****

**M. BELAINOUS Anouar**

**M. ABOULFARAH Mohamed**

**Conception et réalisation d'une application de**

**m-commerce sous les plateformes Android et IOS**

**Méthodes informatiques appliquées à la gestion des entreprises**

**OPTION**

**Ingénierie informatique et réseaux**

**FILIERE**

**ANNÉE UNIVERSITAIRE**

**THEME**

**REALISE PAR**

**ENCADREMENT ACADEMIQUE PAR**

**ENCADREMENT ACADEMIQUE PAR**

**2019-2020**

**LOGO ENTREPRISE**

**M. M. BOUSMAH**

**M. M. CHBAB et M. A. RIAHI**

# DEDICACE

A Mes Très Chers Parents

Tous les mots du monde ne sauraient exprimer l’immense amour que je vous porte, ni la profonde gratitude que je vous témoigne pour tous les efforts et les sacrifices que vous n’avez jamais cessé de consentir pour mon instruction et mon bien-être. C’est à travers vos encouragements que j’ai opté pour cette profession, et c’est à travers vos critiques que je me suis réalisé. J’espère avoir répondu aux espoirs que vous avez fondés en moi.

Je vous rends hommage par ce modeste travail en guise de ma reconnaissance éternelle et de mon infini amour. Que Dieu tout puissant vous garde et vous procure santé, bonheur et longue vie pour que vous demeuriez le flambeau illuminant le chemin de vos enfants.

Je dédie aussi ce travail à :

Tous mes amis et spécialement (**Ismail, Ayoub, Hamza, Nihal, Omar),** en souvenir des bons moments que nous avons passés ensemble, pour leur soutien continu, leur aide précieuse et leur amour

**A EMSI**

**Anouar Belainous**

**A tous ceux qui m’aiment**

A mes très chers parents :

Ma mère pour ses sacrifices, son amour inconditionné, ses prières ; et son soutient tout au long de mon cursus scolaire.

Mon père, celui qui m’a indiqué la bonne voie en me rappelant que la volonté fait toujours les grands Hommes.

Merci à vous d’être toujours présents pour faire mon bonheur.

A mes très chers frère et sœur **Ismail** et **Fatima.**

Que vous trouveriez ici, le témoignage d’une fraternité indéfectible, que votre vie soit pleine de succès et de bonheur.

A mes chères amies Pour tous les moments magnifiques et inoubliables que j’ai passés avec vous. Pour tout l’amour et le soutien que vous m’avez offert, je vous remercie très fort, je ne vous oublierai jamais.

A tous mes amis.

**A EMSI**

**Mohamed ABOULFARAH**

**A tous ceux qui m’aiment**

# REMERCIMENTS

**Nous tenons à remercier toutes les personnes qui ont contribué au succès de notre projet de fin d’étude et qui nous ont aidé lors de la rédaction de ce rapport.**

**Dans un premier temps, nous remercions toute l’équipe pédagogique de l’École Marocaine de Sciences de l’Ingénieure et les intervenants professionnels responsables de la formation génie informatique.**

**Nous tenons également à exprimer notre remerciement envers Mme Ouafae AMARA, qui a accepté de nous accueillir au sein de son équipe à SMILE MAROC en nous préparant les conditions favorables au bon déroulement du projet.**

**Nous exprimons aussi notre ample reconnaissance et notre considération envers nos encadrants, Mr Anas RIAHI et Mr Mohamed CHEBAB de SMILE MAROC, qui n’ont ménagé aucun effort pour nous aider et nous orienter tout au long du projet.**

**Également, nous tenons à présenter nos plus vifs remerciements à notre encadrant, Mr**

**BOUSMAH Mohamed, enseignant à l’École Marocaine de de Sciences de l’Ingénieure de**

**Casablanca, pour sa confiance, ses directives précieuses et ses conseils pertinents qui nous**

**ont permis de mener à bien ce projet, et également pour le partage de son savoir et son**

**expérience qui nous ont été très utiles.**

**Que madame, messieurs les membres du jury trouvent ici l’expression de notre reconnaissance pour avoir accepté de juger notre travail.**

**Que tout le corps professoral de SMILE MAROC trouve ici le témoignage de nos profondes reconnaissances pour leur contribution à notre formation.**

**Enfin, nous tenons à remercier nos professeurs de nous avoir incité à travailler en mettant à notre disposition leurs expériences et leurs compétences.**

# **RESUME**

**Le présent document détaille la description et la mise en évidence des différentes phases de notre travail effectué dans le cadre de notre Projet de Fin d’Etudes au sein de l’entreprise SMILE MAROC.**

**L’objectif de ce projet est la mise en place d’une application mobile dédié au commerce afin de faciliter l’opération d’achat à l’ensemble des utilisateurs.**

**Le projet consiste à développer une application mobile en IOS et Android native, développée avec SWIFT et KOTLIN, utilisant les IDE XCODE et Android Studio. Ainsi le langage de requêtes et son environnement d’exécution GraphQl.**

# ABSTRACT

**This document details the description and highlighting of the different phases of our work carried out as part of our End of Studies Project within the company SMILE MAROC.**

**The objective of this project is to set up a mobile application dedicated to commerce in order to facilitate the purchase operation to all users.**

**The project involves developing a mobile application in IOS and native Android, developed with SWIFT and KOTLIN, using IDE XCODE and Android Studio. So, the query language and the GraphQl execution environment.**

# **LISTE DES ABREVEATIONS**

# LISTE DES FIGURES

[Figure 1:Organigramme de l'entreprise SMILE Maroc 16](#_Toc50808002)

[Figure 2:Modèle de cycle de vie en cascade 20](#_Toc50808003)

[Figure 3:Modèle de cycle de vie en V 21](#_Toc50808004)

[Figure 4:Modèle de cycle de vie en Spirale 22](#_Toc50808005)

[Figure 5: Diagramme use case internaute. 25](#_Toc50808006)

[Figure 6:Diagramme use case du client. 26](#_Toc50808007)

[Figure 7:Diagramme de séquence authentification. 27](#_Toc50808008)

[Figure 8:Diagramme de séquence inscription. 28](#_Toc50808009)

[Figure 9:Diagramme de séquence produit. 29](#_Toc50808010)

[Figure 10:Diagramme de classe. 31](#_Toc50808011)

[Figure 11:Diagramme d’activité. 33](#_Toc50808012)

[Figure 12:Diagramme de Gantt 34](#_Toc50808013)

[Figure 13:Architecture MVVM 35](#_Toc50808014)

[Figure 14:Organigramme MVVM. 36](#_Toc50808015)

[Figure 15 : Wireframe - menu 48](#_Toc50808016)

[Figure 16 - Wireframe - Home page 49](#_Toc50808017)

[Figure 17 - Wireframe - Page recherche 50](file:///C:\Users\Anouar\Downloads\Rapport%20PFE%20Anouar%20Belainous%20et%20Mohammed%20Aboulfarah%205%20IIR%20Maarif.docx#_Toc50808018)

[Figure 18 - Pages Contenus 51](#_Toc50808019)

[Figure 19 - Wireframe - Contact 52](file:///C:\Users\Anouar\Downloads\Rapport%20PFE%20Anouar%20Belainous%20et%20Mohammed%20Aboulfarah%205%20IIR%20Maarif.docx#_Toc50808020)

[Figure 20 - Wireframe - magasins 53](#_Toc50808021)

[Figure 21 - wireframe - Catalogue 54](#_Toc50808022)

[Figure 22 - Wireframe - liste des produits 55](file:///C:\Users\Anouar\Downloads\Rapport%20PFE%20Anouar%20Belainous%20et%20Mohammed%20Aboulfarah%205%20IIR%20Maarif.docx#_Toc50808023)

[Figure 23 - Wireframe - Fiche produit 56](file:///C:\Users\Anouar\Downloads\Rapport%20PFE%20Anouar%20Belainous%20et%20Mohammed%20Aboulfarah%205%20IIR%20Maarif.docx#_Toc50808024)

[Figure 24 - Wireframe - Guide checkout 58](file:///C:\Users\Anouar\Downloads\Rapport%20PFE%20Anouar%20Belainous%20et%20Mohammed%20Aboulfarah%205%20IIR%20Maarif.docx#_Toc50808025)

[Figure 25 - Parcours livraison à domicile 59](file:///C:\Users\Anouar\Downloads\Rapport%20PFE%20Anouar%20Belainous%20et%20Mohammed%20Aboulfarah%205%20IIR%20Maarif.docx#_Toc50808026)

[Figure 26 - wireframe – ajout panier - Click & Collect 64](#_Toc50808027)

SOMMAIRE

[DEDICACE 2](#_Toc51690227)

[REMERCIMENTS 4](#_Toc51690228)

[RESUME 5](#_Toc51690229)

[ABSTRACT 5](#_Toc51690230)

[LISTE DES ABREVEATIONS 6](#_Toc51690231)

[LISTE DES FIGURES 7](#_Toc51690232)

[INTRODUCTION GENERALE 12](#_Toc51690233)

[13](#_Toc51690234)

[CHAPITRE 1 : CONTEXTE GENERAL 13](#_Toc51690235)

[I. PRESENTATION DE LA SOCIETE 14](#_Toc51690236)

[INTRODUCTION 14](#_Toc51690237)

[L’organisme d’accueil 14](#_Toc51690238)

[Activité de l’entreprise 14](#_Toc51690239)

[1. Digital 15](#_Toc51690240)

[2. Business Apps 15](#_Toc51690241)

[3. Embedded et IOT 15](#_Toc51690242)

[4. Infrastructure 15](#_Toc51690243)

[Organigramme de SMILE Maroc 16](#_Toc51690244)

[Conclusion 16](#_Toc51690245)

[II. CADRE GENERALE DU PROJET 17](#_Toc51690246)

[Introduction 17](#_Toc51690247)

[Contexte général du projet : 17](#_Toc51690248)

[Problématique 17](#_Toc51690249)

[Solution envisagée 17](#_Toc51690250)

[Étude de l’existant 17](#_Toc51690251)

[Les besoins fonctionnels 17](#_Toc51690252)

[Les besoins non fonctionnels 18](#_Toc51690253)

[III. CONDUITE DE PROJET 19](#_Toc51690254)

[Cycle de vie 19](#_Toc51690255)

[1 Définition 19](#_Toc51690256)

[2 Les activités d’un cycle de vie 19](#_Toc51690257)

[3 Quelques exemples de modèle de cycles de vie 20](#_Toc51690258)

[4 Notre choix : 22](#_Toc51690259)

[23](#_Toc51690260)

[CHAPITRE 2 : ANALYSE ET CONCEPTION 23](#_Toc51690261)

[1. Choix de la méthodologie de conception 24](#_Toc51690262)

[2. Conception du système : 24](#_Toc51690263)

[Diagrammes de cas d’utilisation 24](#_Toc51690264)

[Diagrammes de séquence : 27](#_Toc51690265)

[Diagramme de classes 30](#_Toc51690266)

[Diagramme d’activité 32](#_Toc51690267)

[Diagramme de Gant 34](#_Toc51690268)

[3. L’architecture MVVM 35](#_Toc51690269)

[4. Conclusion 36](#_Toc51690270)

[37](#_Toc51690271)

[CHAPITRE 3 : ASPECTS TECHIQUES 37](#_Toc51690272)

[INTRODUCTOIN 38](#_Toc51690273)

[LES TECHNOLOGIES UTILISEES 38](#_Toc51690274)

[Android 38](#_Toc51690275)

[Gradle 38](#_Toc51690276)

[Apollo 38](#_Toc51690277)

[Node Js 39](#_Toc51690278)

[GraphQL 39](#_Toc51690279)

[Langage Kotlin 39](#_Toc51690280)

[Langage XML 39](#_Toc51690281)

[Matériel design 40](#_Toc51690282)

[Swift 40](#_Toc51690283)

[Outils utilisés 41](#_Toc51690284)

[Android Studio 41](#_Toc51690285)

[XCode 41](#_Toc51690286)

[Visual Paradigm : 41](#_Toc51690287)

[GANTT Project 42](#_Toc51690288)

[Postman 42](#_Toc51690289)

[Outils de gestion de projet 43](#_Toc51690290)

[Gitlab 43](#_Toc51690291)

[Redmine 43](#_Toc51690292)

[Les spécifications fonctionnelles de l’application 44](#_Toc51690293)

[Le périmètre fonctionnel 44](#_Toc51690294)

[Internationalisation 44](#_Toc51690295)

[Devise 44](#_Toc51690296)

[Versions et compatibilités 44](#_Toc51690297)

[ERGONOMIE ET DESIGN 45](#_Toc51690298)

[Gestion de l’application Mobile 45](#_Toc51690299)

[Back end 45](#_Toc51690300)

[API 45](#_Toc51690301)

[Firebase 46](#_Toc51690302)

[47](#_Toc51690303)

[CHAPITRE 4 : TRAVAIL REALISE 47](#_Toc51690304)

[Home page et contenus 48](#_Toc51690305)

[Splashscreen & Guide 48](#_Toc51690306)

[Menu 48](#_Toc51690307)

[3.3 Page d’accueil 49](#_Toc51690308)

[3.4. Recherche 50](#_Toc51690309)

[3.5 Contenus 51](#_Toc51690310)

[3.6 Page Contact 52](#_Toc51690311)

[3.7 Magasins 53](#_Toc51690312)

[4 Produits 54](#_Toc51690313)

[4.1 Catalogue 54](#_Toc51690314)

[4.2 Liste des produits 55](#_Toc51690315)

[4.3 Fiche produit 56](#_Toc51690316)

[4.4 Comparateur 57](#_Toc51690317)

[5 Parcours d’achat 58](#_Toc51690318)

[5.1 Livraison à domicile 59](#_Toc51690319)

[5.1.1 Ajout au panier 59](#_Toc51690320)

[5.1.2 Panier 60](#_Toc51690321)

[5.1.3 Livraison 61](#_Toc51690322)

[5.1.4 Paiement 62](#_Toc51690323)

[5.1.5 Confirmation 63](#_Toc51690324)

[5.2 Click & Collect 64](#_Toc51690325)

[5.2.1 Ajout au panier 64](#_Toc51690326)

[5.2.2 Panier 65](#_Toc51690327)

[5.2.3 Livraison 66](#_Toc51690328)

[5.2.4 Paiement 66](#_Toc51690329)

[5.2.5 Confirmation 66](#_Toc51690330)

[6 Espace client 67](#_Toc51690331)

[6.1 Création de compte 67](#_Toc51690332)

[6.2 Authentification 68](#_Toc51690333)

[6.3 Mot de passe oublié 68](#_Toc51690334)

[6.3 Tableau de bord 69](#_Toc51690335)

[6.4 Information du compte 69](#_Toc51690336)

[6.5 Carnet d’adresse 70](#_Toc51690337)

[6.6 Liste des favoris 71](#_Toc51690338)

[6.7 Liste des commandes 72](#_Toc51690339)

[6.8 Détail de la commande 73](#_Toc51690340)

[**Conclusion général** 74](#_Toc51690341)

[BIBLIOGRAPHIES 75](#_Toc51690342)

[WEBOGRAPHIES 75](#_Toc51690343)

# INTRODUCTION GENERALE

**Des ventes de mains en mains, vers des ventes virtuelles, passent les priorités des opérations de ventes des biens et des services, ce qui nous rend obligés de donner plus d’importance à la vente électronique.**

**Les boutiques en ligne sont depuis des années, largement conseillés pour les sociétés qui se**

**basent sur la vente des produits et même des services Ces types d’application mobile**

**représentent un dispositif global fournissant aux clients un pont de passage à l’ensemble des**

**informations, des produits, et des services à partir d’un portail unique en rapport avec son**

**activité.**

**Les applications de vente en ligne permettent aux clients de profiter d’une foire virtuelle**

**disponible est quotidiennement mise à jour sans la moindre contrainte, ce qui leur permettrai de ne jamais rater les coups de cœur, ainsi Une foire sans problèmes de distance géographique, ni d’horaire de travail ni de disponibilité de transport. D’une autre part ces sites offrent à la société de profiter de cette espace pour exposer ses produits à une plus large base de clientèle.**

**Notre projet est réalisé dans le cadre de stage de projet de fin d’études ayant comme objectif principal : la conception et la réalisation d’une application m-commerce.**

**Le présent rapport va présenter en détails les étapes que nous avons suivie pour réaliser notre application mobile.**

# CHAPITRE 1 : CONTEXTE GENERAL

# PRESENTATION DE LA SOCIETE

## INTRODUCTION

Dans ce chapitre nous allons présenter l’entreprise de stage SMILE Maroc dans trois parties. La première partie est une présentation de l’organisme d’accueil, puis un organigramme qui décrit la hiérarchie au sein de l’entreprise et enfin la fiche technique de l’entreprise.

