REPUBLIQUE TUNISIENNE

Ministère de l’Enseignement Supérieur et de la Recherche Scientifique

Institut Supérieur des Études Technologiques de Sousse

**Rapport Projet Fin d’Étude**

Mise en Place d’une Solution e-Commerce « MedEspoir Shop »

Encadrante académique : Hédia Jegham

Encadrant entreprise : Zied Memi

Remerciement

Au terme de mon projet de fin d’études, je tiens à adresser mes plus vifs remerciements à toutes les personnes qui, de près ou de loin, ont contribué à l’aboutissement de ce travail dans les meilleures conditions. Je m’adresse en premier lieu aux membres de l’honorable jury que je remercie d’avoir accepté d’examiner ce rapport. Je tiens à exprimer toute ma gratitude à mon encadrante madame **Hedia JGHAM**, pour le temps consacré à la lecture et aux réunions qui ont rythmées les étapes de mon projet de fin d’étude. Je la remercie aussi pour sa disponibilité à encadrer ce travail à travers ses critiques et ses propositions d’amélioration. Un remerciement particulier à mon encadreur dans la société monsieur **Zied MEMI**, pour le soutien qu’il m’a apporté tout au long du stage. Nous profitons de cette occasion, pour exprimer notre profonde reconnaissance à tous les enseignants pour leurs formations et leurs précieux conseils tout au long de nos années d’étude.

Table des matières

[Introduction Générale 1](#_Toc74873385)

[Chapitre 1 : Lancement du Projet "Med Espoir Shop" 3](#_Toc74873386)

[1. Présentation de l’organisme d’accueil 4](#_Toc74873387)

[1.1. TouchLink 4](#_Toc74873388)

[1.2. Domain d’expertise 4](#_Toc74873389)

[2. Présentation du projet 5](#_Toc74873390)

[2.1. Opportunité et client du projet 5](#_Toc74873391)

[2.2. Public cible 5](#_Toc74873392)

[2.3. Produits et services vendus 6](#_Toc74873393)

[2.4. Contexte générale et problématique 6](#_Toc74873394)

[2.5. Solution proposée 6](#_Toc74873395)

[3. Cahier des charges de Med Espoir Shop 8](#_Toc74873396)

[3.1. Identification des acteurs 8](#_Toc74873397)

[3.2. Etude des besoins fonctionnels 8](#_Toc74873398)

[3.2.1. Inscription des clients 8](#_Toc74873399)

[3.2.2. Exposition des produits 9](#_Toc74873400)

[3.2.3. Gestion d’un panier 9](#_Toc74873401)

[3.2.4. Gestion des favoris 9](#_Toc74873402)

[3.2.5. La confirmation de la commande 9](#_Toc74873403)

[3.2.6. Le Mode de livraison 9](#_Toc74873404)

[3.2.7. Le paiement 9](#_Toc74873405)

[3.2.8. La finalisation de la transaction en ligne 10](#_Toc74873406)

[3.2.9. Le Dashboard 10](#_Toc74873407)

[3.3. Etude des besoins non fonctionnels 10](#_Toc74873408)

[3.3.1. La Sécurité 10](#_Toc74873409)

[3.3.2. L’Ergonomie 10](#_Toc74873410)

[3.3.3. La Performance 11](#_Toc74873411)

[4. Méthodologie de travail 11](#_Toc74873412)

[4.1. Etude comparative des méthodes 11](#_Toc74873413)

[4.2. Choix de la méthode 12](#_Toc74873414)

[4.3. Présentation du cadre de processus SCRUM 12](#_Toc74873415)

[4.3.1. Les événements de Scrum 13](#_Toc74873416)

[4.3.2. Les rôles dans Scrum 14](#_Toc74873417)

[4.3.3. Les objets de Scrum 15](#_Toc74873418)

[5. Elaboration et priorisation du Product Backlog 15](#_Toc74873419)

[6. Planifications des sprints 17](#_Toc74873420)

[7. Représentation schématique du PB par diagramme de cas d’utilisation 18](#_Toc74873421)

[8. Architecture globale du projet 20](#_Toc74873422)

[8.1. Architecture physique 20](#_Toc74873423)

[8.2. Architecture logique 20](#_Toc74873424)

[Chapitre 2 Sprint1 : Les Modules d’Inscription et Authentification de Med Espoir Shop 22](#_Toc74873425)

[1. Backlog du premier sprint 23](#_Toc74873426)

[2. Analyse 24](#_Toc74873427)

[2.1. Raffinement du cas utilisation du sprint 1 24](#_Toc74873428)

[2.1.1. Cas « inscription » 25](#_Toc74873429)

[2.1.2. Cas « authentification » 25](#_Toc74873430)

[2.1.3. Cas « restaure mot de passe » 26](#_Toc74873431)

[2.1.4. Cas « vérifier email » 26](#_Toc74873432)

[3. Conception 27](#_Toc74873433)

[3.1. Diagramme de classes du sprint 1 27](#_Toc74873434)

[3.2. Diagrammes des séquences 27](#_Toc74873435)

[3.2.1. Cas « inscription » 28](#_Toc74873436)

[3.2.2. Cas « authentifier » 29](#_Toc74873437)

