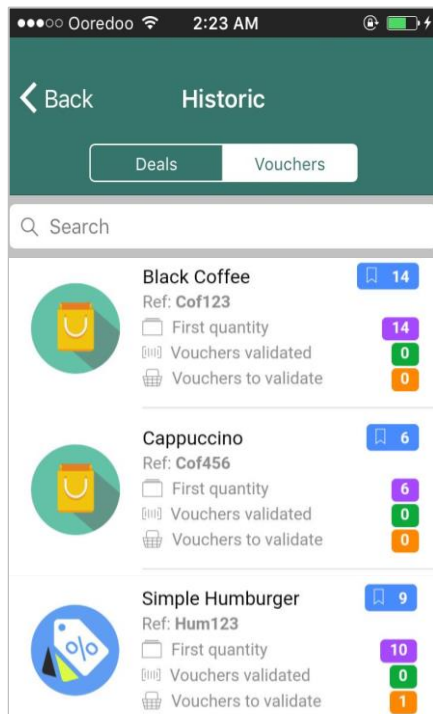


Figure 49: interface historique IOS.

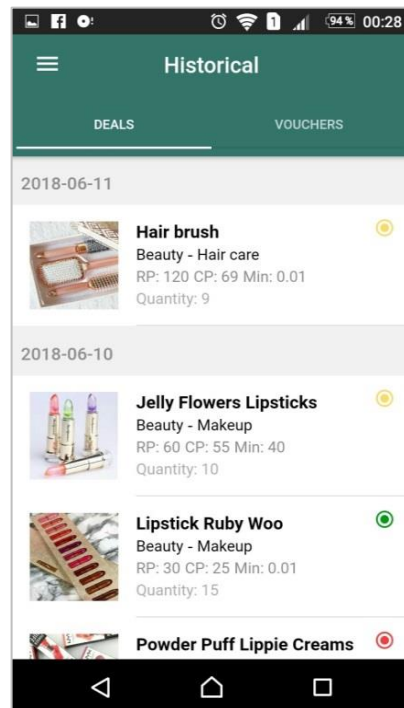


Figure 50: Interface historique Android.

Les figures 49, 50 représentent la page historique des interactions dont elle est caractérisée par un segment qui contient **deals** (Vision globale sur le stock) et **vouchers** :

- **Vouchers** : contient l'historique de tous les interactions d'achat.
- **Deals** : contient l'historique des articles trié par date, avec un statut de chacun :
 - Rouge** : article refusé par l'administrateur.
 - Orange** : article en attente.
 - Vert** : article disponible sur le store.

Partie administrateur

Interface accueil « sidebar »

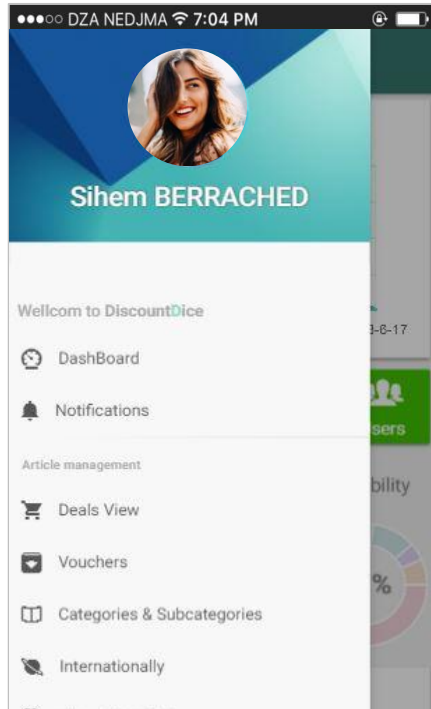


Figure 52: Interface accueil administrateur "sidebar" IOS.

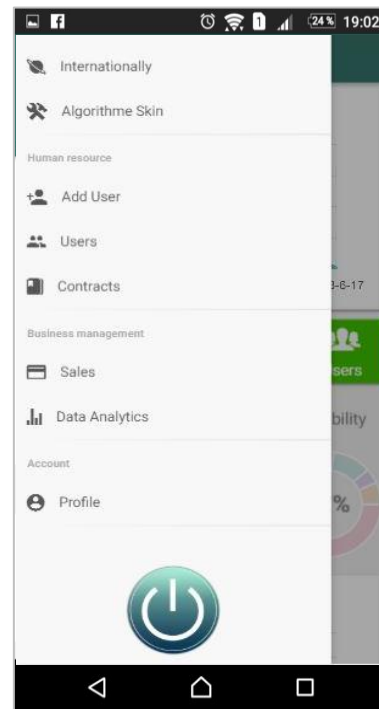


Figure 51: Interface accueil administrateur "sidebar" Android.