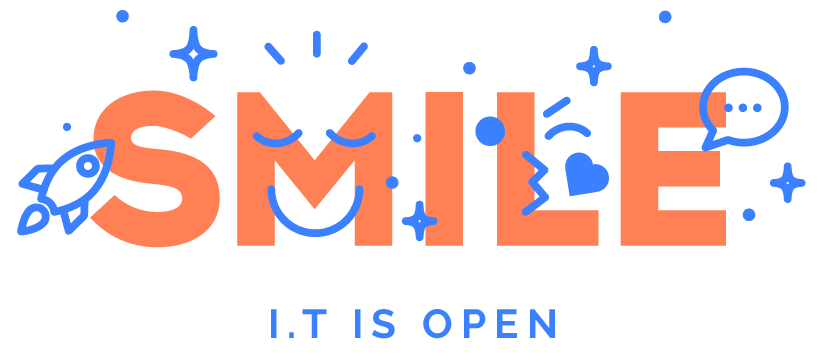
## L’organisme d’accueil

SMILE est une société de services en logiciels libres française spécialisée dans l’intégration de solutions open source. Elle est créée en 1991 et présente en France, en Suisse, aux Pays-Bas, en Belgique, au Luxembourg, au Maroc et en Ukraine. SMILE intervient en conseil, en conception, intégration et réalisation, dans le déploiement de solutions open source, ainsi que la conception et la réalisation de plateformes appuyées sur des produits et outils open source. Son offre couvre aussi la formation, le support et l'hébergement.

Smile Maroc a été la première agence Smile à franchir les frontières françaises, depuis 2004 à Casablanca, c’est la première agence internationale du groupe. Smile Maroc contribue activement à la réalisation des projets digitaux, à la fois sur le marché français et marocain.

Pour répondre au mieux aux besoins des clients, l’agence s'organise autour de deux pôles complémentaires :

* Le pôle Local qui accompagne les grands comptes marocains (Orange, OCP, Bank Al-Maghrib, ONCF, Ministères, Al Omran...) en apportant des solutions open source pérennes, matures et à forte valeur ajoutée.
* Le pôle Skill Center Groupe qui représente plus de 70% du chiffre d’affaires de l’agence grâce à plusieurs CDS (Centres De Service) pour plusieurs grands comptes.



## Activité de l’entreprise

SMILE Maroc a été la première agence SMILE à franchir les frontières françaises, depuis 2004 à Casablanca, c’est la première agence internationale du groupe. SMILE Maroc contribue activement à la réalisation des projets digitaux, à la fois sur les marchés français et marocain.

Agiles, open et engagés, chez SMILE, l’objectif du SMILE est avoir un coup d’avance pour garder sa place sur le podium des acteurs majeurs du digital.

Pour cela, ils ont développé 4 offres pour accompagner ses clients dans leur transformation numérique :

### 1. Digital

* Concevoir des expériences marquantes en couplant UX design & performances.
* Assurer le développement et l’évolution des plateformes de digital commerce, sites responsives design, applications mobiles et solutions collaboratives.
* Piloter, monitorer et optimiser les compagnes de marketing digital pour maximiser la visibilité de dispositifs.

### 2. Business Apps

* Accompagner la création des applications métiers qui servent à gérer les activités de l’entreprise et qui sont conçues selon les besoins métiers exprimés par le client.
* Faciliter l’intégration des solutions basant sur les nouvelles technologies (Java, Angular, API, NoSQL, …)
* Optimiser les coûts de production, gérer et sécuriser les flux de données et automatiser les prises de commandes.

### 3. Embedded et IOT

* Prototyper et concevoir les systèmes embarqués à fortes contraintes temporelles, économiques ou techniques.
* Valoriser les données, maîtriser les solutions de stockage, de traitement et de restitution.
* Développer des nouveaux services en se concentrant sur la valeur ajoutée.

### 4. Infrastructure

* Accompagner l’industrialisation de l’infrastructure.
* Mettre en place l’infrastructure et assurer la configuration et l’optimisation des performances.
* Automatiser le processus de développement et assurer la sécurité des infrastructures.

## Organigramme de SMILE Maroc

Ci-dessous l’organigramme de l’entreprise :

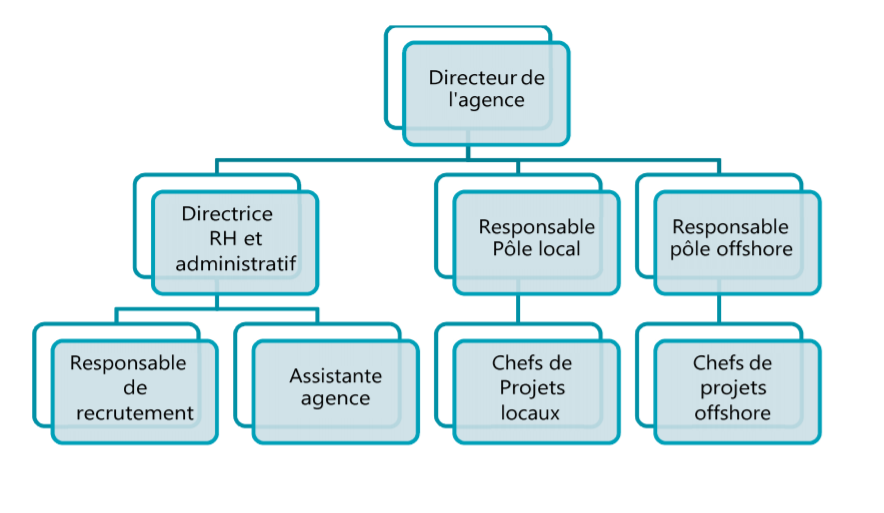


Figure :Organigramme de l'entreprise SMILE Maroc

## Conclusion

Dans ce chapitre nous avons présenté l’organisme d’accueil, ainsi les différentes activités de l’entreprise et enfin une fiche technique.

# CADRE GENERALE DU PROJET

## Introduction

La première étape c’est de consister à mettre le projet dans son cadre général, afin de donner une vision globale de projet. La première partie se focaliser sur le cadre du projet à travers une présentation du contexte général du projet. La deuxième partie, analyse le contexte pour trouver le besoin de l’entreprise en donnant une définition du travail demandé et de la problématique qu’il doit résoudre, ainsi que les besoins fonctionnels et non fonctionnels de l’application conçue.

## Contexte général du projet :

## Problématique

Vu l’importance que prend le mobile dans l’e-commerce, le client Electroplanet souhaite s’adresser à ces clients, via le canal mobile, en offrant des services à valeurs ajoutée,

L’application web déjà mise en place, permet à l’utilisateur d’acheter en ligne. Mais cette application ne peut pas être suffisante de nos jours où le digital a envahi notre quotidien.

et Etant donné cette diversité dans des systèmes d’exploitation et des types des applications mobiles, nous avons dû choisir entre développer une application spécifique à une plateforme ou bien une application multiplateforme.

## Solution envisagée

Pour une entreprise de vente, l’application proposée doit être ergonomique, rapide, riche en fonctionnalités et fiable. Suite à notre étude nous avons constaté que les applications natives offrent les meilleures performances. Et nous allons choisir les systèmes d’exploitation en Android et IOS, utilisant le langage de requêtes et l’environnement d'exécution GraphQL.

## Étude de l’existant

Dans cette section nous allons vous présenter la solution qui existe avant de mettre en place l’application mobile.

Electroplanet présente une application web simple à utiliser, mais il y a le problème de compatibilité avec les interface des téléphones portable, pour cela nous avons créer l’application mobile compatible avec les différents interfaces des téléphones et tablette mobile

## Les besoins fonctionnels

Nous allons vous présenter dans cette partie les actions qu’un acteur peut effectuer à travers l’application mobile mise à sa disposition.

Toutes les fonctionnalités de l’application mobile sont disponibles au client afin de lui faciliter plusieurs opérations.

* Accéder à la page d’accueil.
* Consulter la liste des produits par catégorie ou par marque.
* Consulter la liste des magasins par région.
* S’inscrire ou s’authentifier.
* Consulter la liste des favoris.
* Ajouter des produits au panier.
* Consulter la liste des commandes.
* Choisir le mode de paiement.

## Les besoins non fonctionnels

* **La sécurité**

L’application doit garantir la confidentialité en assurant la validité de l’identité de l’utilisateur. Ceci peut se faire entre autres par moyen d’un mot de passe assurant le contrôle d’accès

* **La facilité**

La solution doit présenter une interface facile et compréhensible. La manipulation de l’interface doit être facile, simple et claire afin de facilité le travail des modérateurs.

* **La rapidité**

L’application doit être rapide afin de permettre au modérateur d’effectuer le maximum de modération par jour.

# CONDUITE DE PROJET

## Cycle de vie

### 1 Définition

Le cycle de vie d'une application comprend toutes les étapes depuis sa conception et sa réalisation jusqu’à sa mise en œuvre. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement du logiciel et la vérification de son processus de développement. L'origine de ce découpage provient du constat que les erreurs ont un coût si élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs le plutôt possible.

### 2 Les activités d’un cycle de vie

Le cycle de vie suivi pour réaliser un site e-commercial, comprend généralement au minima les activités suivantes :

**Spécification des besoins** : elle consiste à définir la finalité du projet et son intégration dans une stratégie globale.

**Conception générale** : dans cette activité, il s'agit de la préparation de l'architecture générale du logiciel.

**Conception détaillée** : elle consiste à définir précisément chaque sous-ensemble du logiciel.

**Développement** : il s’agit d’une traduction des fonctionnalités définies dans la phase de conception en langage de programmation.

**Tests unitaires :** ils permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux normes définies dans la conception.

**Intégration** : dite aussi tests systèmes, elle consiste à vérifier que le logiciel correspond exactement au cahier des charges du projet en obtenant enfin un manuelle d’utilisation bien détaillé aux utilisateurs.

**Validation :** c'est-à-dire la validation de conformité du site avec les buts spécifiés à la première étape du cycle de vie.

### 3 Quelques exemples de modèle de cycles de vie

Modèle de cycle de vie en cascade

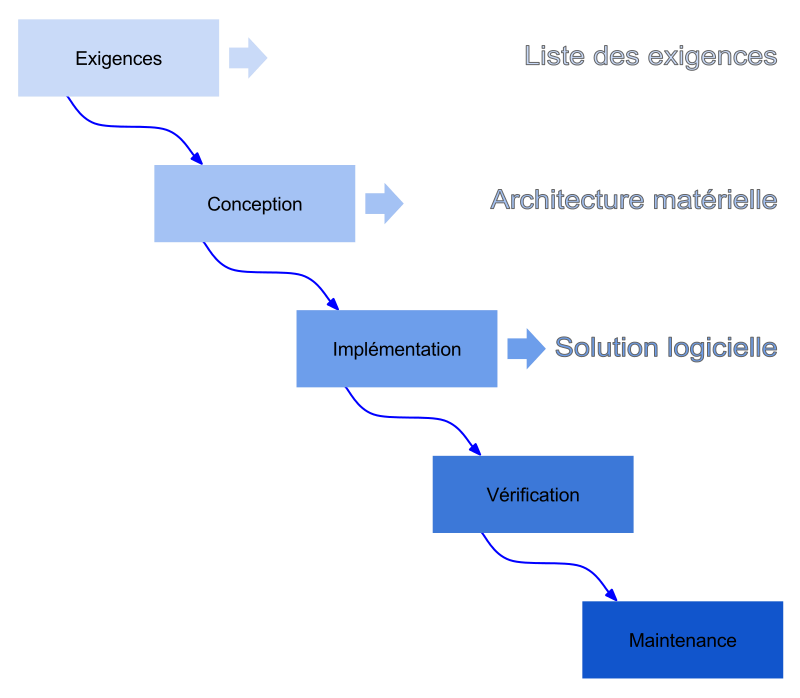


Figure :Modèle de cycle de vie en cascade

Dans ce modèle le principe est très simple : chaque phase se termine à une date précise en produisant certains documents ou logiciels.

Les résultats sont définis à la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés équivalents aux normes. Le modèle original ne comportait pas la possibilité de retour en arrière. Celle-ci a été rajoutée ultérieurement sur la base qu'une étape ne remet en cause que l'étape précédente, ce qui s'avère insuffisant dans la pratique.

Modèle de cycle de vie en V

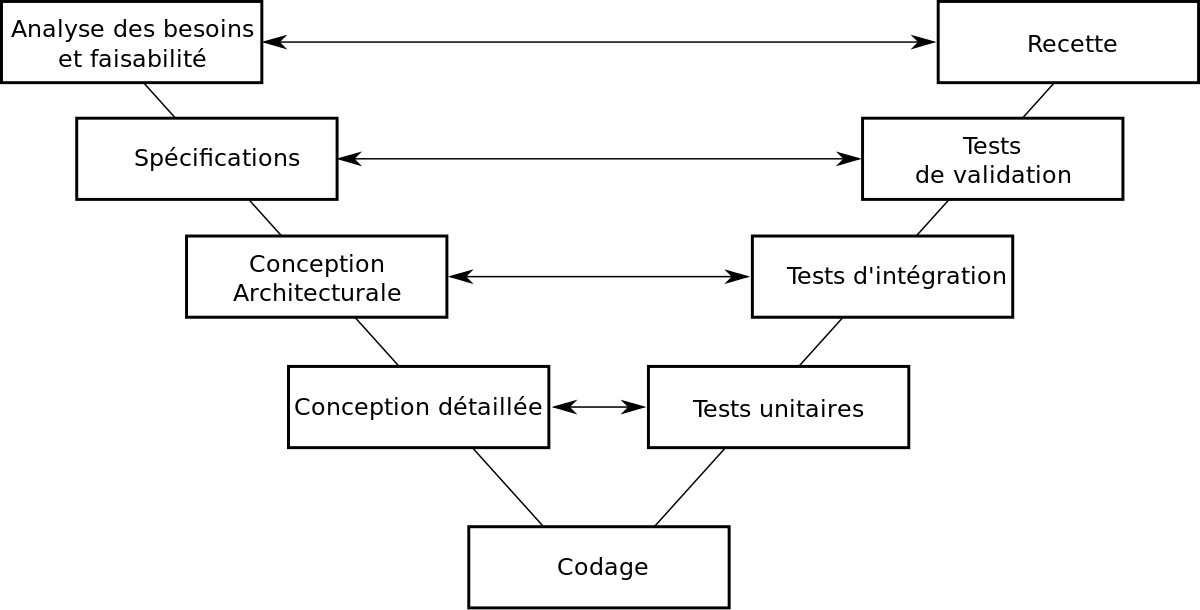


Figure :Modèle de cycle de vie en V

Le modèle du cycle de vie en V est un modèle conceptuel de gestion de projet, imaginé suite au problème de réactivité du modèle en cascade. Il permet, en cas d'anomalie, d’éliminer le retour aux étapes précédentes tardivement.