[3.2.3. Cas « restaurer mot de passe » 30](#_Toc74873438)

[4. Environnement de développement 31](#_Toc74873439)

[4.1. Technologies 31](#_Toc74873440)

[4.2. Environnement logiciel 31](#_Toc74873441)

[4.3. Environnement matériel 32](#_Toc74873442)

[5. Sprint review 32](#_Toc74873443)

[5.1. Cas « authentification » 33](#_Toc74873444)

[5.2. Cas « inscription » 34](#_Toc74873445)

[5.3. Cas « mot de pass oublier » 35](#_Toc74873446)

[Chapitre 3 Sprint 2 : Developpement des modules du back-office de MedEspoir Shop 37](#_Toc74873447)

[1. backlog du deuxieme sprint 38](#_Toc74873448)

[2. Analyse 39](#_Toc74873449)

[2.1. Raffinement du cas utilisation du sprint 2 39](#_Toc74873450)

[2.1.1. Cas « gestion des produits » 40](#_Toc74873451)

[2.1.2. Cas « contacter client » 41](#_Toc74873452)

[2.1.3. Cas « suivi activités » 42](#_Toc74873453)

[3. Conception 42](#_Toc74873454)

[4. Sprint Review 44](#_Toc74873455)

[4.1. Cas « gestion des produits » 44](#_Toc74873456)

[5. Sprint rétrospective 46](#_Toc74873457)

[Chapitre 4 : Sprint 3,4,5 : Developpement des modules du front-office de MedEspoir Shop 48](#_Toc74873458)

[1. Sprint 3 « Gestion du catalogue et module de recherche » 49](#_Toc74873459)

[1.1. Backlog du troisième sprint : 49](#_Toc74873460)

[1.2. Analyse 50](#_Toc74873461)

[1.2.1. Raffinement du cas utilisation du sprint 3 50](#_Toc74873462)

[1.3. Conception 53](#_Toc74873463)

[1.3.1. Diagramme de classes du sprint 3 53](#_Toc74873464)

[1.3.2. Diagramme de séquence sprint 3 54](#_Toc74873465)

[4. Sprint review 55](#_Toc74873466)

[2. Sprint 4 « Gestion du caddy» 59](#_Toc74873467)

[2.1. Backlog du quatrième sprint 59](#_Toc74873468)

[2.2. Analyse 59](#_Toc74873469)

[2.2.1. Raffinement du cas utilisation du sprint 4 59](#_Toc74873470)

[2.3. Conception 61](#_Toc74873471)

[2.4. Sprint review 63](#_Toc74873472)

[3. Sprint 5  « intégration des modules de paiement en ligne » 67](#_Toc74873473)

[3.1. Backlog du cinquième sprint 67](#_Toc74873474)

[3.2. Analyse 67](#_Toc74873475)

[3.2.1. Raffinement du cas utilisation du sprint 5 67](#_Toc74873476)

[3.3. Conception 71](#_Toc74873477)

[3.3.1. Diagramme de class du sprint 5 71](#_Toc74873478)

[3.4. Sprint review 72](#_Toc74873479)

[Conclusion Générale 75](#_Toc74873480)

Liste des figures

[Figure 1: logo enterprise 4](file:///D:\files\web%20work\med-shop\docs\rapport\final%20version.docx#_Toc74880126)

[Figure 2: logo de Med Espoir shop 7](#_Toc74880127)

[Figure 3: cycle de vie de méthode Scrum 13](#_Toc74880128)

[Figure 4: Azure DevOps 13](#_Toc74880129)

[Figure 5: digramme de cas utilisation "client" 18](#_Toc74880130)

[Figure 6: digramme de cas utilisation "admin" 19](#_Toc74880131)

[Figure 7: architecture physique 20](#_Toc74880132)

[Figure 8: architecture logique 20](#_Toc74880133)

[Figure 9: diagramme de cas utilisation Authentification 24](#_Toc74880134)

[Figure 10: diagramme de classe sprint 1 27](#_Toc74880135)

[Figure 11: diagramme de séquence du cas utilisation s'inscrire 28](#_Toc74880136)

[Figure 12: diagramme de séquence du cas utilisation authentifier 29](#_Toc74880137)

[Figure 13: diagramme de séquence du cas utilisation restaurer mot de passe 30](#_Toc74880138)

[Figure 14: cas authentification version web 33](#_Toc74880139)

[Figure 15: cas authentification version responsive -mobile- 33](#_Toc74880140)

[Figure 16: cas inscription version web 34](#_Toc74880141)

[Figure 17: cas inscription version responsive -mobile- 34](#_Toc74880142)

[Figure 18: cas restaurer mot de passe version web 35](#_Toc74880143)

[Figure 19: cas restaurer mot de passe version responsive -mobile- 35](#_Toc74880144)

[Figure 20: diagramme de cas utilisation sprint 2 39](#_Toc74880145)

[Figure 21: diagrammes de class sprint 1 43](#_Toc74880146)

[Figure 22: liste des produits 44](#_Toc74880147)

[Figure 23: cas ajouter un produit 44](#_Toc74880148)

[Figure 24: cas supprimer un produit 45](#_Toc74880149)