Les interfaces 51 et 52 représentent le menu latéral avec les différentes fonctionnalités:

- **Tableau de bord** : contient la partie analyse de données avec des différents digrammes comme le nombre de vente par jour, le gain, le nombre d'article par catégories et d'autres statistiques.
- **Notifications** : cette page contient 3 segments notification de demande d'ajout d'un nouvel article, notification de demande d'ajout d'un nouveau marchand et notification de renouvellement du contrat d'un marchand existe déjà.
- **Ajouter utilisateur** : contient un formulaire dynamique avec des champs variés selon le type d'utilisateur voulu ajouter (administrateur ou marchand).
- **Utilisateurs** : cette page contient une liste des utilisateurs avec trois segments : administrateurs, marchand et client.
- **Ventes** : contient des informations concernant les finances des ventes.
- **Articles** : contient la liste de tous les articles avec leurs détails ainsi qu'à quel marchand appartient ce produit.
- **Catégories et sous catégories** : contient tous les catégories et leurs sous catégories, et la possibilité de créer des nouveaux.
- **Configuration algorithmes** : pour la configuration des algorithmes de génération des pourcentages et réductions et de prix.

- **International** : contient tous les pays du monde afin de pouvoir activer la disponibilité de l'application côté marchand pour quelques pays et désactiver pour d'autres.
- **Vouchers** : contient tous les articles de tous les marchands avec la quantité initiale, la quantité restante, le nombre des vendus récupérés de cet article ainsi que le nombre des vendus non récupérés.
Dans cette page on a trois types d'article : article jamais vendu, article avec quelque nombre de vendu et article tout vendu (le nombre de stock zéro), on peut différencier entre ces trois types par les photos a côté chacun de ces derniers a une photo différente que l'autre type afin de faciliter l'analyse pour une première vue. Cette page contient aussi searchbar pour une recherche plus rapide.
- **Analyse de données** : contient l'analyse de données de l'application, le nombre de vente, de catégorie, de marchand ... et d'autres informations.
- **Profile** : contient les informations privées de l'administrateur avec la possibilité de les modifier.

Interface de Tableau de bord

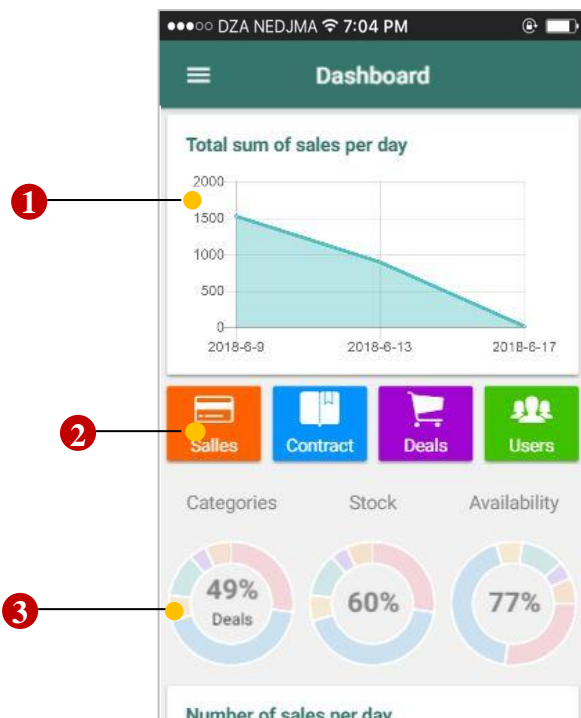


Figure 54: Interface tableau de bord IOS.

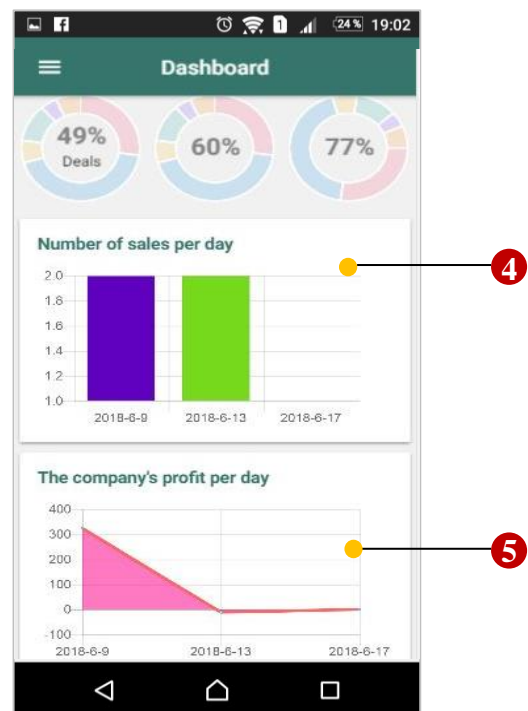


Figure 53: Interface tableau de bord Android.

Les figures 53 et 54 sont composées par plusieurs sections indiquées par les pointeurs ci-dessus qui signifient :

- **Pointeur 1** : représente un graphique des revenus totaux (gain de la société plus la somme dédié aux marchands) des interactions par jour.

- **Pointeur 2** : section qui contient les fonctionnalités les plus utilisées, pour un accès rapide.
- **Pointeur 3** : contient trois graphes sous forme de jauge principaux de notre plateforme :
 - **Catégories** : indique le nombre d'article par catégorie.
 - **Stock** : indique le nombre total restant des articles disponibles sur le store.
 - **Disponibilité** : désigne la catégorie la plus active est réciproquement.
- **Pointeur 4** : graphe qui représente le nombre de vente par jour.
- **Pointeur 5** : représente le gain de la société par jour.