Les avantages du modèle du cycle de vie en V sont les suivants :

* La qualité de la mise en œuvre des tests.
* Modèle éprouvé dans l’industrie.
* Normalisé (ISO-12207, MILSTD-498…)
* Deux types de tâches sont réalisées en parallèle :

Verticalement on prépare l’étape suivante et Horizontalement : on prépare la vérification de la tâche en cours.

**Ses inconvénients** :

* La validation finale par le client très tardive augmente les risques de dépassement de délai et donc l’augmentation du coût.
* Phases séquentielles.
* Rigidité face à une évolution du besoin.

Modèle de cycle de vie en Spirale

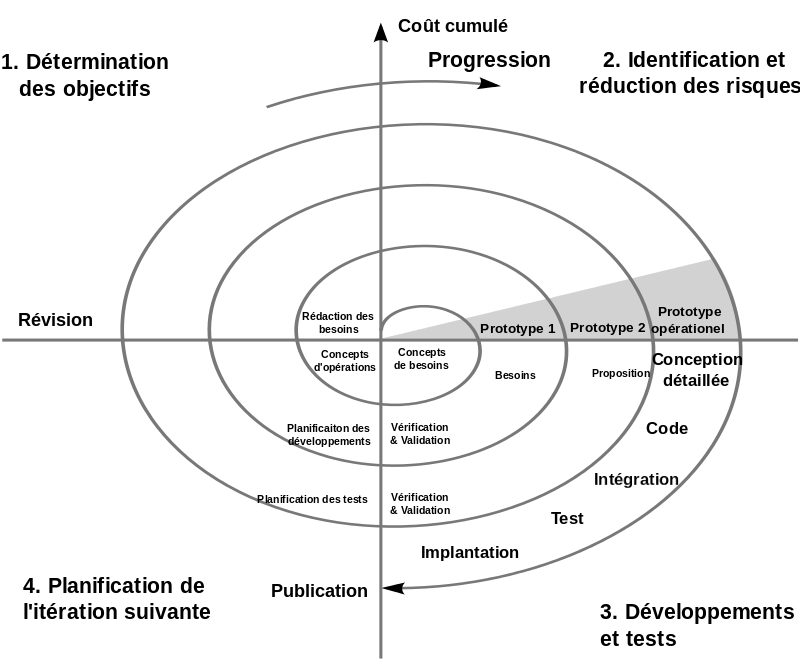


Figure :Modèle de cycle de vie en Spirale

Le modèle en spirale (*spiral modèle*) est un modèle de cycle de vie qui reprend les différentes étapes du cycle en V. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet. Il met cependant plus l'accent sur la gestion des risques que le cycle en V.

## 4 Notre choix :

Afin de concevoir et développer notre application, nous avons opté pour le modèle de cycle de vie en V. Ce choix revient au fait que ce cycle est le plus efficace avec son principe de travail qui nécessite la vérification de chaque étape et la possibilité de corriger les fautes avant de se lancer vers l’étape suivante.

# CHAPITRE 2 : ANALYSE ET CONCEPTION

## Choix de la méthodologie de conception

Dans la cadre de notre projet, nous avons opté pour le langage UML comme une approche de conception. Ci-dessous, nous présentons ce langage puis nous justifions notre choix.



UML (Unified Modeling Language) est un langage formel et normalisé en termes de modélisation objet. Ce Langage est né de la fusion de trois méthodes orientées objet BOOCH, OMT et OOSE, conçues respectivement par Grady Booch, James Rumbaugh et Ivar Jacobson. Son indépendance par rapport aux langages de programmation, aux domaines de l’application et aux processus, son caractère polyvalent et sa souplesse ont fait lui un langage universel. En plus UML est essentiellement un support de communication, qui facilite la représentation et la compréhension de solution objet.

UML ne propose pas de méthode de réalisation. UML est totalement indépendant des langages objet de développement. Une fois la problématique modélisée, une méthode de conduite de projet axée sur la qualité est généralement suffisante pour mener à bien le projet.

Sa notation graphique permet d’exprimer visuellement une solution objet, ce qui facilite la comparaison et l’évaluation des solutions. L’aspect de sa notation, limite l’ambigüité et les incompréhensions. UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues.

## Conception du système :

### Diagrammes de cas d’utilisation

Ce diagramme représente les fonctions du système du point de vue de l’utilisateur. Il permet de modéliser des processus métiers en les découpant en cas d’utilisation.

Il se compose :

D’acteurs : ce sont des entités externes qui utilisent le système à représenter (Operateur, personne…)

Des cas d'utilisations : ce sont des fonctionnalités proposées par le système. Autrement dit c’est l’ensemble des uses cases qui décrivent les objectifs (le but) du système.

#### Des relations de base entre cas d’utilisation et acteurs :

Les inclusions

Les extensions

L’héritage

Les figures suivantes représentent les différents cas d’utilisations de l’acteur internaute qui représente un acteur en mode déconnecté (figure 5) et de l’acteur membre qui représente un acteur en mode connecté (figure 6).

Identification des acteurs de notre application :

Acteur 1 : internaute est un utilisateur de l’application mobile en mode non-connecté. Il a accès seulement aux fonctionnalités qui ne nécessite pas une authentification.

Acteur 2 : Client est un utilisateur de l’application mais en mode connecté. Il a accès à toutes les fonctionnalités et les opérations en mode non-connecté ainsi ce qui sollicitent l’authentification.

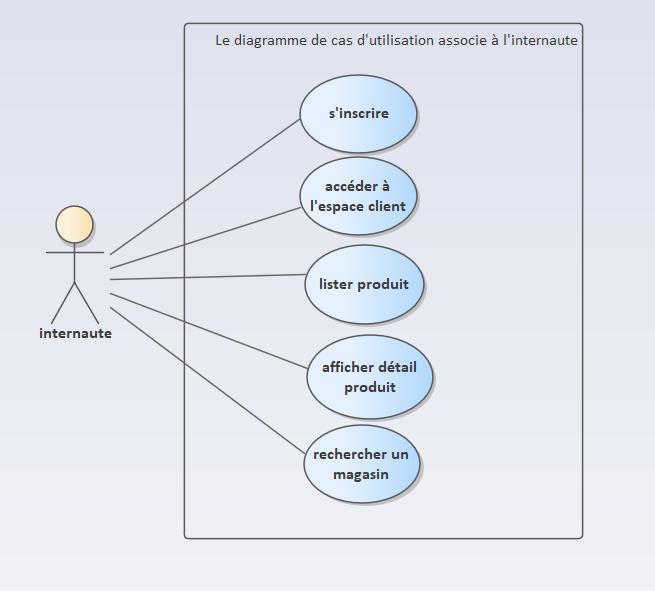


Figure : Diagramme use case internaute.

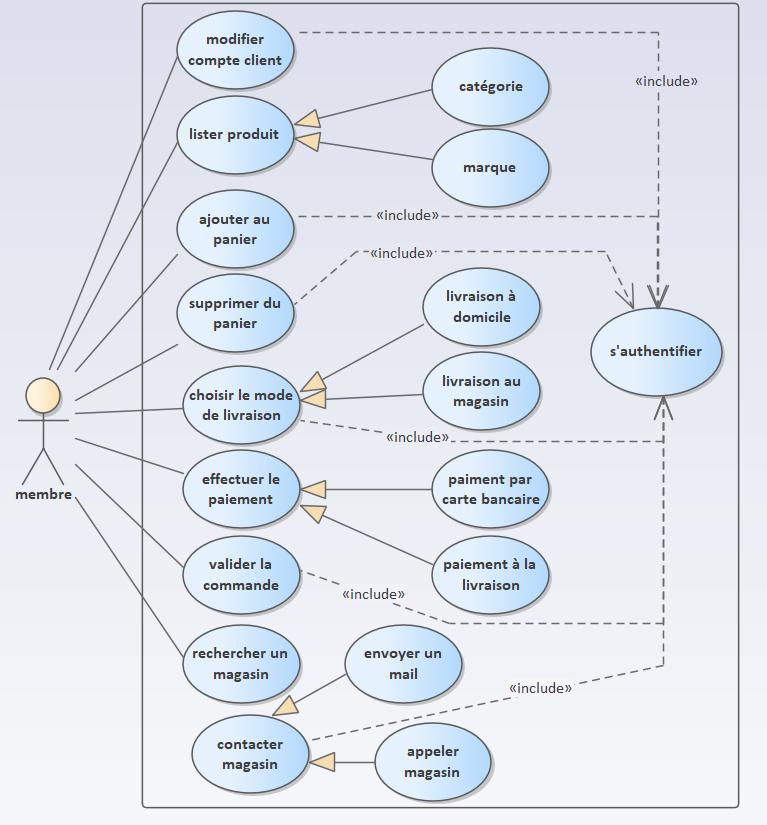


Figure :Diagramme use case du client.

### Diagrammes de séquence :

Scénarios du système

Le diagramme de séquence modélise l’aspect dynamique du système. Il s’agit d’une séquence d’interactions d’un point de vue temporel entre le système et les acteurs.

On va tracer le scénario de quelque cas d’utilisation de notre application :

Le premier diagramme modélise l’aspect de l’authentification

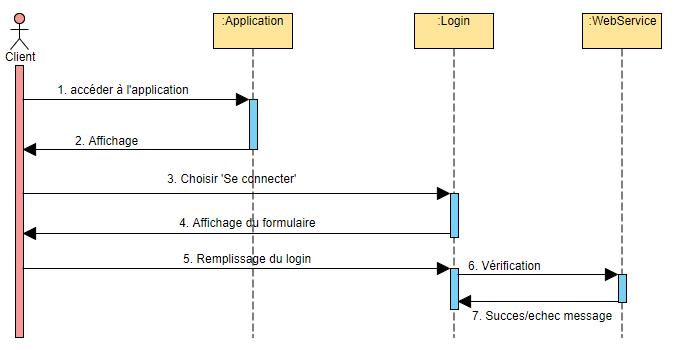


Figure :Diagramme de séquence authentification.

Le deuxième diagramme sert à modéliser l’aspect d’inscription.

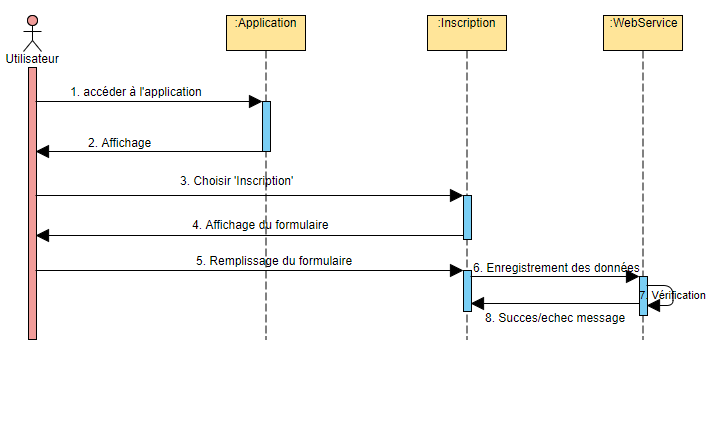


Figure :Diagramme de séquence inscription.

Ce dernier diagramme de séquence sert à nous montrer les étapes d’affichage du détail produit.



Figure :Diagramme de séquence produit.

### Diagramme de classes

Il représente les classes intervenant dans le système. Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations.

Chaque application qui va mettre en œuvre le système sera une instance des différentes classes qui le composent. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. La classe est un concept abstrait qui permet de représenter toutes les entités d'un système.

Dans le diagramme ci-dessous nous avons le diagramme de classe qui nous décrit les différentes classes de notre application.

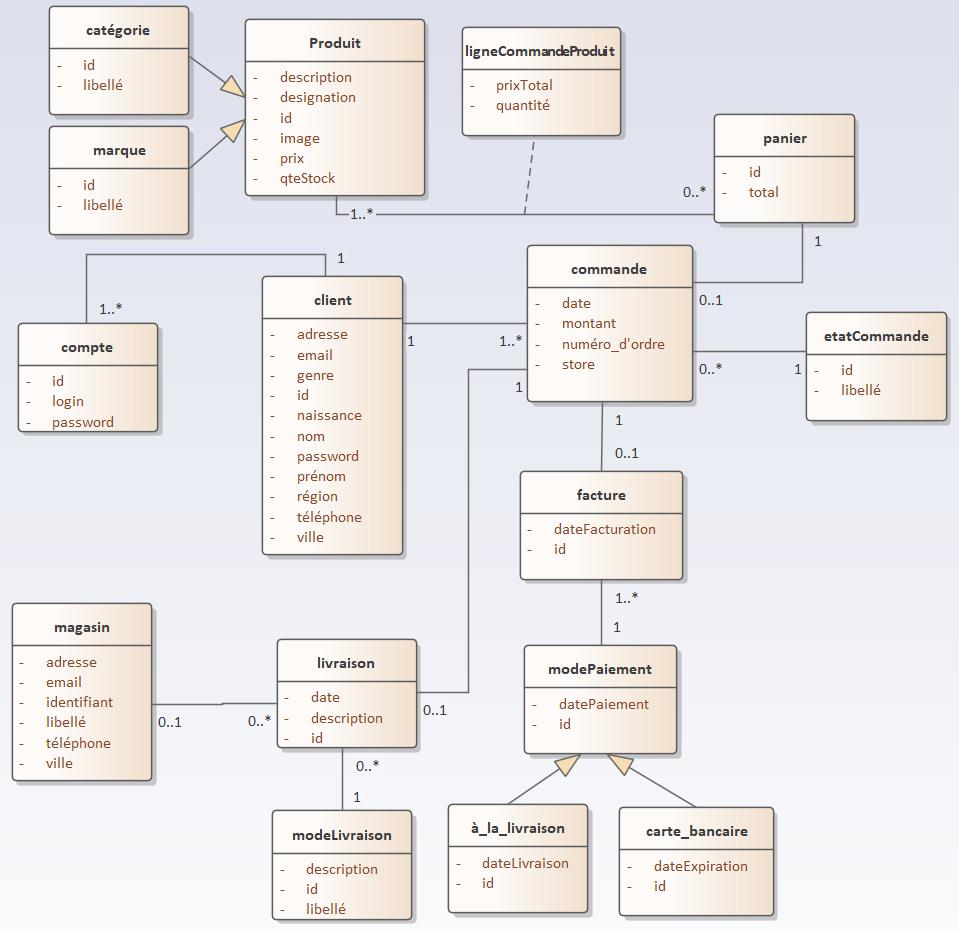


Figure :Diagramme de classe.

### Diagramme d’activité

C’est un Diagramme associé à un objet particulier ou à un ensemble d'objets, qui illustre les flux entre les activités et les actions. Il permet de représenter graphiquement le déroulement d'un cas d'utilisation.