[Figure 25: cas modifier un produit 45](#_Toc74880150)

[Figure 26: diagramme de cas utilisation sprint 3 50](#_Toc74880151)

[Figure 27: diagramme de classe sprint 3 53](#_Toc74880152)

[Figure 28: diagramme de séquence de recherche global 54](#_Toc74880153)

[Figure 29: filtre produit sous-catégorie 55](#_Toc74880154)

[Figure 30: cas recherche générale 56](#_Toc74880155)

[Figure 31: detail du produit 56](#_Toc74880156)

[Figure 32: commenter sur un produit 57](#_Toc74880157)

[Figure 33: cas ajoute produit à wishlist 58](#_Toc74880158)

[Figure 34: raffinement de cas utilisation du sprint 4 59](#_Toc74880159)

[Figure 35: diagramme de séquence sprint 4 -gestion caddy- 62](#_Toc74880160)

[Figure 36: cas ajout au panier -caddy vide- 63](#_Toc74880161)

[Figure 37: cas ajout au panier -produit existe- 64](#_Toc74880162)

[Figure 38: cas ajout au panier -diffèrent produit- 64](#_Toc74880163)

[Figure 39: cas supprimer un produit de panier 65](#_Toc74880164)

[Figure 40: cas modifier la quantité de produit 65](#_Toc74880165)

[Figure 41: drawer du caddy -version mobile- 66](#_Toc74880166)

[Figure 42:raffinement de cas utilisation du sprint 5 68](#_Toc74880167)

[Figure 43: diagramme de class sprint 5 71](#_Toc74880168)

[Figure 44: cas "choisir la méthode de paiement" 72](#_Toc74880169)

[Figure 45: méthode de paiement stripe 72](#_Toc74880170)

[Figure 46: méthode de paiement paypal 73](#_Toc74880171)

[Figure 47: transaction monétique avec paypal 73](#_Toc74880172)

Liste des tableaux

[Tableau 1: étude comparative des méthodes 11](#_Toc74873528)

[Tableau 2: product backlog ''user stories'' 16](#_Toc74873529)

[Tableau 3 : planification des sprints 17](#_Toc74873530)

[Tableau 4: backlog sprint1- authentification- 23](#_Toc74873531)

[Tableau 5 : description textuelle inscription 25](#_Toc74873532)

[Tableau 6 : description textuelle authentification 25](#_Toc74873533)

[Tableau 7: description textuelle de restaurer mot de passe 26](#_Toc74873534)

[Tableau 8: description textuelle de vérifier email 26](#_Toc74873535)

[Tableau 9: les technologies de web 31](#_Toc74873536)

[Tableau 10: les logiciels 31](#_Toc74873537)

[Tableau 11: pc specs 32](#_Toc74873538)

[Tableau 12: backlog sprint 2-dashboard- 38](#_Toc74873539)

[Tableau 13: description textuelle pour ajoute un produit 40](#_Toc74873540)

[Tableau 14: description textuelle pour modifier un produit 40](#_Toc74873541)

[Tableau 15: description textuelle pour supprimer un produit 41](#_Toc74873542)

[Tableau 16: description textuelle pour contacter un client 41](#_Toc74873543)

[Tableau 17: description textuelle pour le suivi des activités 42](#_Toc74873544)

[Tableau 18: backlog sprint 3-gestion catalogue et recherche- 49](#_Toc74873545)

[Tableau 19: description textuelle de filtre produit 51](#_Toc74873546)

[Tableau 20: description textuelle de recherche générale 51](#_Toc74873547)

[Tableau 21: description textuelle ajouter au wishlist 52](#_Toc74873548)

[Tableau 22: description textuelle review produit 52](#_Toc74873549)

[Tableau 23: backlog sprint 4 -gestion de caddy- 59](#_Toc74873550)

[Tableau 24: description textuelle cas ajout produit au panier 60](#_Toc74873551)

[Tableau 25: description textuelle cas supprimer produit de panier 60](#_Toc74873552)

[Tableau 26: description textuelle cas de modification de panier 61](#_Toc74873553)

[Tableau 27: backlog sprint 5-integration des modules de paiement en ligne 67](#_Toc74873554)

[Tableau 28: sprint 5 - description textuelle de vérifier les ordres placés 68](#_Toc74873555)

[Tableau 29: sprint 5-descriprion textuelle de choisir méthode de paiement 69](#_Toc74873556)

[Tableau 30: sprint 5 - description textuelle de voir historique est recommander 69](#_Toc74873557)

[Tableau 31: sprint 5-description textuelle de client suivi son ordre 70](#_Toc74873558)

[Tableau 32: sprint 5-description textuelle d'administrateur suivi les ordres 70](#_Toc74873559)

Liste Abréviations

* **API** = **A**pplication **P**rogramming **I**nterface
* **JS** = **J**ava**S**cript
* **PHP** = **P**rogramming **H**yperText **P**reprocessor
* **UML** = **U**nified **M**odel **L**anguage
* **SEO** = **S**earch **E**ngine **O**ptimization
* **SEM** = **S**earch **E**ngine **M**arketing
* **B2C** = **B**usiness to **C**onsumer
* **PDF** = **P**ortable **D**ocument **F**ormat
* **UX** = **U**ser E**x**perience
* **UI** = **U**ser **I**nterface
* **MVC**= **M**odel **V**iew **C**ontroller
* **URL** = **U**niform **R**esource **L**ocator
* **CRUD** = **C**reate, **R**ead, **U**pdate, **D**elete

# Introduction Générale

Le développement informatique ou le « coding » est une discipline à la mode, il est très riche car il est assujetti à une énorme variété de méthodes et de technologies. Il touche la quasi-totalité des domaines de vie de l’être humain rendant tout travail facile, convivial, rapide, précis et bien organisé.