Interface de notifications

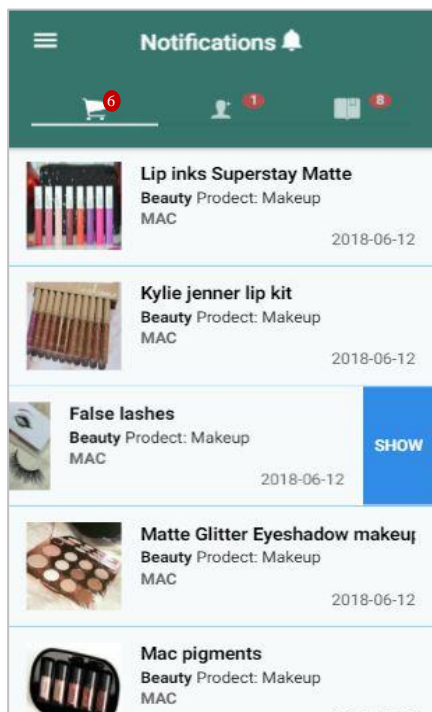


Figure 55: Interface notification article IOS.

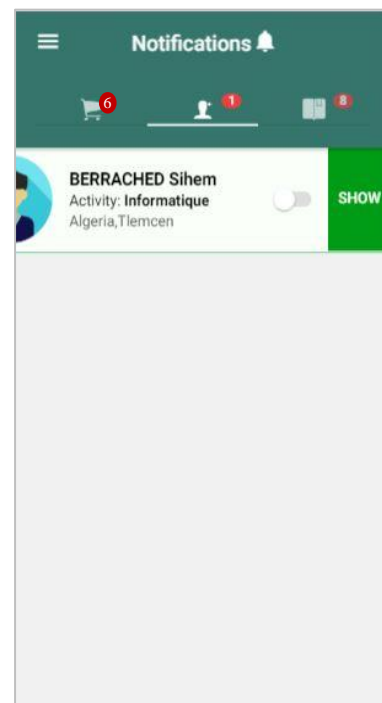


Figure 56: Interface notification utilisateur.

Les figures 55, 56 représentent la page notification à un segment qui contient les articles à validé, les marchands à confirmer et les contrats à renouvelé :

- **Segment Article** : contient les articles à validé de tous les marchands, l'administrateur peut voir les détails de ces derniers en cliquant sur le bouton « **show** » afin de décidé la confirmation de cet article ou le refus, en cas de refus il doit remplir un formulaire de justification et l'envoyé au marchand.
- **Segment Marchand** : le marchand ne peut pas s'inscrire directement sur notre application, il doit d'abord envoyer une demande d'ajout à l'administrateur qui vérifie si c'est vrai il a une boutique et puis lui envoyé le contrat par mail pour la signé, si tout est valide il va lui créer un compte.

- **Segment Contrat** : un marchand peut renouveler son contrat en envoyant une demande à l'administrateur.

Interface de Ressource humaine

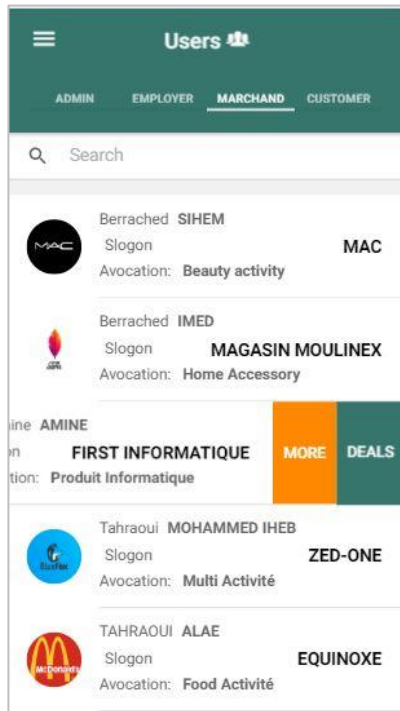


Figure 58: Interface ressource humaine IOS.

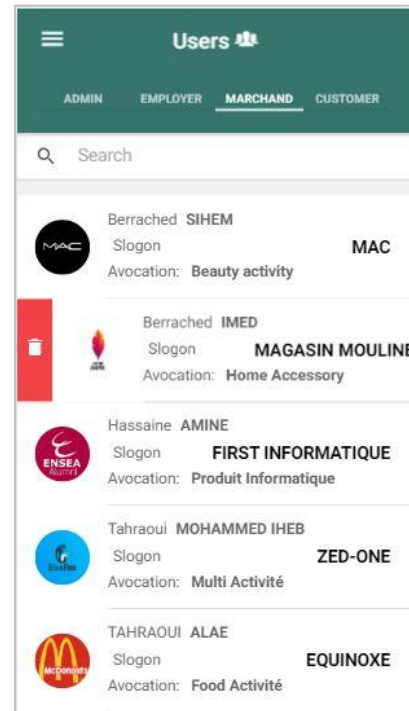


Figure 57: Interface ressource humaine.