Le diagramme d’activité se compose des éléments suivants :

Une activité représente une exécution d'un mécanisme, autrement dit, un déroulement d'étapes séquentielles. Une transition qui représente Le passage d'une activité vers une autre. Cette transition peut être automatique, qui se déclenche par la fin d'une activité, provoquent le début immédiat d'une autre ou conditionnelle, qui ne se déclenche qu’après la satisfaction de la condition qu’on appelle aussi garde.

Les gardes qui représentent la condition de passage d’une activité à une autre dans les transitions conditionnelles. Ils sont symbolisés par des losanges.

Les barres de synchronisation sont des barres représentées par une ligne épaisse, le rôle cette barre est de synchroniser le départ de plusieurs transitions qui arrivent de déférentes activités, aboutissant toutes à une activité commune.

Ce diagramme d’activité sert à illustrer le fonctionnement détaillé de l’application.

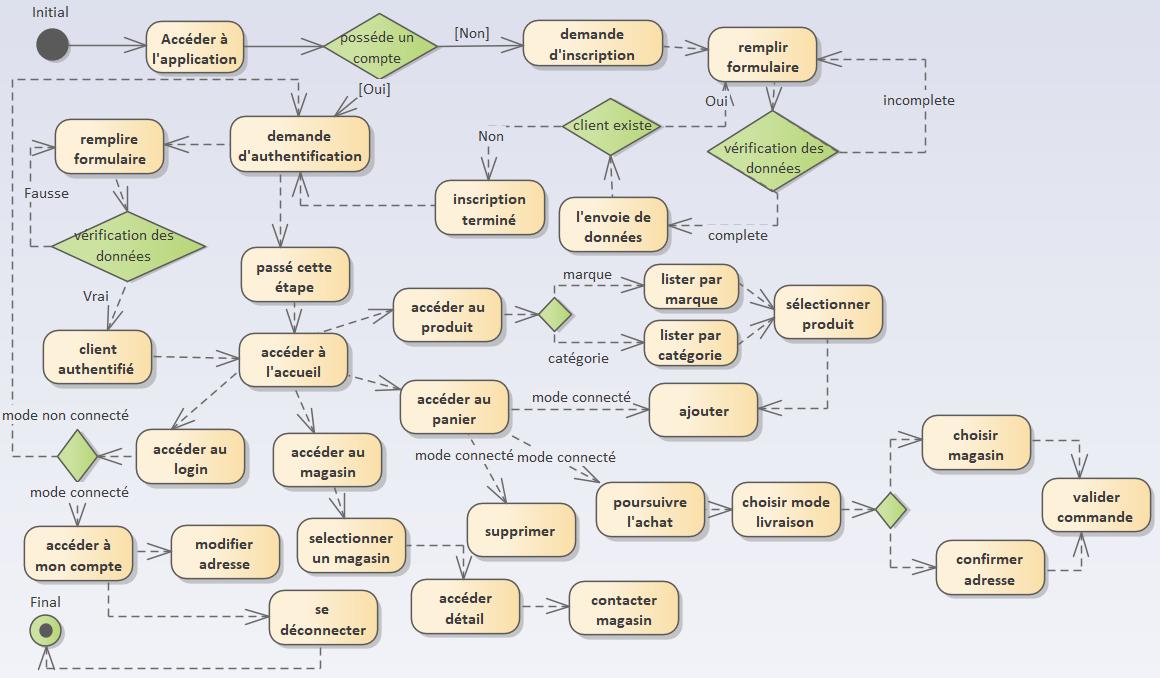


Figure :Diagramme d’activité.

### Diagramme de Gant

La planification d’un projet est considérée comme une étape importante d’avant-projet, elle est la clé de sa réussite et consiste à prévoir le déroulement du projet tout au long de ses phases.

Mon stage du projet de fin d’étude au sein de SMILE Maroc représente une période de travail de 6 mois.

Après la réunion de démarrage, on a pu établir un planning de différentes tâches à réaliser.

Ce planning avait pour objectif de nous aider à respecter le temps qui nous a été accordé dans le cadre de ce stage.

La durée des phases de travail a été estimée objectivement par le volume de chaque partie et, bien entendu, pouvait varier en fonction des imprévus ou des difficultés rencontrés au cours de la réalisation.

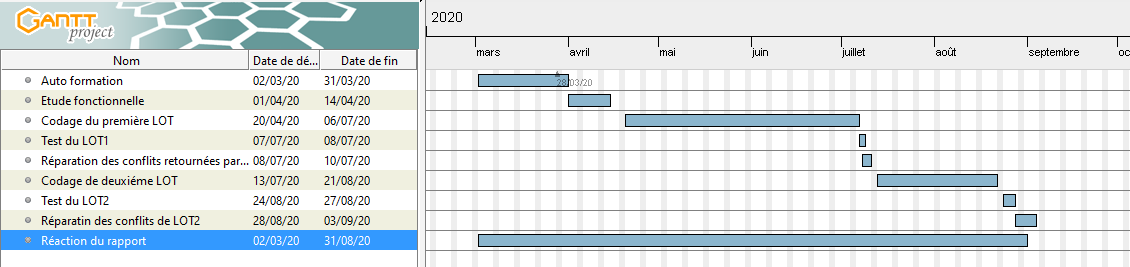


Figure :Diagramme de Gantt

## L’architecture MVVM

L’acronyme MVVM signifie Model View ViewModel, pour certains ce sont des bonnes pratiques lors de la phase développement, pour d’autres ce sont des Framework qui imposent une manière d’organiser son code. Dans les deux cas se servir d’un design pattern permet une bonne maintenabilité et testabilité de l’application développée.

MVVM c'est un modèle architectural pour implémenter des interfaces utilisateur, et fournit à ses utilisateurs une structuration du code respectant de fortes contraintes de développement dans le but de rendre le code plus clair et maintenable.

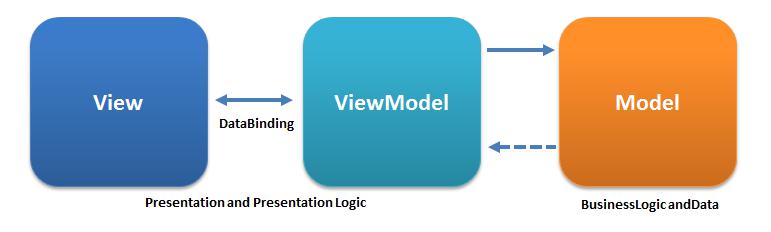


Figure :Architecture MVVM

Les principaux acteurs du modèle MVVM sont :

La vue : l'interface utilisateur que les utilisateurs voient et visualisent envoient et reçoivent les données du modèle de vue

Le ViewModel : il expose des flux de données pertinents pour la vue et fournit les données pour un composant d'interface utilisateur spécifique, tel qu'un fragment ou une activité, et contient une logique métier de traitement des données pour communiquer avec le modèle.

Le DataModel : il résume la source de données. Le ViewModel fonctionne avec le DataModel pour obtenir et enregistrer les données. Il représente les objets métier qui encapsulent les données et le comportement de l'application.

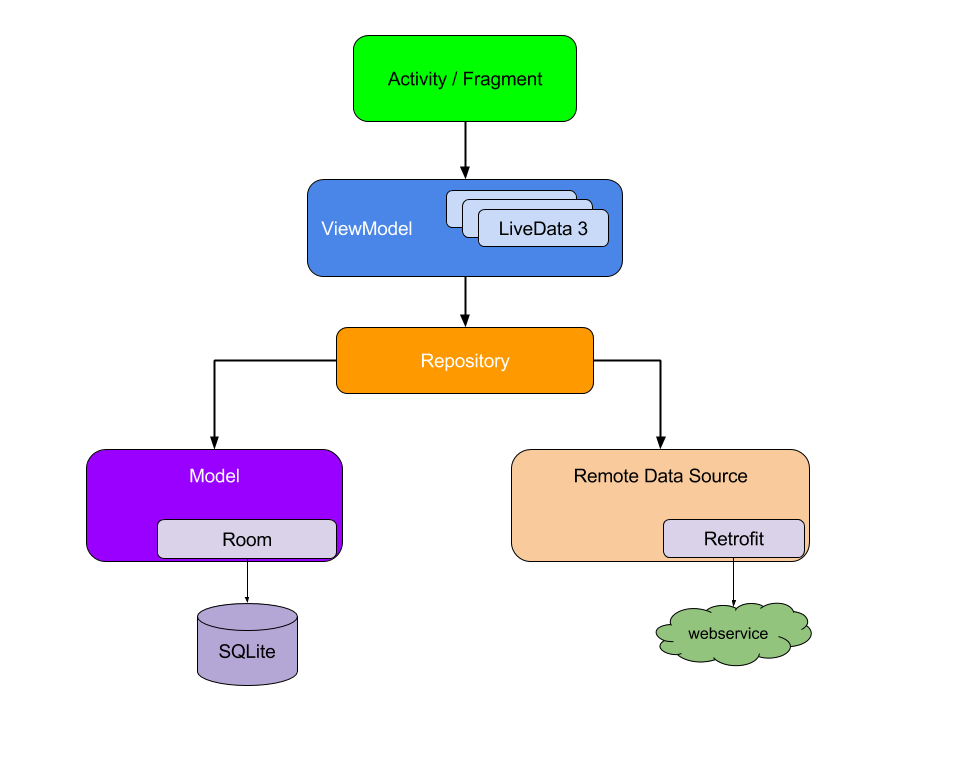


Figure :Organigramme MVVM.

## Conclusion

Ce chapitre à servi à modéliser notre système et à présenter l’architecture suivie avant de de passer à la réalisation pour mieux comprendre ses fonctionnalités.

L’analyse est l’étape la plus importante de la modélisation. Elle permet de modéliser le domaine d'application, et les contraintes de réalisation.

# CHAPITRE 3 : ASPECTS TECHIQUES

## INTRODUCTOIN

Ce chapitre est dédié à la présentation des différents outils de réalisation et du développement de l’application mobile et les choix de technologie.

## LES TECHNOLOGIES UTILISEES

### Android

Android est un système d’exploitation basé sur Linux, ouvert dont le code source est librement accessible ce qui permet à n’importe quel fabricant de l’intégrer dans son système gratuitement.



### Gradle

Gradle est intégré à Android Studio. Il est utilisé pour construire et gérer des projets Android avec le langage Groovy. Il permet, entre autres, de gérer très facilement la construction d’un projet grâce à plusieurs modules et dépendances de librairies Maven.



### Apollo

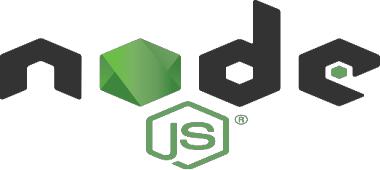
[Apollo](https://www.apollographql.com/) est un Framework qui facilite l’utilisation de [Graphql.](https://graphql.org/) Il est décliné en plusieurs langages, et il existe une version client et serveur.

La force de Apollo Graphql, c’est avant tout la normalisation de votre api, son auto documentation et sa flexibilité dans la résolution des réponses, notamment grâce au système de [schéma](http://graphql.org/learn/schema/) et à des outils de test comme [playground.](https://github.com/graphcool/graphql-playground)

### Node Js

Node.js est une plateforme logicielle libre en JavaScript et orientée vers les applications réseau qui doivent pouvoir monter en charge.

C’est un environnement bas niveau permettant l’exécution de JavaScript côté serveur. Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS.



### GraphQL

GraphQL est un langage de requête et un environnement d'exécution côté serveur pour les [interfaces](https://www.redhat.com/fr/topics/api/what-are-application-programming-interfaces) [de programmation d'application (API)](https://www.redhat.com/fr/topics/api/what-are-application-programming-interfaces) qui s'attache à fournir aux clients uniquement les données qu'ils ont demandées, et rien de plus.

GraphQL est conçu pour mettre à la disposition des développeurs des API rapides, flexibles et faciles à utiliser.

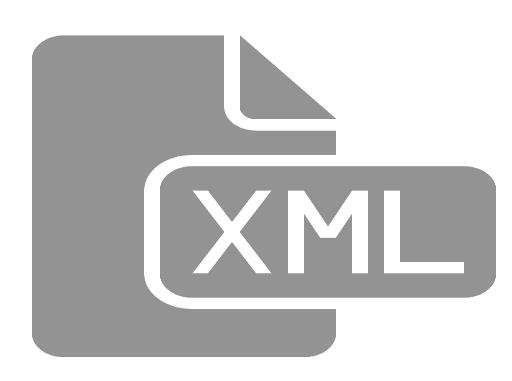


### Langage Kotlin

Kotlin est le nouveau langage de programmation Open Source supporté par Google pour les applications Android (mais aussi pour le backend Java et les moteurs JavaScript des navigateurs web). Il a pour but d’améliorer la productivité des développeurs, tout en restant compatible avec le code existant. Il est également possible de le compiler en natif.



### Langage XML

L'Extensible Markup Langage, généralement appelé XML, « langage de balisage extensible » en français, est un métalangage informatique de balisage générique qui est un sous-ensemble du Standard Generalized Markup langage.

### Matériel design

Le Matériel design est un langage visuel et interactif créé par Google. Mais c’est aussi un guide pour concevoir une interface graphique (Design system).

L’appellation « Matériel Design » correspond à un ensemble de **règles de design** proposées par Google. Il est parfois traduit par « conception matérielle » ou par « design contexturel ». C’était d’abord un design destiné aux applications mobiles (mobile first). Il est ensuite devenu l’une des grandes tendances du design d’interface.

### Swift

Swift est un langage de programmation objet compilé, multiparadigmes, qui se veut simple, performant et sûr. Il est développé en open source.

Le projet de développement de Swift est géré par Apple, qui en est également le principal contributeur ; mais de nombreux membres de la communauté Swift, ainsi que d'autres acteurs, tels que Google et IBM, participent également à son développement. Swift est officiellement supporté sur les systèmes d'exploitation Ubuntu, iOS, MacOs, Watch OS et tv OS



## Outils utilisés

### Android Studio

Android Studio est un nouvel environnement pour développement et programmation entièrement intégré qui a été récemment lancé par Google pour les systèmes Android. Il a été conçu pour fournir un environnement de développement et une alternative à Éclipse qui est l'IDE le plus utilisé.



### XCode

Xcode est un environnement de développement pour MacOs, ainsi que pour iOS, Watch OS et tv OS. L'API Cocoa permet de programmer avec les langages suivants : Objective-C, Ruby et Swift.



### Visual Paradigm :

Visual Paradigm dispose de tous les diagrammes UML et des outils DRE essentiellement dans le système et la conception de base de données. Outils de modélisation innovants comme le catalogue de ressources, transitaire et Nicknamer rend la modélisation du système facile et rentable. Doc. Composer vous permet de produire spécification détaillée prêt à l'emploi en discussion avec seulement quelques clics. Jetez un regard plus profond sur Visual Paradigm, et vous saurez pourquoi nous sommes votre choix.



### GANTT Project

Le diagramme de Gantt est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet. Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin. Ce diagramme permet de visualiser d'un seul coup d'œil :

* Les différentes tâches à envisager
* La date de début et la date de fin de chaque tâche
* La durée escomptée de chaque tâche
* La date de début et la date de fin du projet dans son ensemble

### Postman

Postman est une extension de Google Chrome qui est un navigateur propriétaire fonctionnant sous Windows, Mac, Linux, Android et IOS. Il est conçu pour tester les web services, est gratuit, facile et rapide à manipuler. Il possède également une ergonomie simple et conviviale.