Le commerce électronique comme tout autre domaine, n’échappe pas à cette règle. Ce concept a montré ses capacités à fournir des solutions de vente en ligne de produits et de services. Il a boosté le monde du Business et des affaires en créant une multitude de modèles : le B2C, le B2B, le B2B2C. Avec la survenue de la pandémie COVID-19, les entreprises de commerce de tout type, se sont acharnées pour développer leurs plateformes de vente en les connectant à toute la chaine de valeurs nécessaire : commande, paiement, livraison, promotion, retour, conseil et assistance… Tout ceci dans l’objectif de continuer à tirer le maximum de profit et pour que le client ne quitte pas chez lui et ne risque pas la contamination.

Certes le développement de plateformes de commerce électronique semble atteindre une phase de maturité et de démocratisation par l’offre de logiciels de type "Site Builder" ou encore par l’émergence des plateformes de commerce électronique de type SAAS "Software As A Service", ceci ne cache pas leur complexité de plus en plus croissante engendrée par l’ampleur des fonctionnalités qu’ils doivent constamment offrir et parfaire. Le catalogue, les recherches internes générale et spécialisée, les offres promotionnelles, la disponibilité en stock, l’implication des clients par les systèmes de vote et de notation, la gestion des favoris, la fidélisation, la commande et la relance de commande, le suivi des états de commandes, l’offre d’une large panoplie de modes de paiements, la livraison, l’assistance et la recommandation etc. Et bien que dans cette liste nous n’avons cité que des fonctionnalités visibles par le consommateur final, ils restent encore d’autres dédiés vers les administrateurs des plateformes de commerce électroniques.

C’est à ce titre qu’au cours de mon stage de fin d’études pour l’obtention du Diplôme de Licence en Développement Web et Multimédia, l’entreprise IT **TouchLink** m’a confié la tâche de développer une plateforme de commerce électronique qui répond aux attentes des clients de l’entreprise **Med espoir.** Cette dernière œuvre dans le domaine de l’esthétique et du bien-être et souhaite que sa plateforme soit baptisée **"Med Espoir Shop"** et soit d’une grande qualité.

Notre projet sera guidé par le cadre de processus **Scrum** et reposera sur deux technologies populaires : La bibliothèque **React-Js** pour le développement coté client et le Framework **Laravel** pour le développement coté server.

Le premier release du projet **"Med Espoir Shop"** a été atteint en 6 sprints, ce qui nous a amené à structurer notre rapport en 4 chapitres.

Dans le **premier chapitre** « lancement du projet » sera un chapitre introductif qui va placer le projet dans son contexte général.

Le **deuxième** chapitre est dédié pour le premier sprint qui a l’objectif de développer un module d’authentification.

Le **troisième** chapitre traite le deuxième sprint qui a pour objectif le développement des modules de back office.

Le **quatrième** chapitre est réservé pour les troisième, quatrième et cinquième sprint qui décrivent la gestion du catalogue et recherche, la gestion de caddy et intégration de module de payement.

Finalement une conclusion viendra évaluer le degré de satisfaction de notre client de ce premier release.

**Chapitre 1**

**Lancement du Projet "Med Espoir Shop"**

**Introduction**

Dans ce premier chapitre, nous présenterons tout d’abord le cadre général du mon projet ainsi que l’organisme d’accueil « **TouchLink** ». Ensuite, nous exposerons le cahier de charge de mon projet en mettant l’accent sur ses spécificités.

Durant ce chapitre, nous procèderont aussi à une étude comparative des différentes méthodologies de gestion de projet jusque-là utilisées afin de justifier notre choix.

1. **Présentation de l’organisme d’accueil**
   1. **TouchLink**

Le projet est réalisé au sein de l’entreprise TouchLink qui est une agence digitale. Sa mission est de guider ses clients vers une transformation digitale réussie en comptant sur son expertise, créativité et fiabilité. Pour une identification visuelle rapide de cette startup, la figure 1 montre son logo

Logo

Description automatically generated with medium confidence

Figure 1: logo enterprise

* 1. **Domain d’expertise**

TouchLink offre 4 grands types des services :

* **Développent Web et Mobile**

En couvrant tous les éléments nécessaires pour développer une application web ou mobile allant de la stratégie à la conception et au codage de tous les composants du produit tout en se basant sur l’originalité, la vitesse, la sécurité et flexibilité.

* **Marketing Digital**

TouchLink aide sa clientèle à concevoir une stratégie de marketing digital pouvant faciliter l’atteinte de ses objectifs grâce à plusieurs services offerts (SE0 & SEM, Social Media).