Les figures 57 et 58 représentent l'interface de ressource humaine dont elle est caractérisée par un segment qui contient quatre types d'utilisateurs : **Administrateur**, **Employés**, **Marchand** et **Client**.

Nous présentons la partie **marchand** qui contient la liste de tous les fournisseurs avec leurs noms, leurs activités ainsi que le nom de boutique et le logo. Pour avoir plus d'information d'un marchand, nous pouvons y accéder à travers le bouton « **More** », et aussi le bouton « **Deals** » pour consulter tous les articles d'un fournisseur.

L'administrateur ne peut supprimer un marchand sauf si son contrat est expiré donc le bouton supprimer sera actif. Il peut aussi utiliser la recherche avec des mots clés.

Interface des articles par fournisseur

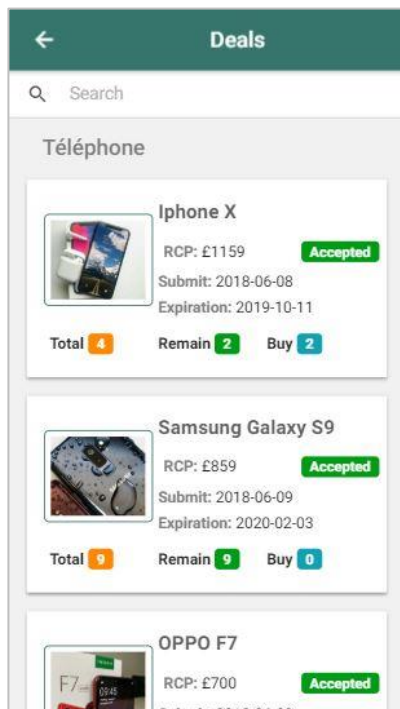


Figure 60: interface article marchand IOS.

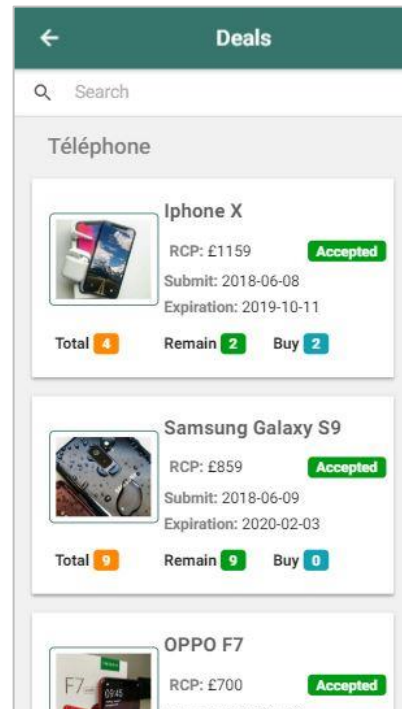


Figure 59: interface article marchand Android.

Les figures 59 et 60 représentent l'interface que nous l'accédons à travers le bouton « Deals » de la figure 58. Cette page contient tous les articles d'un marchand sélectionné avec leurs informations, tel que le nom, le nombre de produit total ainsi que le nombre des vendus et de reste. L'administrateur peut utiliser le searchbar pour une recherche plus rapide.

Interface des informations par fournisseur

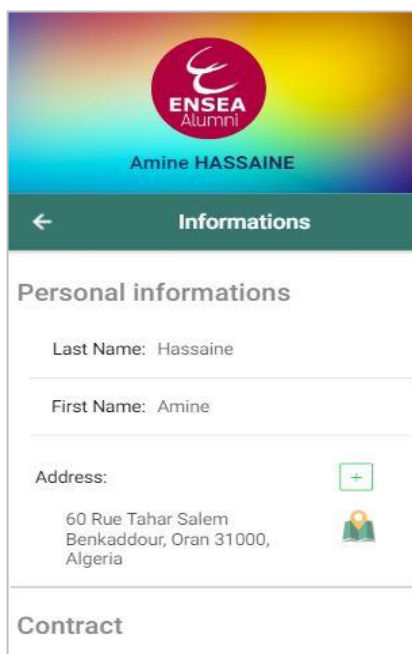


Figure 62: Interface informations par fournisseur IOS.

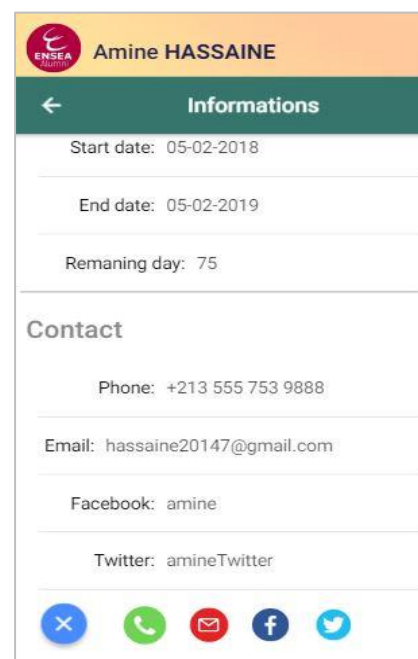


Figure 61: Interface informations par fournisseur Android.