## Outils de gestion de projet

### Gitlab

Gitlab, c’est une plateforme permettant d’héberger et de gérer des projets web de A à Z. Présentée comme la plateforme des développeurs modernes, elle offre la possibilité de gérer ses dépôts Git et ainsi de mieux appréhender la gestion des versions de vos codes sources. Initialement connu pour sa capacité de gestion de versions des codes sources, Gitlab s’est développé au cours des dernières années pour devenir aujourd’hui un outil incontournable de gestion de projet web.



### Redmine

Redmine est une application open source de [gestion de projet en ligne.](http://www.prodmine.com/logiciel-gestion-de-projet-en-ligne/) Développé par Jean-Philippe Lang, l’outil fonctionne avec la plateforme Ruby on Rails.

Redmine offre plus d’avantages par rapport aux autres outils de gestion de projet en ligne. En effet, Redmine est l’outil par excellence pour la gestion et la conduite d’un projet voire de plusieurs projets simultanément. Sa première particularité est sa simplicité d’utilisation tout en étant un outil complet et abouti.



## Les spécifications fonctionnelles de l’application

### Le périmètre fonctionnel

Les fonctionnalités ont été évoquées dans le détail avec les équipes de Electroplanet et la DSI de Marjane lors des ateliers de conception, validation de la charge, et choix de parcours et solutions technique.

La conception s’appuie sur le site electroplanet existant, ainsi que les évolutions évoquées par Electroplanet lors des ateliers de cadrage.

Toutes les fonctionnalités sont listées sur ce document. Les fonctionnalités non listées dans ce présent document sont inexistantes dans le périmètre du projet.

### Internationalisation

La langue utilisée dans l’application mobile est le français.

Aucune autre langue n’est prévue dans la première version.

### Devise

La devise utilisée sur l’application est celle retournée par le site web, aucune modification à prévoir dans l’application en indépendance du site.

Les prix des produits affichés sont ceux retournés par les APIs concernés.

### Versions et compatibilités

Les versions supportées par l’application mobiles seront :

*IOS :* supérieur à la version 10

*Android :* à partir de la version 6.0 (Marshmallow)

Une seule application sera développée par système d’exploitation, sur tablette et IPad, l’affichage s’adaptera avec la largeur.

### ERGONOMIE ET DESIGN

Le design s’appuiera sur les maquettes validées par EP et la DSI Marjane.

Dans ce document, les wireframes validées font référence à l’affichage. Dans le cas de conflit entre texte et image c’est le texte qui fait foi.

## Gestion de l’application Mobile

### Back end

Le back end de l’application mobile d’Electroplanet sera représenté complètement par le backoffice du site Web e-commerce d’Electroplanet, qui est le Backoffice Magento.

Toutes les données affichées dans l’application seront récupérées depuis le site e-commerce de la boutique en ligne.

Les libellés d’affichage et des formulaires seront par contre codés indépendamment dans l’application.

Aucune adaptation n’est prévue dans le back end pour modifier le design (forme / couleur) de l’application mobile. Seules les données seront contribuables, couleurs et formes d’affichages seront fixés selon les maquettes validées.

### API

L’application mobile sera basée sur l’API de la boutique en ligne d’EP, l’API mise en place pour l’application mobile sera développée en GrapheQL.

Le développement de l’application mobile va s’appuyer sur la version 1.6 du contrat d’interface partagé par la DSI Marjane et validée par Smile.

L’API implémentée devra respecter le contrat d’interface, vu que les développements du côté de l’application mobile vont commencer avant la livraison de l’API.

L’application mobile va s’appuyer sur l’API pour récupérer les données à afficher à savoir contenu, produits, catégories, données du clients et magasin ainsi que pour synchroniser les informations clients avec le serveur à savoir modification d’information de compte, ajout ou suppression d’adresse, création de commande, et bien évidemment de contacter le service client via le formulaire de contact.

### Firebase

#### Analytics

Firebase offre une interface pour afficher l’Analytics de l’application mobile.

Seules les fonctionnalités de base de Firebase seront utilisées aucune adaptation n’est prévu dans le cadre de cette version.

Les évènements qui seront écoutés sont :

* Visualiser une page catégorie
* Visualiser une page de produit
* Ajout au panier
* Abandon panier
* Choix du parcours d’achat
* Choix de livraison
* Choix du mode de paiement
* Confirmation de la commande

#### Notifications

Deux types de push sont à prévoir dans l’application d’EP :

* Notifications promotionnelles : envoie d’une même notification à un ensemble d’utilisateurs
* Notifications personnalisées : envoie d’une notification à un utilisateur spécifique, suite à un évènement spécifique : changement de statut de la commande

Les notifications seront gérées par Magento, qui devrait appeler Firebase en envoyant le message, l’image et les tokens Firebase qui seront notifiées.

Le contrat d’interface de l’API de Firebase sera partagé avec la DSI Marjane.

# CHAPITRE 4 : TRAVAIL REALISE

# Home page et contenus

Ce chapitre va représenter les parties contenues de l’application, à savoir :

* Splashscreen & guide d’utilisation
* Menu
* Page d’accueil
* Recherche
* Contenus
* Contact
* Magasins

## Splashscreen & Guide

A l’accès à l’application, afin d’occuper le temps de chargement, un splashscreen sera affiché.

Au premier accès à l’application un guide, présentant les moyens d’exploitation de l’application sera proposé, l’utilisateur aura la main de passer.

## Menu

Un menu en bas de page sera affiché en permanence afin de faciliter la navigation entre les pages les plus importantes.



Figure : Wireframe - menu

Le menu sera composé de cinq éléments à savoir :

• Accueil : Home page de l’application

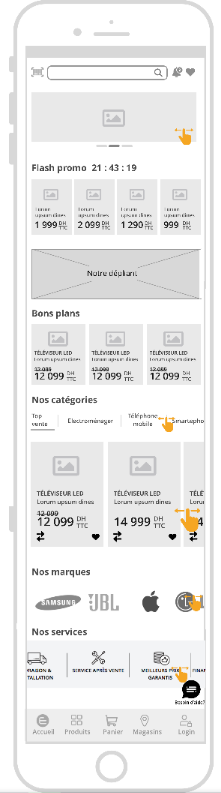
• Produits : Le catalogue

• Panier : Le panier du client

• Magasin : Liste des magasins

• Login / compte : Page d’authentification ou l’espace client selon si le client est connecté ou pas

## 3.3 Page d’accueil

****

La page d’accueil sera représentée comme le montre l’interface à gauche.

**Au top de la page :**

* **La barre de recherche**  : Au clique sur la barre de recherche, la vue recherche sera affiché ( voir le paragraphe lié à la recherche )
* **Favoris :** le nombre de produits favoris sera affiché sur l’icône si le client à des produits favoris.  
  Au clique sur l’icône, si le client est connecté la liste des favoris sera affichée, sinon la page d’authentification sera affiché avec un message expliquant que, uniquement les clients connectées, peuvent avoir une liste de favoris.

*Aucune icône notifications est à prévoir dans l’application, les notifications sont géré par l’OS*

**En premier plan, des données promotionnelles seront affichées, à savoir :**

* **Le slider** : les compostants du slider sont complètement contribuables depuis le backoffice Magento :
  + Nombre des slides affichées
  + Image de chaque slide
  + Texte affiché
  + Navigation : au clique sur le slide en question, le client peut être redirigé vers une page produit, une page catégorie ou une page CMS affichée en webview dans l’application.

*Les sliders seront animés automatiquement, comme le client peut passer au slide suivant en utilisant le swipe.*

* **Promotion Flash :** Une liste de produits en promotion sera affiché. Le client peut voir la suite en utilisant le swipe. un nombre spécifique de produit en promotion sera affiché ( 16 seulement ).

*Aucun compte à rebours ne sera affiché dans la première version de l’application cette information n’étant pas disponible dans le site.*

* **Dépliant :** Une image mettant en avant le dépliant d’Electroplanet peut être affichée. Au clique, une webview affichant le dépliant d’electroplanet déjà chargé dans le site.
* Figure - Wireframe - Home page

**Un accès direct est prévu pour quelques pages clés :**

* **Nos catégories :** affichant la liste des catégories en onglets, chaque onglet affichera une liste de 16 produit de la catégories en question. Les catégories à afficher dans cette partie doivent être configurés dans le Backoffice Magento.

**Nos marques :** une liste de marque à mettre en avant seront affichés, le client pourra naviguer via swipe au clique dans chaque marque, la page de la catégorie filtré par marque sera affichée

* **Nos services :** Le bloc nos services ne sera pas affiché dans la page d’accueil, mais comme lien dans la page de l’espace client.

## 3.4. Recherche

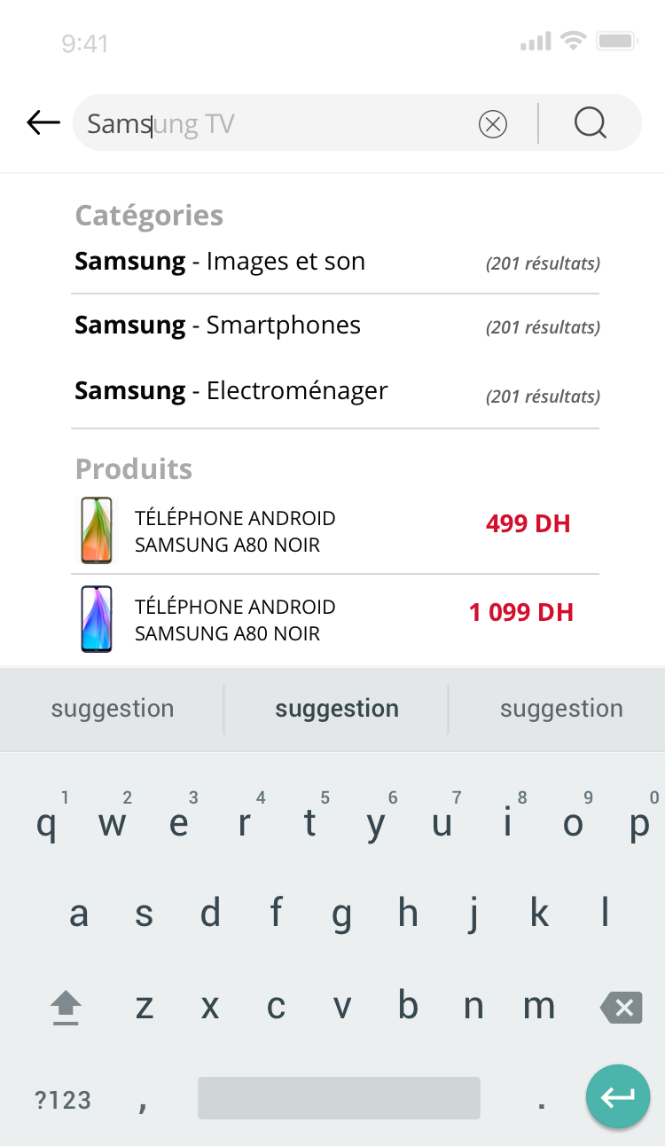
Le clique sur la recherche ouvre une nouvelle page, la page de recherche affiche :

Figure - Wireframe - Page recherche

* Une zone texte de recherche, un bouton pour vider la recherche,
* Un historique des recherches effectués par le client.

*Les recherches populaires ne seront pas affiché dans ce périmètre, aucune API n’a été prévu pour ce point.*

*La recherche vocal n’est pas prévu dans ce périmètre*

*Aucune recherche par code à barre ou par image n’est prévu dans ce périmètre*

Les résultats de recherche s’affiche en auto-complétion. L’auto complétion dépend du retour de l’API de recherche.

Le résultat en auto complète sera affiché sur deux parties :

• Catégorie

• Produits

Le clique sur la catégorie ou le produit, redirigera vers la page en question.

## 3.5 Contenus

Les pages de contenus seront affichées en webview dans l’application.

Sauf pour les pages accessibles depuis le slider de base, les autres pages sont accessible depuis la page authentification, ou l’espace client.



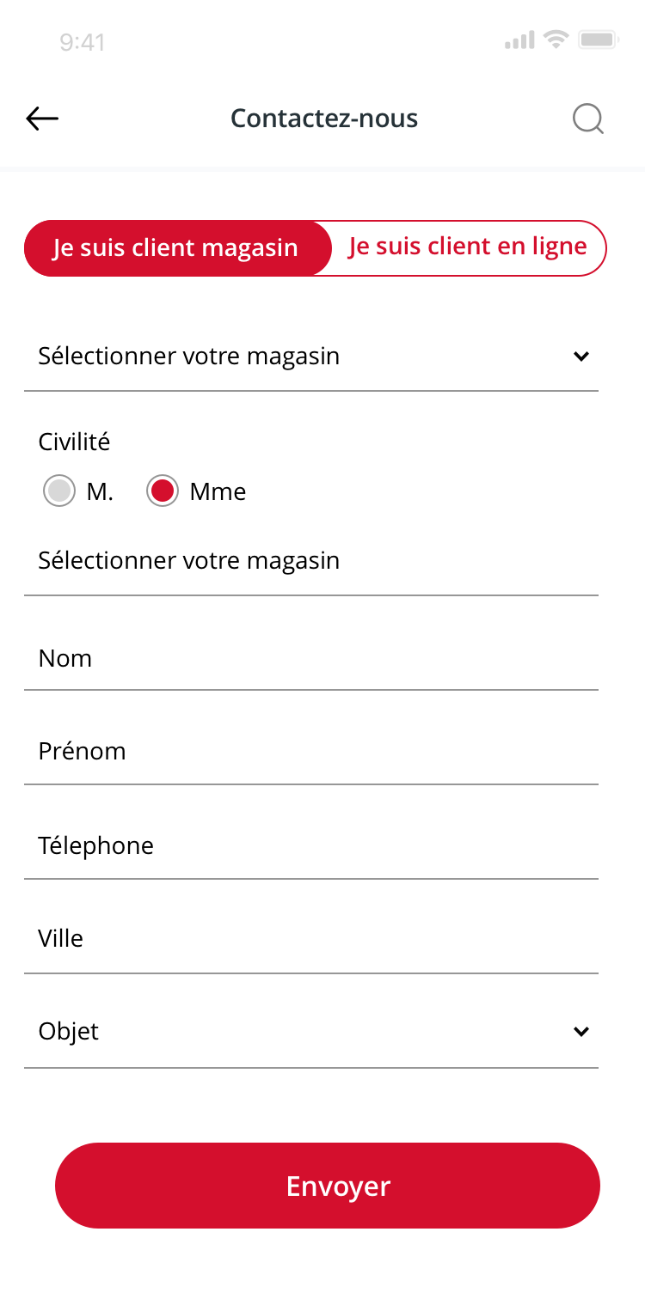
Le bloc encadré en rouge regroupe les liens vers les pages de contenus à savoir :

* Conditions générales de ventes
* Conditions générales d’utilisations
* Services EP
* A propos d’EP
* FAQ

Les pages / lien affichées dans ce bloc doivent être fournis par EP avant le début des développement.

Figure - Pages Contenus

## 3.6 Page Contact



Le formulaire de contact est affiché avec un switcher selon si le client veut contacter le magasin ou le site.

Le formulaire devrait être différent selon le choix.