* **Design Graphique**

Enrichir l’interactivité et l’accessibilité des produits pensés dans les moindres détails, conçus pour optimiser l’expérience utilisateur en ajoutant des styles qui promeuvent les objectifs et les valeurs du produit

* **Content Marketing**

Le contenu est une pièce maitresse du site web, TouchLink dispose de rédacteurs de contenu multilingues bien formés à la rédaction de contenu constructif et à valeur ajoutée pour le client, en plus de son optimisation pour les recherches Google.

1. **Présentation du projet**

L’objectif de cette partie est de mettre le projet dans son contexte, à ce titre je vais détecter les problèmes rencontrés par son entreprise cliente Med Espoir afin de l’aider à formuler ses objectifs et lui présenter des solutions.

* 1. **Opportunité et client du projet**

Mon projet de fin d’étude intitulé « mise en place d’une solution e-commerce d’une boutique en ligne ‘’**Med Espoir Shop**’’ » me permettra d’obtenir un Diplôme de Licence en Technologies de l’Informatique de l’Institut Supérieure des Études Technologique de Sousse.

Med espoir, le client de notre projet, est une agence de tourisme médical spécialisée dans l’organisation de séjours médicaux et les offres des soins en chirurgie esthétique en Tunisie. Med espoir propose des services et des interventions de qualité dans tous les domaines de la chirurgie et offre aux patients tunisiens, la possibilité unique de bénéficier de tous sortes de soins à des conditions avantageuses.

* 1. **Public cible**

Les produits vendus dans Med Espoir Shop seront destinés aux femmes entre 18 et 55 ans et les personnes qui ont un grand intérêt aux produits cosmétiques. Les clients de Med Espoir sont aussi les cliniques qui utilisent les produits esthétiques. Ils sont aussi les jeunes Tunisiens et Français.

* 1. **Produits et services vendus**
* Alimentaires
* Produits de beauté
* Artisanat
* Livres
* Shooting photos
* Soins et massage
* Médecine esthétique
  1. **Contexte général et problématique**

La majorité des utilisateurs rencontrent aujourd’hui et surtout les Tunisiens des problèmes autour des e-boutiques et des transactions électroniques. Généralement, en Tunisie les utilisateurs préfèrent être en lieu pour faire leurs achats.

Les sites de vente en ligne permettent aux clients de profiter de sortes de foires virtuelles disponibles 24/24, 7/7 en plus d’être quotidiennement mise à jour. Cette omniprésence leur permettra de ne jamais rater les coups de cœur, les promotions d’autant plus qu’elles leur feront oublier les problèmes de distanciation géographique, l’horaire de travail et la disponibilité des moyens de transport.

De plus la limitation à avoir de la liquidité ou bien à avoir un carnet de chèque ou une carte bancaire, sachant que ces deux derniers obligent l’ouverture d’un compte bancaire, constituent un blocage pour l’accomplissement d’une transaction commerciale et engendrent des pertes d’occasions de ventes pour Med Espoir. Penser à intégrer un large spectre de modes de paiement est sans doute un effort investi pour garder son client et ne jamais le laisser partir vers la concurrence.

* 1. **Solution proposée**

Med Espoir-Shop est une application web de vente en ligne de type B2C, offrant un large éventail de fonctionnalités permettant de coordonner entre les acteurs de l’écosystème commercial (Vendeur, Clients, Organismes de paiement, Livreurs). Ce dernier et destiné vers l’entreprise « Med Espoir » d’une façon directe. La figure 2 donne une idée visuelle de notre client MedEspoir.

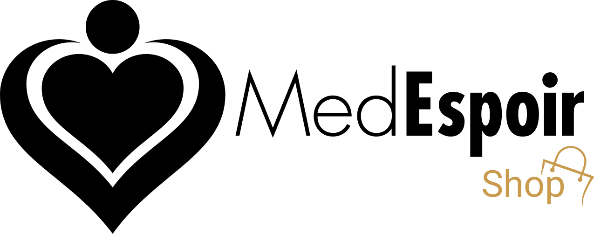


Figure 2: logo de Med Espoir shop

L’objectif du site est de donner à Med Espoir une présence continue à travers le web, qui lui permettra de publier ses services, ses produits, ses promotions, ses offres spéciales. Recevoir des commandes et des règlements en lignes et par conséquent augmenter son chiffre d’affaires.

Med Espoir souhaite aussi, par le biais de son site Med Espoir Shop, faciliter toutes les activités commerciales et financières de back-office qui incombent à ses employés.

Med Espoir souhaite en outre, offrir à sa clientèle et ses employés un design attrayant mais minimal et moderne facilement accessible par tout utilisateur. Med Espoir Shop devra être constitué des modules suivants :

* **Gestion des produits**

C’est un module qui permet à l’administrateur de gérer tout ce qui relève des fiches de produits, stocks, catégorisation, promotions, nouveautés en entrée de gamme en plus des avis, commentaires et feed-back des clients concernant les produits, est la gestion des actualités

* **Gestion Financière**

Ce module est utilisé par l’administrateur et orienté vers le gérant de l’entreprise facilite la gestion des factures et élabore diverses statistiques sur les ventes et le chiffre d’affaires.