Les figures 61 et 62 représentent l'interface que nous l'accédons à travers le bouton More de la figure 58. Cette page contient les informations personnelles d'un marchand sélectionné, l'adresse de son boutique en utilisant les coordonnées GPS, les informations du contrat ainsi que du contact.

Interface de traçabilité des articles, tickets validés et contrats

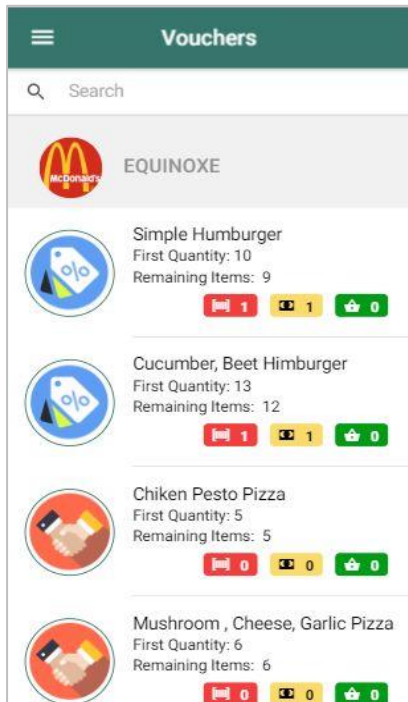


Figure 64: Interface de traçabilité des articles et tickets validés IOS.

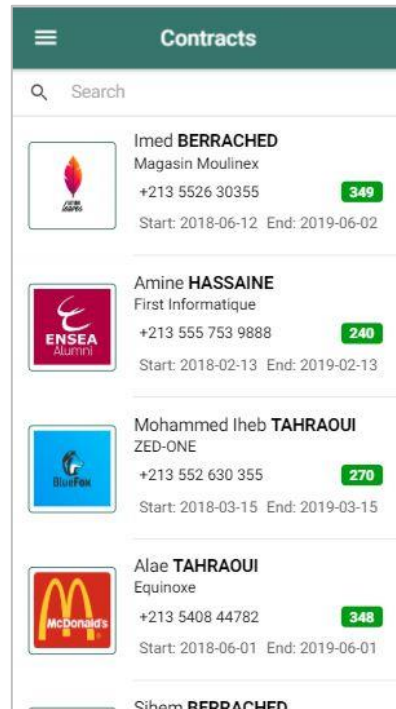


Figure 63: Interface contrat Android.

La figure 64 contient tous les articles de la plateforme triée par boutique, avec les informations de chaque produit : la quantité initiale, le nombre des produits vendus et restant.

La figure 63 contient les informations des contrats des marchands avec leur date de début et de fin ainsi que le nombre de jours restants. Un searchbar pour une recherche plus rapide.

Interface catégories

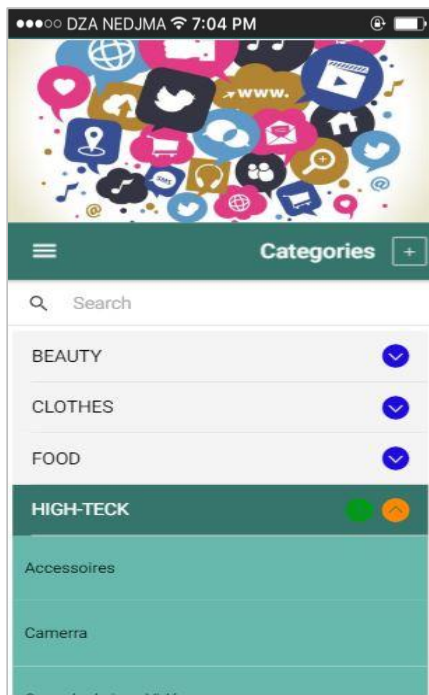


Figure 66: Interface catégorie IOS.

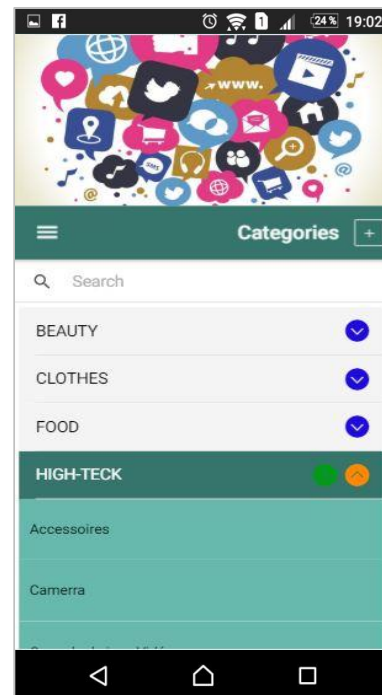


Figure 65: Interface catégorie Android.

Les figures 65, 66 représentent la page de gestion des catalogues, elle contient tous les catégories et les sous catégories, avec la possibilité de la recherche rapide grâce au searchbar. L'administrateur peut créer une nouvelle catégorie et l'associer à des sous catégories, comme il peut aussi modifier et supprimer.