Les champs du formulaires seront repris du site web en cohérence avec l’API ( contrat d’interface )

Le formulaire de contact est accessible depuis :

* L’icône de contact affiché flottant dans toutes les pages
* Le lien contact dans le bloc de l’espace client/ authentification

Figure - Wireframe - Contact

## 3.7 Magasins

La liste des magasins est aussi accessible depuis le menu. Les vues prévues pour les magasins sont les suivantes :



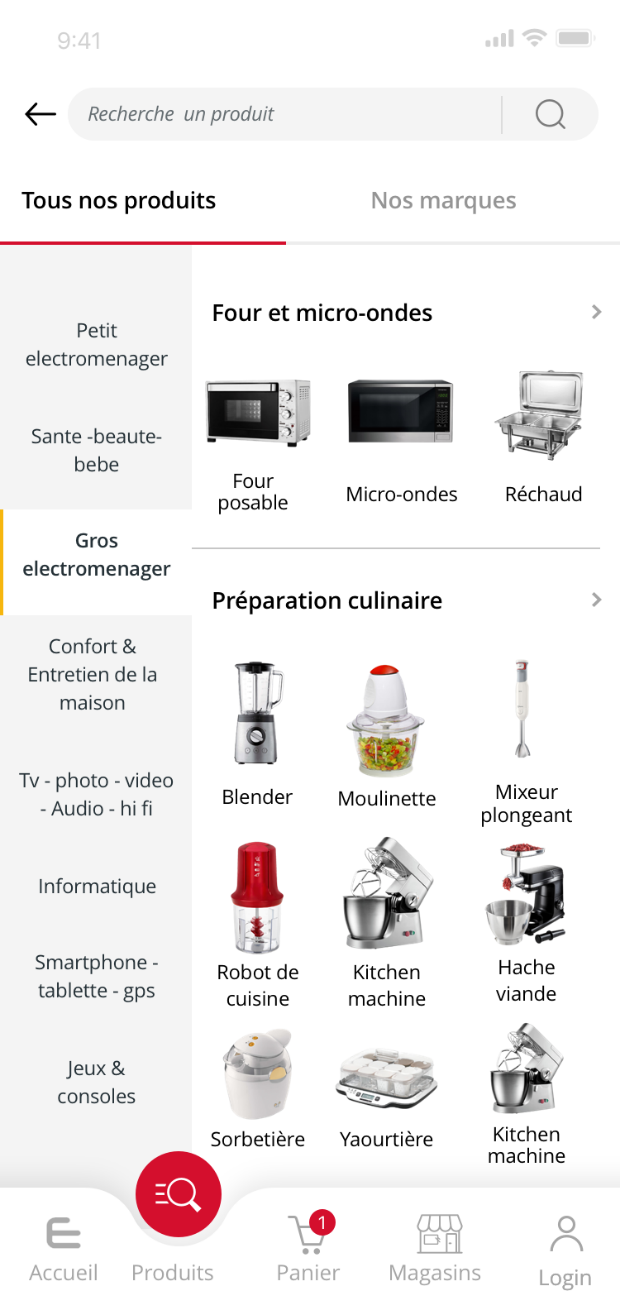
Figure - Wireframe - magasins

La liste des magasin est affiché, avec la possibilité de voir l’emplacement dans la carte ( natif OS ) , ou de chercher par ville via dropdown., A la selection d’un magasin, une fiche magasine est affichée, mettant en avant :

* Image
* Nom
* Horaires
* Descriptif
* Bouton pour contact / appel

# 4 Produits

## 4.1 Catalogue



L’affichage validé pour le catalogue est représenté dans le wireframe à gauche, le design sera validé avec les maquettes. Le catalogue sera affiché sur deux onglets :

* Le premier pour tous les produits, en les regroupant par catégories / sous catégories
* Le deuxième pour les marques , en affichant les marques en vedettes au top, puis la liste de toutes les marques. Au clique dans chaque marque la page catégorie filtré par la marque sera affichée.

Cette page est accessible en cliquant sur le menu produits

Figure - wireframe - Catalogue

## 4.2 Liste des produits

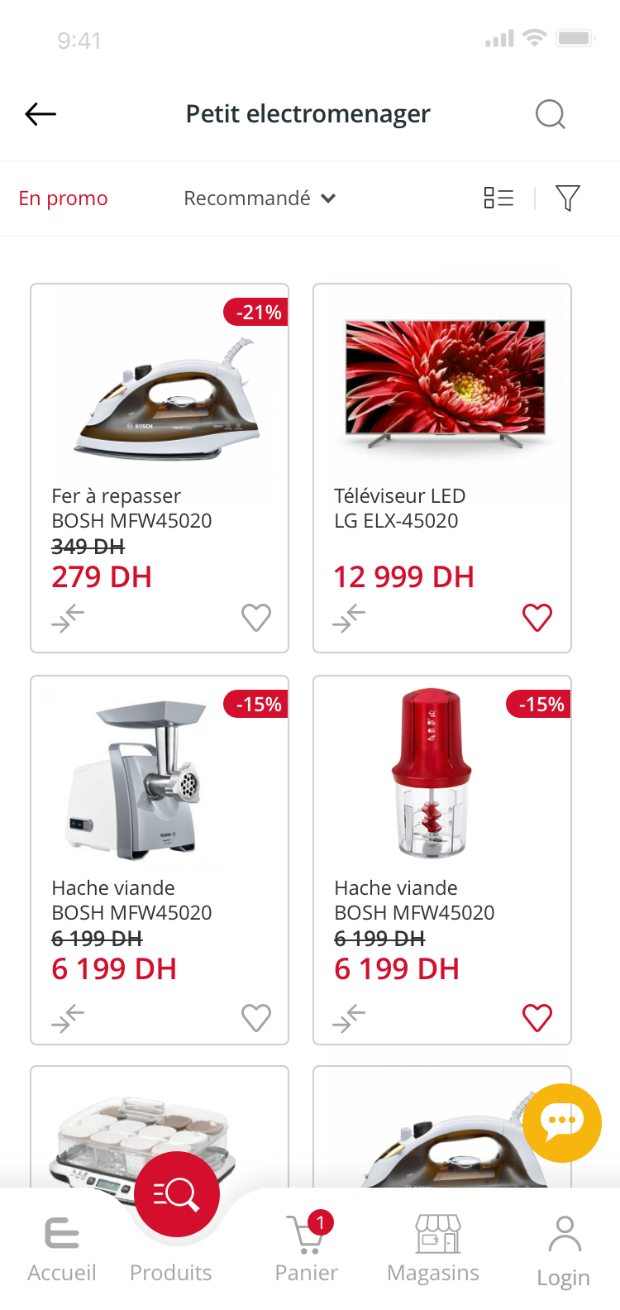


Figure - Wireframe - liste des produits

La liste des produits est affichée comme dans le wireframe avec les éléments suivants :

* Zone de recherche pour accès à la recherche
* Titre de la catégorie en question
* Possibilité de retour en arrière
* Une barre pour gérer :
  + Filtres
  + Ordre d’affichage
  + Possibilité de switcher à l’affichage liste/grille

Filtres et ordre seront affiché comme récupéré depuis le site web.

* Les produits : chaque élément sera représenté par :
  + Image
  + Prix
  + Lien d’ajout au Comparateur
  + Lien d’ajout à la liste des favoris

Les produits seront affiché en infinite scroll.

## 4.3 Fiche produit



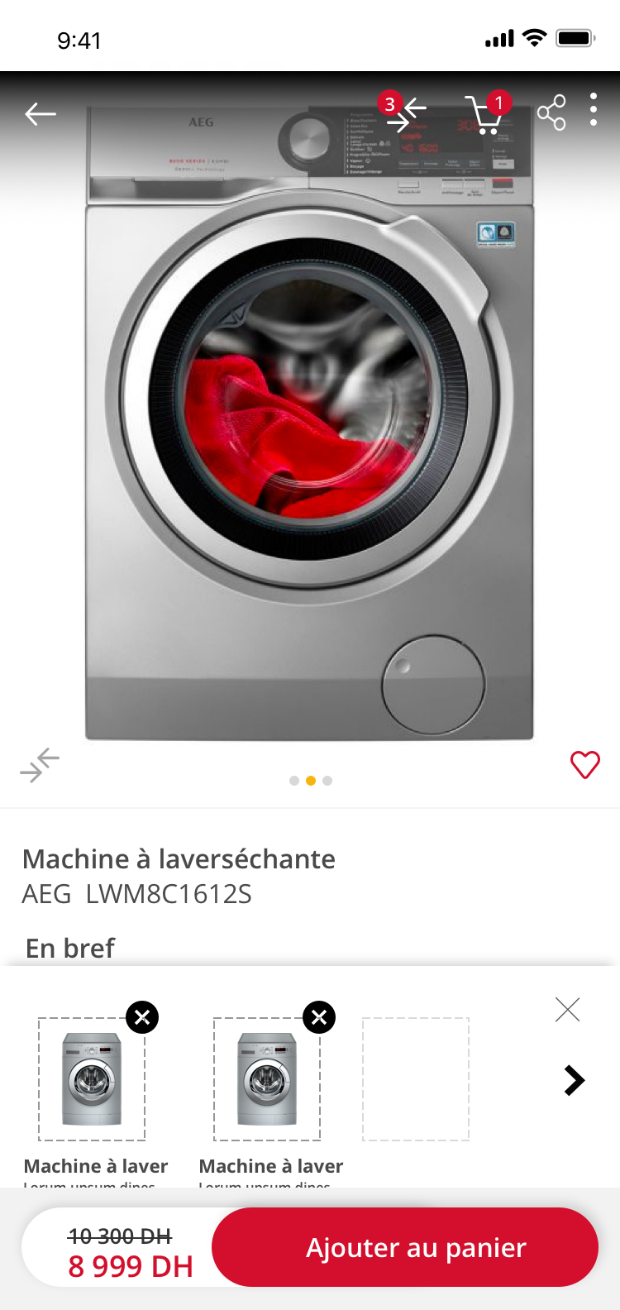
Figure - Wireframe - Fiche produit

La fiche produit sera affichée comme suit avec :

* Une barre top contenant :
  + Retour en arrière
  + Panier
  + Partage réseaux sociaux
* Image produit
* Lien d’ajout au favoris
* Lien d’ajout au comparateur
* Titre du produit
* Prix
* Descriptif du produit ( liste d’attribut et leur valeurs )
* La description sera affiché en cliquant sur un lien en webview, la page a été mise en place dans le site web
* Bouton D’ajout au panier

*Les avis produits ne sont pas inclus dans la première version du projet.*

## 4.4 Comparateur



L’ajout d’un produit au comparateur se fera selon les règles de gestion prédéfini dans la boutique en ligne.

* La limite des produits dans le comparateur est de trois.
* On ne peut pas ajouter des produits de familles différentes au comparateur.

Ces règles sont gérées complètement du côté de l’API, aucune gestion des produits à ajouter dans le comparateur n’est géré dans l’application.

Une modification de ces règles de gestion dans le site génèrera forcément une modification de ces règles dans l’application.

Les attributs affichés dans le comparateur par famille produit sont récupérés depuis l’API.

# 5 Parcours d’achat

Deux parcours d’achat sont possibles dans l’application EP :

• Livraison à domicile

• Click & collect

Figure - Wireframe - Guide checkout

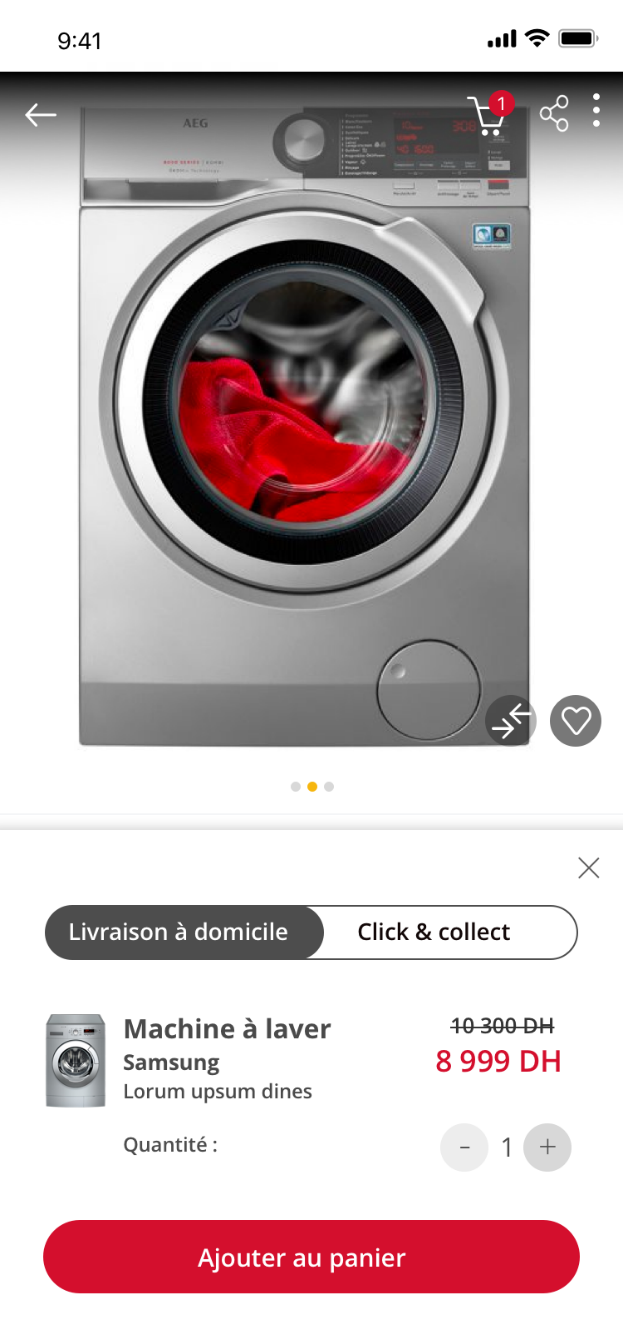
Le client fait son choix à l’ajout du premier produit au panier. Une fois le produit ajouté, le client choisit son parcours. Les produits suivants à ajouter dans le panier seront dans le même parcours ; le cas contraire, le client devrait vider son panier et ajouter le nouveau produit.

Durant tous le parcours, à chaque étape, un appel vers l’API est effectué, pour récupérer les information du panier ou client, ou enregistrer ces informations.

Une démonstration explicatif du parcours sera affiché au premier accès à une page produit, comme le montre le guide suivant :

## 5.1 Livraison à domicile

### 5.1.1 Ajout au panier



La livraison à domicile est le choix par défaut au clique sur l’ajout au panier.