* **Paiement en ligne**

Pour dépasser l’option de paiement à la livraison, ce module destiné aux clients doit offrir une multitude de modes de paiement en ligne. Il garantira la retenue d’une large clientèle.

Il nous a été donc demandé de nous intégrer dans l’équipe web de TouchLink pour concevoir et implémenter les modules spécifiques de la plateforme Med Espoir Shop.

1. **Cahier des charges de Med Espoir Shop**
   1. **Identification des acteurs**

Après une analyse profonde sur les interactions internes et externes de notre système, nous sommes arrivés à conclure que les acteurs d’un site e-commerce sont presque standard :

* **Le visiteur**

C’est un individu qui est entrain de fouiller sur le net, cherchant un produit pour l’acheter ou pour avoir une idée sur les produits et les prix.

* **Le client**

Cet acteur est un visiteur ayant déjà créé un compte sur le site marchand, il peut donc suivre le processus d’achat des produits en toute sécurité sachant que le système de Med Espoir Shop doit être l’unique responsable de la confidentialité des données personnelles de ses client.

* **L’administrateur**

Pour les sites web on l’appelle généralement « le web master ». C’est celui qui assure le dynamisme du site et veille sur les mises à jour des produits, les blogs, les stocks, la gestion de paiement et la gestion des livraisons. En réalité il s’agit de plusieurs métiers et spécialités : l’administrateur informatique, le commercial, le responsable marketing, le gestionnaire de stock, le livreur, le financier, le rédacteur de contenu. Par souci de simplification l’acteur Admin réunit ces différents rôles.

Remarque : nous avons utilisé la langue anglaise dans les diagrammes pour rassembler au code qui est écrite en anglais.

* 1. **Etude des besoins fonctionnels**

Un besoin fonctionnel est utilisé pour spécifier les actions qu’un système doit effectuer, dans cette partie de chapitre on va lister les besoins fonctionnels de notre système en les rapprochant aux modules cités plus haut.

* + 1. **Inscription des clients**

Le client est toujours anonyme mais pour pouvoir passer à une étape plus sérieuse (commander ou ajouter aux favoris), il faut qu’il s’inscrive, ce là se fait uniquement pour la première commande mais après, notre client peut s’authentifier avec son courrier électronique et son mot de passer pour voir l’historique de ses commandes ou passer une autre commande.

Le client doit recevoir un courrier de vérification lors de création de compte et sera capable de restaurer son mot de passe si oublie.

* + 1. **Exposition des produits**

Le site doit disposer d’une vitrine virtuelle à travers laquelle le client peut consulter une grande variété des produits classés par catégories, il sera donc indispensable d’y présenter les prix et les caractéristiques de chaque produit pour faciliter au client la sélection.

* + 1. **Gestion d’un panier**

Après le choix d’un produit le client doit mentionner la quantité qu’il désire, celle-ci se répercutera automatiquement dans la présentation du panier et sur le prix total. La suppression et la modification seront aussi permises dans le panier sans besoin de retour dans le catalogue

* + 1. **Gestion des favoris**

Une liste de souhaits permet aux acheteurs de créer des collections personnalisées de produits qu’ils souhaitent acheter et de les enregistrer dans leur compte utilisateur pour référence future. Les listes de souhaits signifient l’intérêt d’un client pour un produit sans intention immédiate d’achat.

* + 1. **La confirmation de la commande**

Cette phase nécessite la présence d’un client, un panier et une adresse de livraison, elle déclenche la transformation du panier (caddy) en une commande. La commande ne passera qu’après la validation de toutes les informations relatives au client. Celles-ci sont affichées dans une seule interface avant de passer à la phase de paiement.

* + 1. **Le Mode de livraison**

Un client qui a déjà confirmé sa commande, est libre de choisir si la livraison devra être faite dans une clinique de Med Espoir ou bien à domicile. Par croissant à domicile le client devrait remplir soigneusement un formulaire contenant les informations nécessaires.

* + 1. **Le paiement**

C’est une phase très sensible et importante, pour cela elle doit être très sécurisée. Pour terminer la procédure de paiement avec succès le client doit être connecté à son compte tout d’abord ensuite choisir la méthode de paiement qui lui convient (Carte, PayPal ou à la livraison). Une sortie et interfaçage avec une plateforme de paiement est nécessaire. A ce stade l’enjeu de la sécurité est énorme.

* + 1. **La finalisation de la transaction en ligne**

Si la transaction monétaire s’est bien déroulée, un message récapitulatif portant un identifiant unique de la transaction commerciale et présentant son détail est présenté au client avec remerciement. Le message peut être exporté au format PDF en plus d’être dupliqué par le biais d’un envoi de mail. Pour des besoins de sécurité et de marketing le client est réacheminé vers le magasin.

* + 1. **Le Dashboard**

C’est un module qui permet à l’administrateur de gérer tout ce qui est produit, prix, promotions, catégories, stocks ainsi qu’une gestion financière : l’offre de statistiques, les nouveautés et la logistique d’approvisionnement et de livraison.