Interface ajouter sous-catégorie

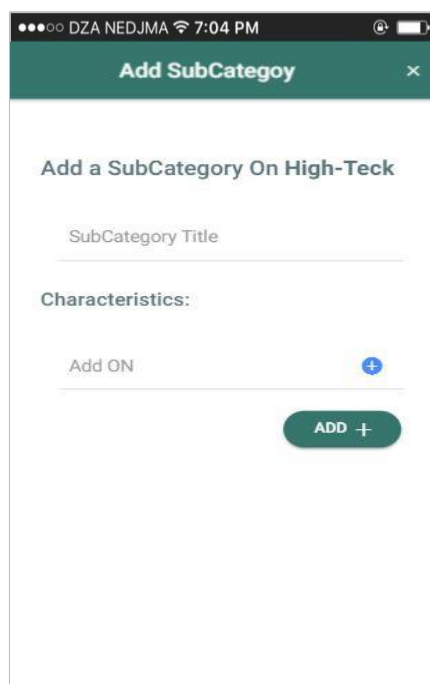


Figure 68: Interface ajouté sous-catégorie IOS.

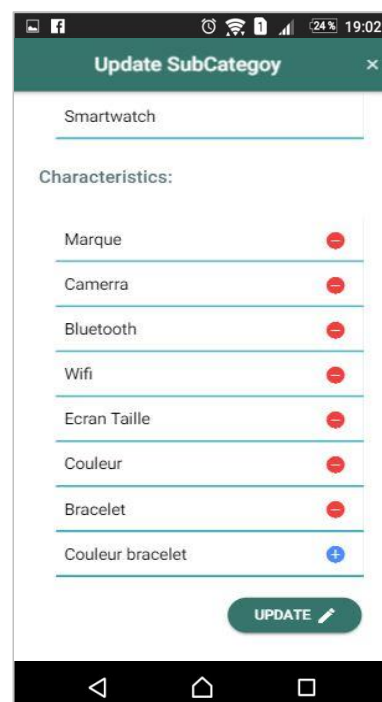


Figure 67: Interface ajouté sous-catégorie Android.

Les figures 67 et 68 ci-dessus représentent l'interface de création d'une nouvelle sous-catégorie, l'administrateur peut créer une sous-catégorie en spécifiant un nombre illimité des caractéristiques selon le besoin, comme il peut aussi supprimer.

Interface algorithmes

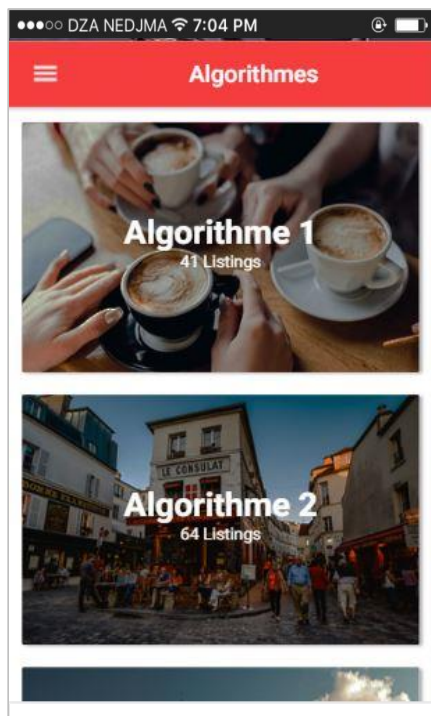


Figure 69: Interface algorithme IOS.

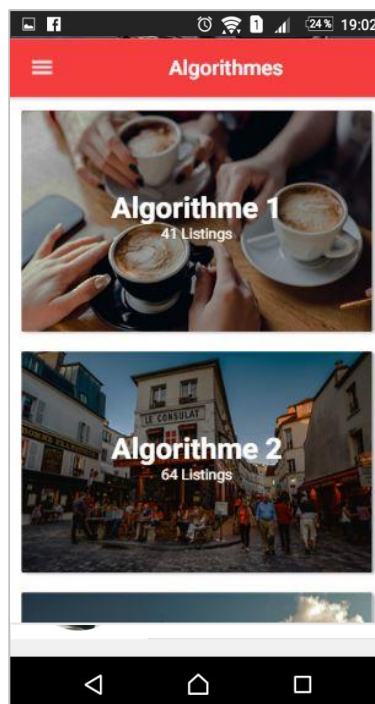


Figure 70: Interface algorithme Android.

Les figures 69 et 70 représentent la page de configuration des scripts responsables sur la génération dynamique et variée des prix.

Interface configuration des pays



Figure 72: Interface configuration des pays IOS.

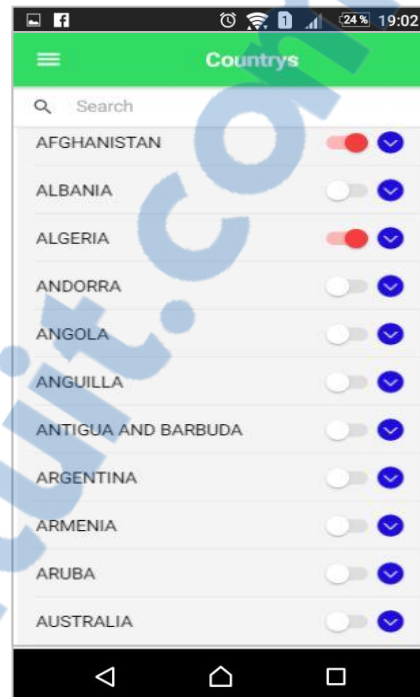


Figure 71: Interface configuration des pays Android.