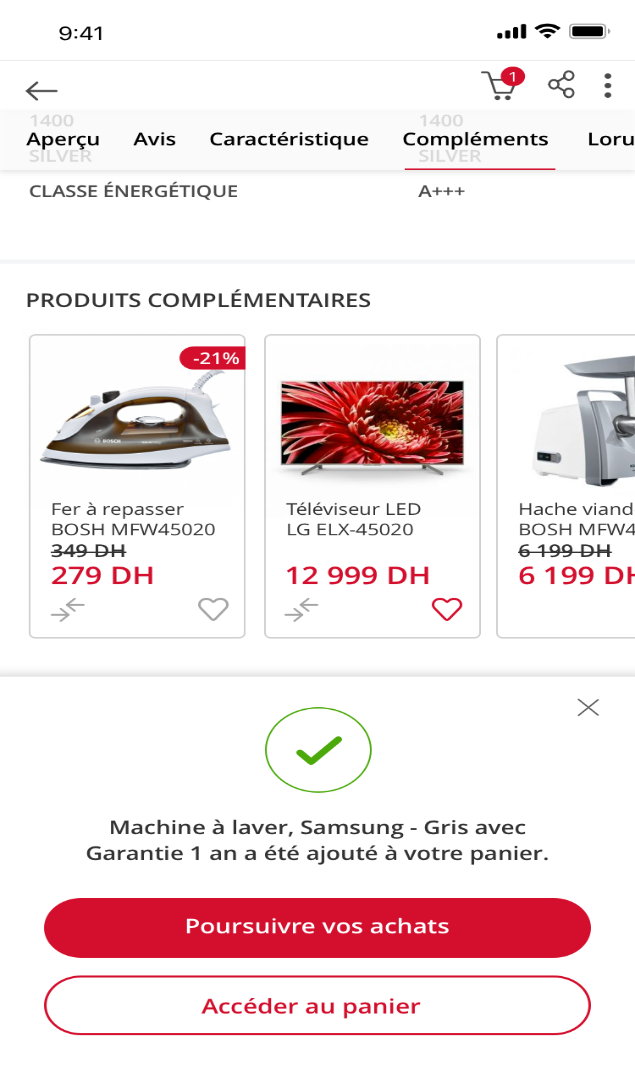
Si le client choisit de se livrer à domicile, il devrait choisir la quantité, aucun choix de couleur n’est prévu dans la première version de l’application.

Une fois ajouté au panier, le client aura le choix entre

• Au checkout pour finaliser sa commande

• Continuer ses achat.

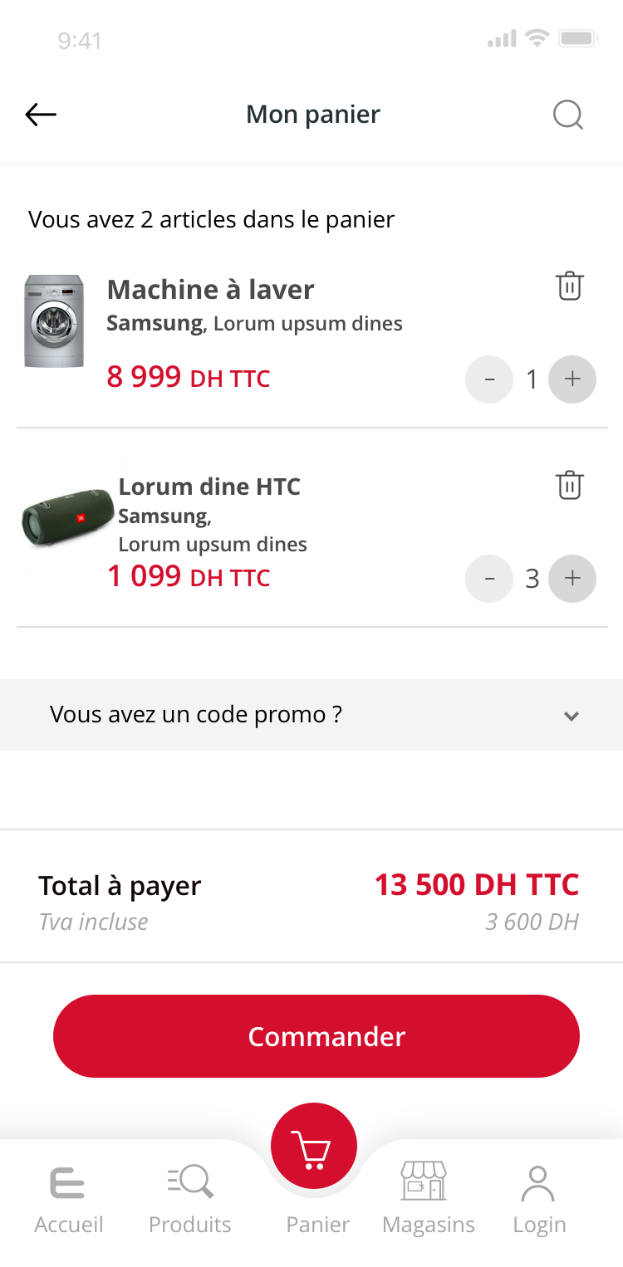
Figure - Parcours livraison à domicile



### 

Figure 26 - Panier choix parcours

### 5.1.2 Panier

Pour ce parcours, aucun magasin ne sera affiché. Le client verra ses produits, ainsi que le total de son panier en TTC avec le détail des taxes, ainsi que la remise.

Le client pourra :

• Modifier la quantité par produit

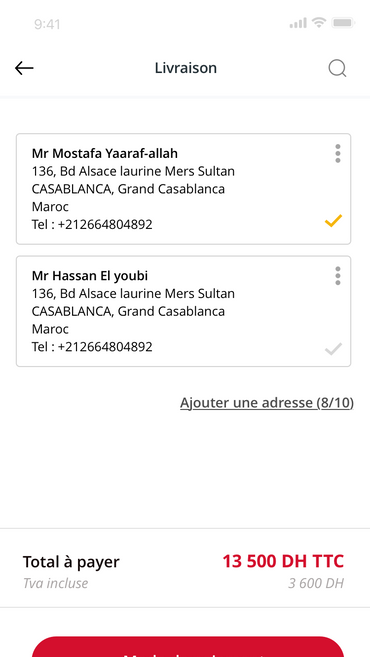
• Supprimer un produit du panier

• Vider son panier

Le client devra approuver les CGU et CGV avant de passer à l’achat.

Figure 27 - Panier liste des produits panier

### 5.1.3 Livraison



Le choix de l’adresse de livraison est la première étape dans le checkout.

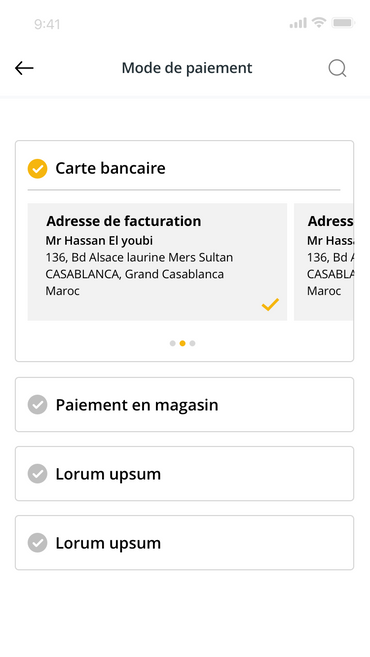
Le client ne peut pas accéder à cette étape si il n’est pas connecté. De ce fait, s’il n’est pas connecté, le client sera invité à se connecter ou créer un nouveau compte ( plus de détail sur le compte client dans le chapitre du compte client ).

Si le client est connecté, il atterrira directement dans l’étape de livraison, ou il devra choisir son adresse de livraison , ou saisir une nouvelle. Si le client n’a pas une adresse déjà enregistrée dans son compte, le formulaire de saisi d’adresse sera affiché.

Afin de valider cet étape, le client doit choisir une adresse ou saisir une nouvelle.

Figure 28 - wireframe - livraison à domicile - livraison

### 5.1.4 Paiement



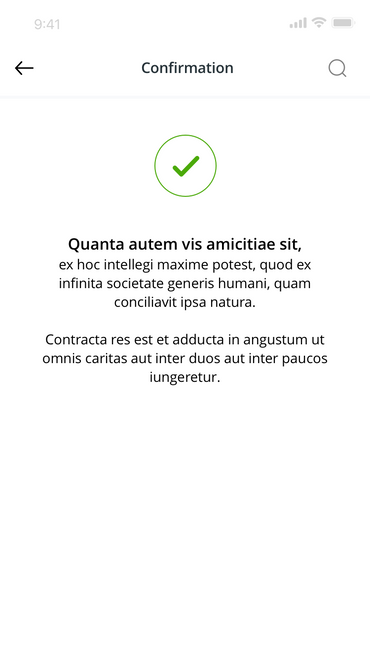
Les modes de paiement disponibles dans la boutique en ligne seront affichées dans cette étape à savoir, paiement à la livraison, paiement par carte, Paiement par Fatourati.

Le client doit choisir un mode de paiement, ainsi que l’adresse de facturation ( choisir dans le carnet d’adresse ou saisir une nouvelle ).

Si le client choisit le paiement par carte bancaire, à la validation de sa commande, la page CMI sera affiché en webview.

Figure 29 - wireframe - paiement

### 5.1.5 Confirmation



Au retour de CMI dans le cas de paiement par carte bancaire, ou à la simple validation de l’étape de paiement pour les autres modes, Une page de confirmation d’achat sera affiché.

Figure 30 - Wireframe - Confirmation

## 5.2 Click & Collect

### 5.2.1 Ajout au panier

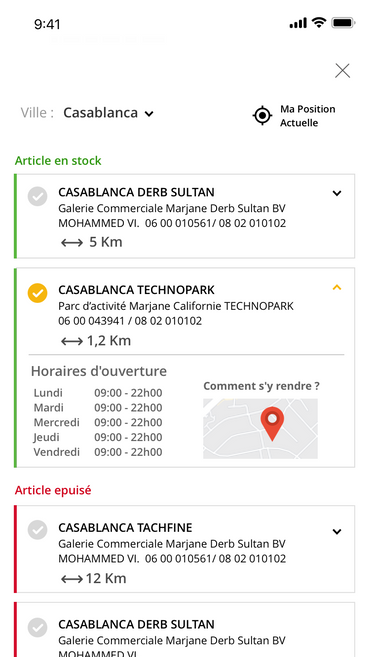
Si le client choisit l’option click & collect, le parcours change. En fait le client devra choisir le magasin dans lequel il devrait récupérer ses produits. Il ne peut, alors, ajouter que les produits en stock dans le magasin qu’il a choisi.

S’il voudrait modifier son magasin, il sera obligé de vider son panier dans le process.

Comme dans le wireframe, le choix du click and collect , oblige le choix du magasin.

* Si le client a déjà un magasin choisi, celui-ci sera sélectionné par défaut tout en donnant la possibilité de le modifier.   
  Un lien comment s’y rendre est aussi affiché afin de donner la possibilité au client de voir le trajet depuis sa position actuelle ( utilisation du natif de l’OS )  
  *Si le client change de magasin alors que son panier contient déjà un produit, son panier sera vidé, à l’ajout du nouveau produit. Une notification explicatif sera affichée avant de vider le panier.*
* Si le client n’a aucun magasin choisi, seul le lien du choix du magasin sera affiché.

Figure - wireframe – ajout panier - Click & Collect



La liste des magasins se présentera avec des

Flags vert pour les magasin dans lequel le produit est

en stock et en rouge pour les magasin dans lequel le

produit est en rupture de stock. Les magasins seront

ordonnés comme dans le wireframe.

Après l’ajout du produit au panier, le parcours est défini comme click&collect avec le magasin choisi.

### 5.2.2 Panier

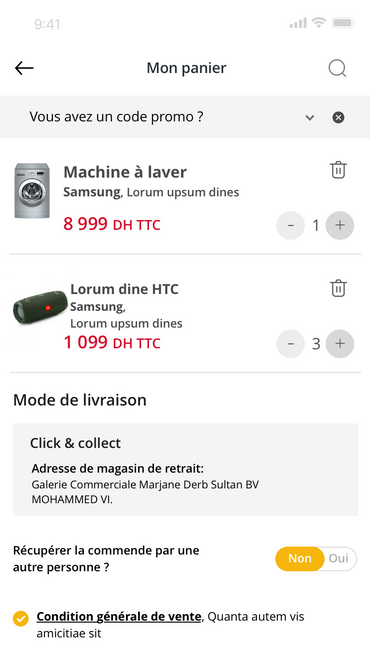
Le panier pour le parcours click & collect est le même que celui affiché pour le parcours de livraison à domicile.

Le magasin du click & collect sera affiché dans le panier.

*Figure 32 - Wireframe - Click & Collect - magasin*

### 5.2.3 Livraison

Comme pour le parcours du livraison en magasin, l’étape de récupération n’est pas disponible en mode non connecté, le client est obligé de se connecter avant de passer à cette étape.



Dans cette étape, le client ne peut pas modifier la quantité du produit ou le supprimer.

Le magasin choisi est affiché, mais le client n’aura pas moyen de le modifier.

Un switcher sera affiché pour donner le choix au client si c’est lui qui va récupérer le produit, ou s’il sera récupéré par une autre personne. Dans le deuxième cas, un formulaire sera affiché pour saisir les informations de la personne qui récupérera le produit.

### 5.2.4 Paiement

L’étape du paiement dans le parcours du click & collect est la même que celle pour le parcours de livraison à domicile.

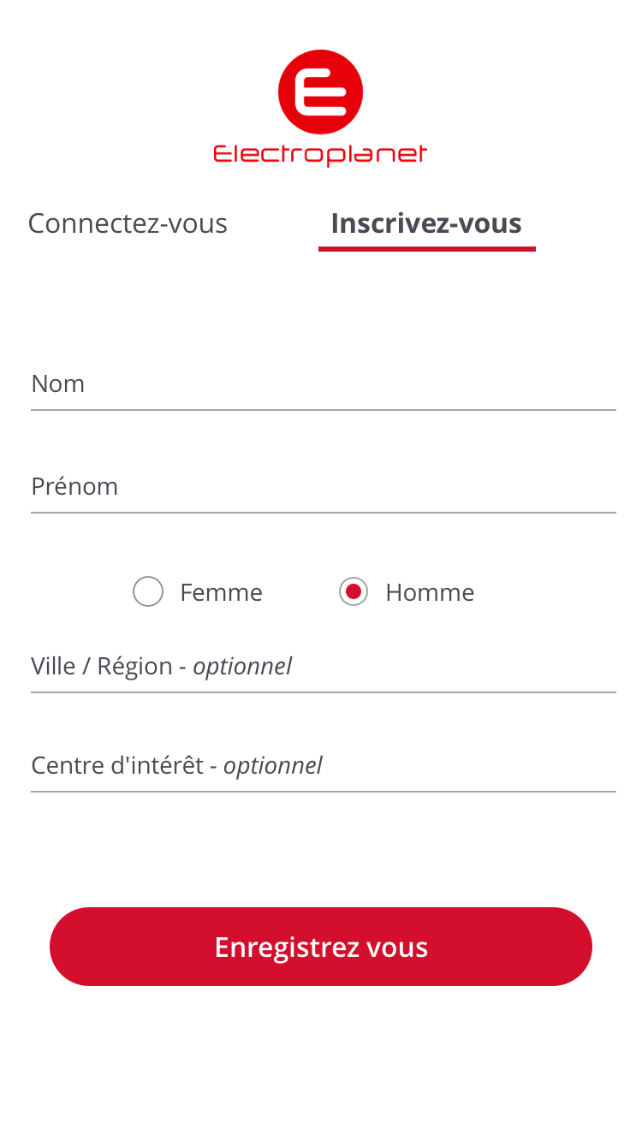
### 5.2.5 Confirmation

L’étape de confirmation est la même que celle du parcours de livraison à domicile

Figure 33 - Wireframe - livraison - Click & Collect

# 6 Espace client

## 6.1 Création de compte

La création de compte sera possible dans l’application mobile, de la même façon que la création de compte sur le site web.

Un formulaire avec les mêmes champs que le site sera implémenté sur l’application mobile,

La possibilité de création de compte via google / facebook sera mise en place sur l’application mobile.