* 1. **Etude des besoins non fonctionnels**

S’ajoutent aux besoins fonctionnels mentionnés dans les paragraphes précédents, le site doit garantir des multiples besoins non fonctionnels.

* + 1. **La Sécurité**

La sécurité est la priorité numéro un de ce projet et de mon plan de développement. Le système doit obéir à plusieurs règles de sécurité qui peuvent être résumées par les points suivants :

* Tous les utilisateurs doivent vérifier leurs emails.
* Tous les mots de passe et les jetons d’accès seront cryptés dans le système avec une période d’expiration.
* Utilisation du protocole HTTPS uniquement pour garantir le cryptage des données de bout en bout.
  + 1. **L’Ergonomie**

Conjointement à la tâche primordiale qui m’a été confiée qui est le développement ce projet, je dois jouer le rôle de graphiste. Je suis constamment tenu de mettre en place la meilleure interface et la meilleure expérience utilisateur pour l’application. Selon les normes UX/UI, l’utilisateur doit avoir une interface soignée avec un flux organisée, un accès facile à toutes les fonctionnalités du site, en plus d’être à la fois responsive, afin de maintenir l’amélioration de la productivité promise qui est l’une des valeurs les plus importantes de **Med Espoir-Shop.**

* + 1. **La Performance**

Le temps de réponse est un point de défi pour moi, l’application web doit réagir dans un délai précis (<2 secondes) quel que soit l’action de l’utilisateur.

1. **Méthodologie de travail**

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d’un planning organisationnel tout au long de son cycle de vie. C’est ainsi qu’est apparue la notion de méthodologie.

* 1. **Etude comparative des méthodes**

Dans le tableau 1 ci-dessous, j’ai effectué une étude comparative entre les méthodologies afin de faire le bon choix qui répond aux besoins du processus de mon projet.

Tableau 1: étude comparative des méthodes

|  |  |  |
| --- | --- | --- |
| **Thème** | **Approche classique** | **Approche agile (Scrum)** |
| Cycle de vie | Cascade. | Itératif et incrémentale. |
| Planification | Prédictive ou l’identification des besoins est établie au préalable et définitive. | Adaptive et toujours ouverte à tout ajustement en fonction des changements survenus. |
| Documentation | Produite en quantité importante. | Réduite au strict nécessaire. |
| Equipe | Une équipe avec des ressources spécialisées dirigées par un chef de Project. | Une équipe responsabilisée ou l’initiative et la communication sont privilégiés. Le Product Owner et le scrum master dirigent le produit et le processus mais l’engagement et la responsabilité incombent à toute l’équipe Scrum. |
| Qualité | Contrôle qualité à la fin du cycle de développement.  Le client découvre le produit fini. | Un contrôle qualité précoce et permanent.  Le client visualise les résultats tôt et fréquemment. |
| Changement | Résistance voire opposition au changement. Processus lourd de gestion des changements acceptés. | Accueil favorable au changement inéluctable, intégré dans le processus. |
| Suivi lancement | Mesure de la conformité aux plans initiaux. Analyse des écarts. | Le nombre de fonctionnalités implémentées et le travail restant à faire. |
| Gestion des risques | Processus distinct, rigoureux, de gestion des risques. | Gestion du risque intégré dans le processus global. |
| Mesure de succès | Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité. | Satisfaction client par la livraison de valeur ajoutée. |

* 1. **Choix de la méthode**

À la suite de l’étude que j’ai effectuée dans la section précédente, et après une délibération basée sur les besoins du mon projet, j’ai opté pour « **SCRUM** » comme méthode de gestion pour mon projet.

En effet, cette méthode fait partie des méthodes agiles, et figure parmi les plus utilisés dans le monde, ce qui montre son efficacité et son rendement et aussi elle est menée dans un esprit collaboratif et s’adapte aux approches incrémentales.

Elle engendre des produits de haute qualité tout en tenant compte de l’évolution des besoins du client. Elle permet aussi de gérer la qualité en continu et de détecter des problèmes le plus tôt au fur et à mesure, permettant ainsi d’entreprendre des actions correctrices sans trop de pénalités dans les coûts et les délais.

* 1. **Présentation du cadre de processus SCRUM**

Scrum offre un cadre précis et souple, parfait pour les projets innovant ou complexes, cette méthode a pour objectif d’améliorer la productivité des équipes et de favoriser le dialogue entre le client et le prestataire, afin d’optimiser la réussite des projets.

Le principe de Scrum est de développer un logiciel de manière incrémentale en maintenant une liste totalement transparente des fonctionnalités à développer, des demandes d’évolutions ou de corrections à implémenter (Backlog Product). Avec des livraisons très fréquentes, le client reçoit à chaque fois un logiciel avec des fonctionnalités nouvelles et en parfait état de fonctionnement. Pour cela, la méthode s’appuie sur des développements itératifs à un rythme constant d’une durée généralement de 2 à 4 semaines (voir un schéma récapitulatif des constituants de Scrum dans la figure 3).