Les figures 71 et 72 représentent la page qui contient tous les pays du monde avec leurs villes. L'administrateur peut activer et désactiver la disponibilité de l'application coté marchand.

Interface tester gain d'un article

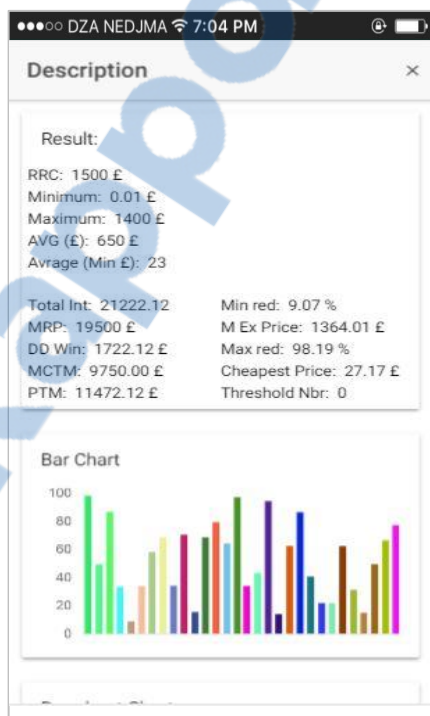


Figure 73: Interface tester gain d'article IOS.

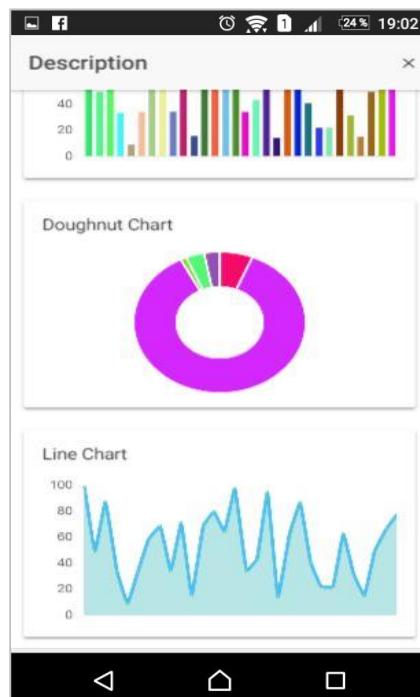


Figure 74: Interface tester gain d'article Android.

Les figures 73 et 74 ci-dessus représentent les résultats des tests des algorithmes pour un article avant de valider la publication.

Conclusion générale

Le travail présent dans ce mémoire se situe dans le cadre de la conception et la réalisation d'un marketplace sous forme d'application mobile. Dans le but de créer des boutiques virtuelles sur la plateforme pour la vente en ligne.

Afin d'évaluer la performance de notre application nous avons effectué une recherche pour connaître et analyser les systèmes similaires existants, dans le but d'améliorer et d'ajouter quelques fonctionnalités pour rendre l'application plus riche, répond aux besoins des utilisateurs et surtout pour un bon déroulement des différents gestions :

- Gestion des relations avec les clients,
- Gestion des relations avec les marchands,
- Gestion des commandes,
- Gestion des produits (ajouter, modifier ou supprimer des produits)

Pour la conception de ces fonctionnalités nous avons travaillé avec la généralisation des classes pour qu'on puisse faire les mis à jour et l'ajout des nouveau modules facilement.

Nous avons pu noter que les résultats de notre projet est très satisfaisante, et nous somme conscient que les améliorations restent toujours à faire dans ce domaine, nos futures perspectives sont nombreuses, nous citons :

-Service client virtuel au temps réel : offrir la possibilité au client de communiquer avec le représentant de la boutique s'il y'en a lieu.

-Enrichissement de l'interactivité avec les utilisateurs : votes sur les produits, commentaires, donner leurs avis

-Rendre l'application mobile « **DiscountDice** » en une application web pour une meilleure exploitation.

- L'hébergement de l'application dans App Store et Play Store.

-Notification ou message ou e-mail lors d'une nouvelle catégorie dans le catalogue.

-Intégration de l'option de recommandation des articles.