Après la création de compte, le client devra confirmer son email avant de pouvoir se connecter

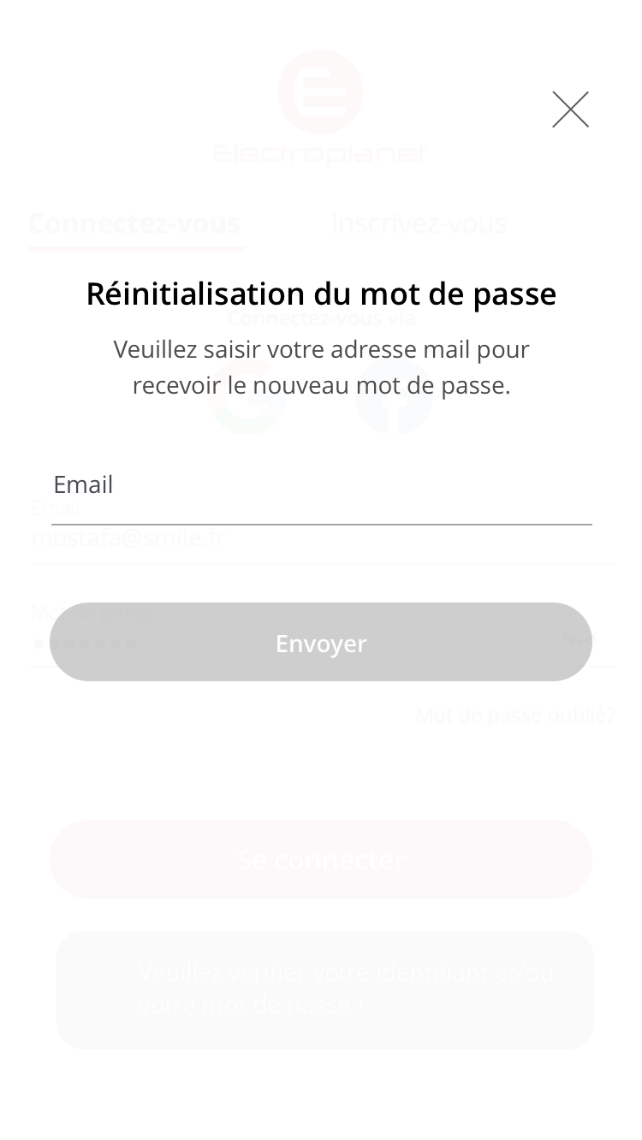
## 

Figure 34 - wireframe - espace client - Création de compte

## 6.2 Authentification

Comme dans le site, sur l’application mobile il est possible de s’authentifier avec son login ( adresse email ) et mot de passe. Comme il est possible de s’authentifier via son compte google / facebook.

Figure 35 - wireframe - espace client - authentification



## 6.3 Mot de passe oublié

Il est aussi possible de récupérer son mot de passe depuis l’application mobile, en saisissant l’adresse email.

Un email sera envoyé, et la page de réinitialisation de mot de passe sera affiché afin de modifier le mot de passe.

Figure 36 - wireframe - espace client - passoublier

## 6.3 Tableau de bord

A l’accès dans l’espace client, le tableau de bord est affiché, donnant accès à toutes les pages de l’espace client.

Figure 37 - wireframe - espace client - dashboard



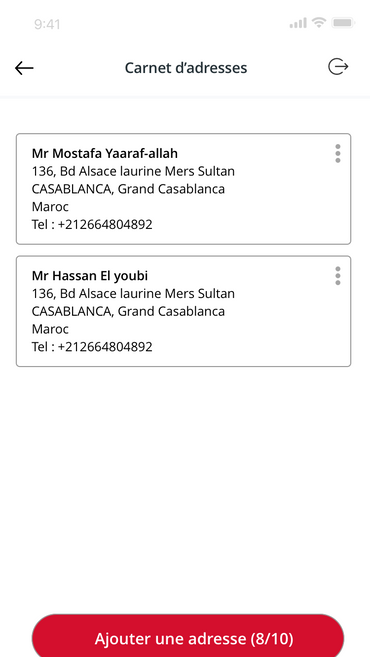
## 6.4 Information du compte

Cette entrée permettra de modifier les informations du client, à savoir :

* Nom
* Prénom
* Adresses mail
* Mot de passe

Figure 38 - wireframe - espace client – modif compte

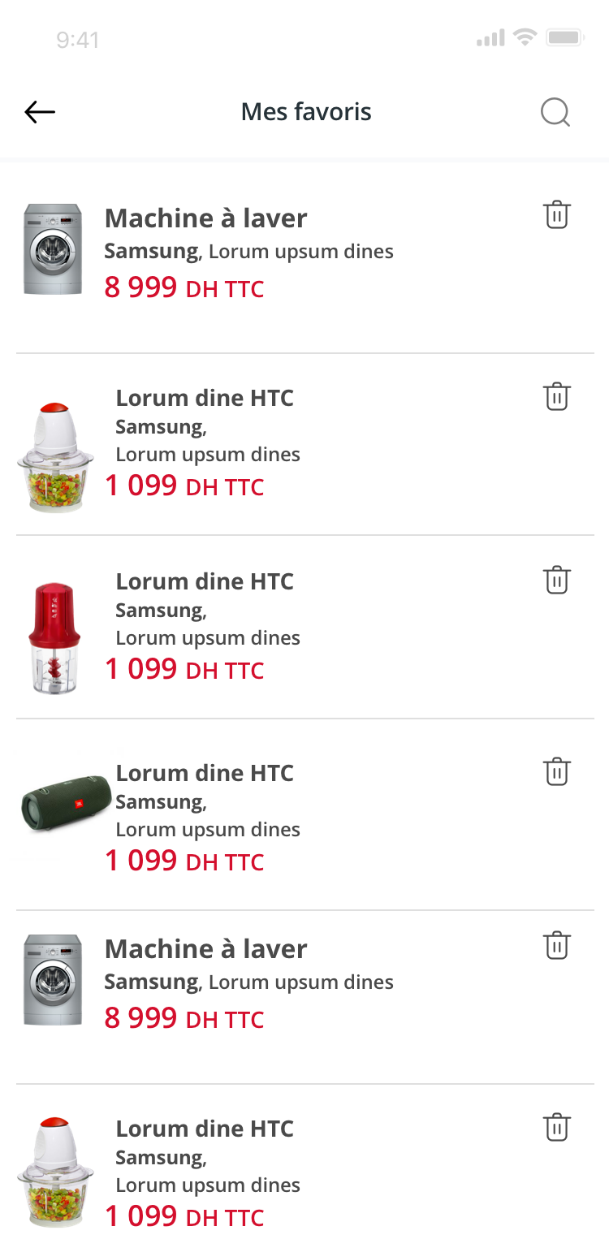
## 6.5 Carnet d’adresse



Magento met à disposition un carnet des adresses pour le client. La liste des adresses client sera affichée dans l’application mobile, le client pourra alors dans la limite de 10 ajouter des adresses, aussi aura-t-il la possibilité de modifier et supprimer une adresse.

*Figure 39 - wireframe - Carnet d’adresse*

## 6.6 Liste des favoris

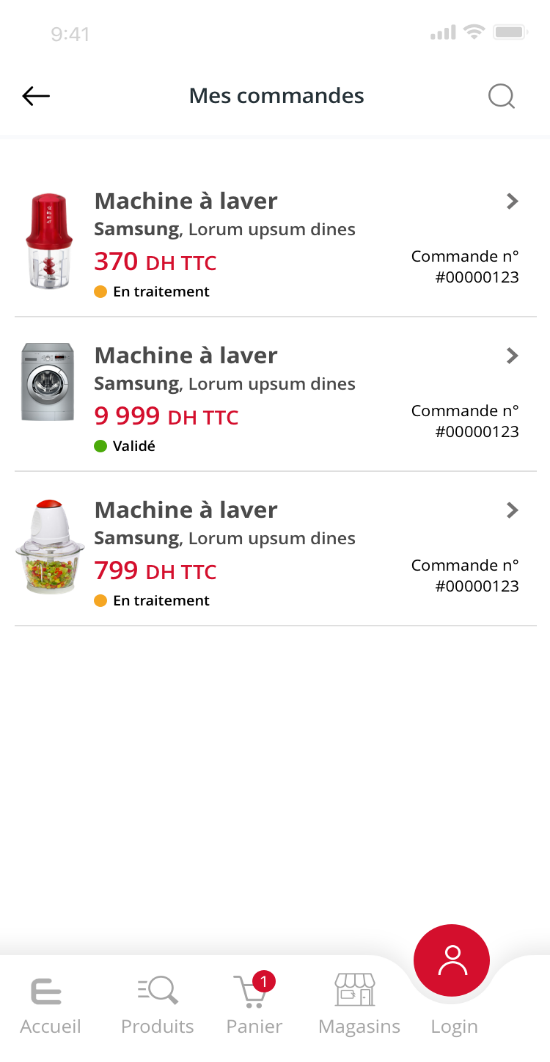


La liste des favoris du client sera aussi associé à son compte. Le client pourra ainsi voir les produits dans sa liste des favoris, il pourra aussi supprimer les produits dans sa liste des favoris

## 

*Figure 40 - wireframe - Liste des favoris*

## 6.7 Liste des commandes



Toute les commandes du client sont affichées dans la liste des commandes de l’espace client.

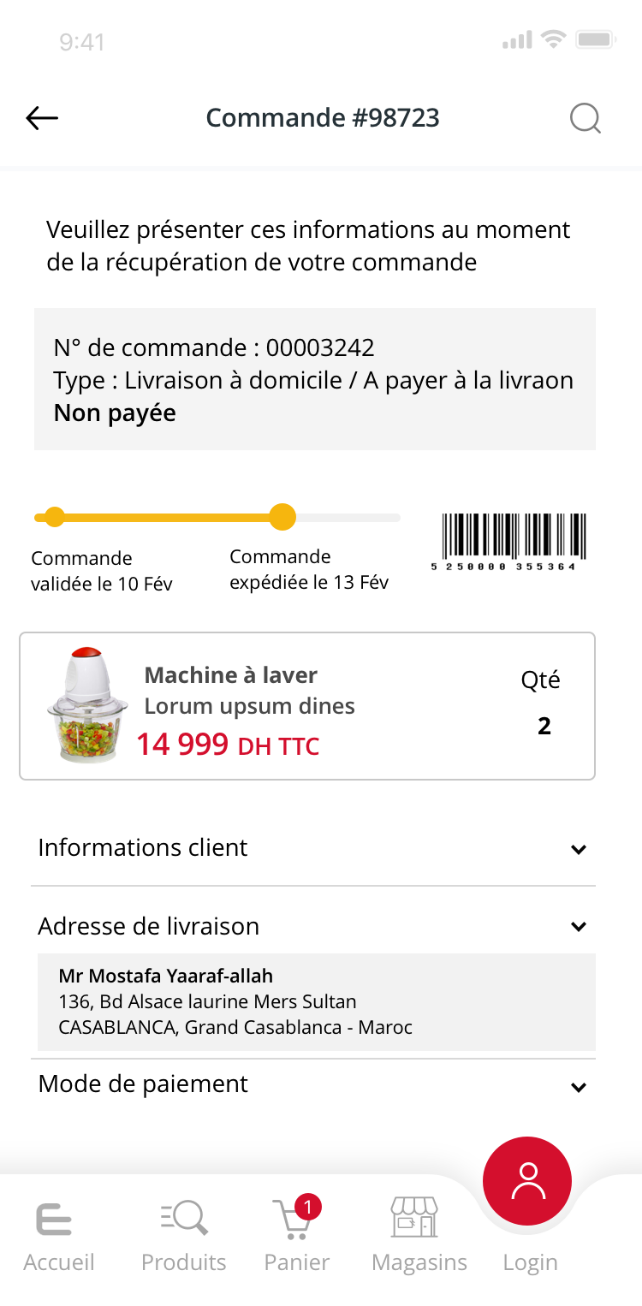
Le client pourra ainsi voir la liste de ses commandes et ainsi voir le détail de chacune dans le détail de la commande.

Dans la liste des commande, le montant de la commande ainsi que son statut sont affichées.

## 

*Figure 41 - wireframe - liste des commandes*

## Détail de la commande



Le détail de la commande affiche l’identifiant de la commande, ainsi que :

* Mode de livraison
* Mode de paiement
* Statut
* Code à barre
* Produits et leur quantités
* Détail du prix total de la commandes

*Figure 42 - wireframe - Détail de la commande*

# **Conclusion général**

Ce projet de fin d’études avait pour but de nous apporter une expérience solide au milieu professionnel. Nous devons avouer que nous sommes satisfaites d’avoir effectué ce travail au sein du SMILE MAROC, et qui était une opportunité pour appliquer toutes les connaissances que nous avons acquise durant notre parcours universitaire.

L’objectif était principalement de réaliser une application pour le commerce mobile sous Android, afin de rendre l’opération de l’achat plus facile et rapide pour le client.

Dans un premier lieu, nous nous sommes documentées sur différentes concepts et technologies contemporaines à travers plusieurs autoformations encadrées par Mr. RIAHI et Mr. CHBAB. Ce qui nous a fait découvrir le sens de la recherche et de la responsabilité, grâce à laquelle nous avons pu surmonter les difficultés et mener à bien un projet en équipe.

Ensuite, nous avons procédé à la réalisation des travaux pratiques dans lesquels nous avons bien familiarisé avec les technologies ainsi avec l’environnement de travail.

Après la validation du cahier des charges de l’entreprise avec son client, nous avons passé à la phase d’analyse de notre projet, dans laquelle nous avons commencé à bien comprendre les besoins fonctionnels et non fonctionnels de notre application.

La dernière phase de notre projet concernait la réalisation de notre application mobile. D’abord, nous avons commencé par la création des interfaces similaires à la maquette validée avec le client. Ensuite, nous avons passé à l’implémentation de la partie web service en parallèle avec la création des interfaces de l’application mobile en utilisant l’IDE Android Studio et XCode.

Ce projet de fin d’études nous a permis du côté technique de découvrir plusieurs tendances technologiques et d'approfondir notre connaissance sur les technologies contemporaines. De même qu’il nous a offert une opportunité pour apprendre à travailler en équipe, à gérer notre temps, à diviser les tâches et à coopérer pour aboutir au résultat requis.

# BIBLIOGRAPHIES

Mark L.Murphy, GraphQL Android, Commons Ware, 2019. 263.

[Anthony Cosson,](https://www.eyrolles.com/Accueil/Auteur/anthony-cosson-145660/) Les fondamentaux du développement d'applications Android, [Collection Epsilon,218,](https://www.eyrolles.com/Informatique/Collection/6745/epsilon/) 466.

Mark Gargenta, Learning Android, O’Reilly, 2011, 268.

# WEBOGRAPHIES

<https://medium.com/swlh/understanding-mvvm-architecture-in-android-aa66f7e1a70b> <https://medium.com/mindorks/using-mvvm-android-jetpack-ebf1d9c1477c> <https://www.supinfo.com/articles/single/145-application-mobile-native-web-hybride> <https://www.definitions-marketing.com/definition/application-native/> <https://github.com/instamobile/android-login-screen> <https://openclassrooms.com/fr/courses/5353106-initiez-vous-a-kotlin> [https://fr.wikipedia.org/wiki/M-Commercehttps://topandroid.fr](https://fr.wikipedia.org/wiki/M-Commerce)

<http://tutos-android-france.com/constraintlayout-partie-1/>

<https://fr.wikipedia.org/wiki/Swift_(langage_d%27Apple)>

https://fr.wikipedia.org/wiki/Xcode