References biographiques

- [1] Colin Combe. Introduction to E-business Management and strategy, First edition 2006.
- [2] Martin Kütz. Introduction to E-Commerce: Combining Business and Information Technology, 1st edition © 2016.
- [3] Ee-Peng Lim and Keng Siau (eds). Advances in Mobile Commerce Technologies, Idea Group Publishing © 2003.
- [4] Philippe Carrot and Adrien Nussenbaum. Marketplace l'E-commerce de Demain, avec la collaboration de Clément Vouillon, Publishing 2012.
- [5] Henri Issac and Pierre Volle. E-commerce de la stratégie à la mise en œuvre opérationnelle, Pearson Education France 2008.
- [6] Marc BOURREAU and Michel GENSOLLEN. Communautés d'expérience et concurrence entre sites de biens culturels, 31 octobre 2003
- [7] Claire Borsenberger (Groupe La Poste). THE CONCENTRATION PHENOMENON IN E-COMMERCE. 22 th CONFERENCE ON POSTAL AND DELIVERY ECONOMICS June 4 th –7 th, 2014, Frascati, Rome, Italy.
- [8] Amazon : consulted 19-02-2018 : <https://www.amazon.com>.
- [9] Cdiscount: consulted 19-02-2018: <https://www.cdiscount.com/>.
- [10] Ebay: consulted 19-02-2018: <https://www.ebay.com/> .
- [11] Renaud Marlet and LaBRI / INRIA. Génie Logiciel Cycle de vie, © 2005-2007
- [12] Olivier Rebière and Cristina Rebière. Maîtriser le diagramme de Gantt: Comprendre et utiliser efficacement le logiciel open source "Gantt Project" (eGuide Education t. 2) Format Kindle, published 2017.
- [13] International Scrum institue. Scrum Reveald Training Book, seconde Edition.
- [14] Juditb Bishop. c-3-0-design-patterns, First Edition 2007.
- [15] Freeman, Adam. Pro ASP.NET MVC 5. Fifth Edition 2013.
- [16] Développement mobile web avec ionic2 version du 29/03/2017, Jean David Olekhovich <https://fr.slideshare.net/gidehault/formation-ionic-2>.
- [17] Pin Payments : Pin Payments Partner Program, consulted 06-06-2018 <https://pinpayments.com/partners/program>

- [18] Draw.io : consulted 02-06-2018 <https://www.draw.io>
- [19] Trello : consulted 02-06-2018 <https://trello.com/>
- [20] Firebase : consulted 02-06-2018 <https://www.axopen.com/firebase/>
- [21] Visual studio code : consulted 02-06-2018 <https://code.visualstudio.com/>
- [22] Workbench : consulted 02-06-2018 <https://www.mysql.com/>
- [23] Android studio : consulted 03-06-2018 <https://developer.android.com>
- [24] DB Browser for SQLite: consulted 03-06-2018 <https://sqlitebrowser.org/>
- [25] Modelio : consulted 03-06-2018 <https://www.modelio.org/>
- [26] Baidu Wifi HotsPot : consulted 03-06-2018 <https://baidu-wifihotspot.en.lo4d.com>
- [27] Joyce Justin Joseph Jude. Learn Ionic 2 Develop Multi-platform Mobile Apps, First Edition 2017.
- [28] X, JOHN M.WARGO. Apache Cordova 3 Programming Covers PHONEGAP 3, First Edition 2013
- [29] Remo H. Jansen, Vilic Vane, Ivo Gabe de Wolff. TypeScript:Modern JavaScript Development , Publishing Ltd, 2016.
- [30] Angular Js : consulted 04-06-2018 <https://angularjs.org/>
- [31] JSON : consulted 04-06-2018 https://2009.jres.org/planning_files/article/pdf/92.pdf
- [32] Mark Murphy. l'art de développement android,SecondeEdition 2010.
- [33] Rudy de Visscher .Découvrez le langage SWIFT, couvre SWIFT 2, First Edition 2016.
- [34] Joyce Justin and Joseph Jude Bonjour. Learn Ionic 2, Develop Multi-platform Mobile App,

Résumé

Notre mémoire se consacre sur l'étude, la conception et la réalisation d'une hybride application mobile Marketplace, un canal de vente en ligne qui est devenu l'avenir de l'e-commerce.

L'objectif majeur de cette application est de créer des boutiques virtuelles sur notre plateforme «**DiscountDice**», et d'exposer les catalogues des produits afin d'être à la disposition de tout le monde pour faciliter les procédures d'achat qui se base sur la génération des prix dynamiques.

Mots clés : e-commerce, e-business, marketplace, vente en ligne.

تلخيص

تتمحور مذكرتنا حول دراسة ، تصميم و انجاز مشروع تطبيق التجارة الالكترونية عبر الهواتف الذكية ، فهي عبارة عن قناة للبيع عبر الإنترنت حيث أصبحت مستقبل التجارة الالكترونية .

الهدف الرئيسي من هذا التطبيق هو انشاء متاجر افتراضية على « **DiscountDice** » و عرض منتجاتها لتكون متاحة للجميع ، و لتسهيل اجراءات الشراء التي اساسها هو توليد اسعار متغيرة .

الكلمات الرئيسية: التجارة الإلكترونية، الأعمال الإلكترونية، السوق، المبيعات عبر الإنترنت.

Abstract

Our dissertation is dedicated to the study, design and realization of a hybrid mobile application Marketplace, an online sales channel that has become the future of e-commerce.

The main objective of this application is to create virtual shops on our platform «**DiscountDice**», and to expose the catalogs of the products in order to be at the disposal of everybody to facilitate the procedures of purchase which is based on the generation of dynamic prices.

Keywords: e-commerce, e-business, marketplace, online sales.