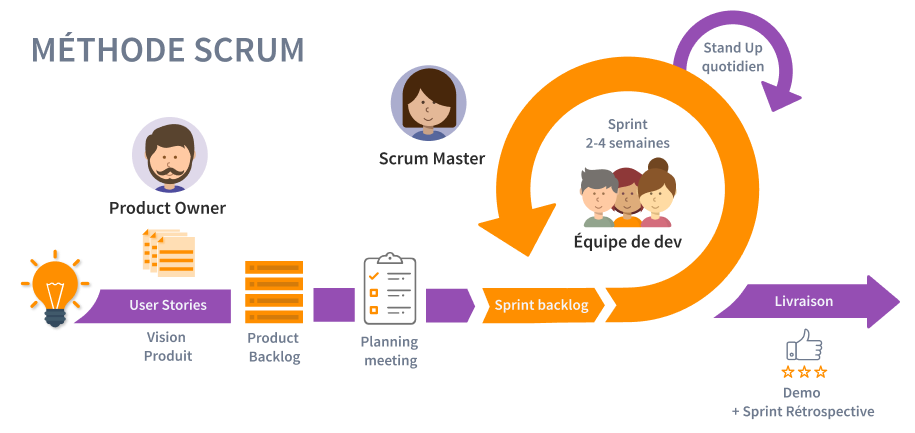


Figure 3: cycle de vie de méthode Scrum

Pour le pilotage du projet l’équipe **TouchLink** utilise « Azure Devos » de Microsoft (figure 4) qui offre plusieurs services pour le management du projet.

Graphical user interface, text

Description automatically generated

Figure 4: Azure DevOps

* + 1. **Les événements de Scrum**
* **Le Sprint :** Le sprint agile représente le cœur de la méthode scrum. Cette qualification lui correspond plutôt bien, puisque tous les développeurs incrémentaux menant petit à petit au produit final du projet sont réalisés au sein du sprint. Un périmètre de développement est défini au début d’un sprint et doit être entièrement réalisé lorsqu’il se termine. Chaque Sprint doit apporter des fonctionnalités supplémentaires à l’application en cours de développement qui doivent être livrées lorsqu’il se termine [1].
* **Le Sprint Planning :** La planification des sprints est une session limitée dans le temps qui prend environ une heure pour chaque semaine de course. Dans une planification rapide, toute l’équipe s’engage à remplir un ensemble de Backlog de produit. Cette convention détermine le Backlog de la course et est basée sur la vitesse ou la capacité de l’équipe et la longueur du sprint [2].
* **Le Sprint Review :** La revue de sprint est une réunion informelle à laquelle participeront l'équipe de développement, le scrum master, le Product Owner et les parties prenantes. L'équipe donne une démonstration sur le produit et déterminera ce qui est fini et ce qui ne l'est pas. Le but de la réunion de Sprint Review est que l'équipe montre aux clients et aux parties prenantes le travail qu'ils ont accompli au cours du sprint et le compare à l'engagement pris au début du sprint.
* **Le Sprint Rétrospective :** La rétrospective de sprint est généralement la dernière chose faite dans un sprint. De nombreuses équipes le feront immédiatement après la revue de sprint. Toute l'équipe, y compris le ScrumMaster et le Product Owner, doit participer. Vous pouvez programmer une rétrospective Scrum jusqu'à une heure, ce qui est généralement tout à fait suffisant[3].
* **Le Daily Meeting :** La réunion quotidienne de mêlée (appelée réunion debout quotidienne) a généralement lieu chaque matin à chaque sprint, le temps dure environ 15 minutes. Dans la pratique agile Scrum, l'équipe tiendra généralement une réunion à l'heure à 9h00 au même endroit[4].
  + 1. **Les rôles dans Scrum**
* **Le Product Owner (PO)** : Le Product Owner ou PO est le responsable de la définition et de la conception d’un produit. Expert de la méthodologie agile, il fait le lien entre la partie métier (bonne vision business) et la partie technique du projet. Il doit porter la vision du produit. Il est l’interface entre l’utilisateur, le Scrum Master et les équipes chargées du développement[5].
* **Le Scrum Master (SM) :** Un Scrum Master est le membre de l'équipe Scrum chargé de favoriser un environnement de travail efficace et productif et de guider les autres pour comprendre les valeurs, les principes et les pratiques de Scrum. Les Scrum Masters ont tendance à être axés sur les personnes, à avoir un niveau élevé d'intelligence émotionnelle et à trouver de la joie à aider les membres de l'équipe à grandir[6].
* **La Développent Team :** Les développeurs sont les personnes de l'équipe Scrum qui s'engagent à créer n'importe quel aspect d'un incrément utilisable à chaque Sprint. Les compétences spécifiques requises par les développeurs sont souvent larges et varient selon le domaine de travail[7].
  + 1. **Les objets de Scrum**
* **Le Product Backlog (PB) :** Un Backlog de produit répertorie et hiérarchise les détails au niveau des tâches requis pour exécuter le plan stratégique défini dans la feuille de route. Le Backlog doit communiquer les prochaines étapes de la liste des tâches de l'équipe de développement au fur et à mesure de l'exécution de la vision d'ensemble de la feuille de route. Les éléments typiques d'un Backlog de produit incluent des histoires d'utilisateurs, des corrections de bogues et d'autres tâches[8].

Le Backlog est une traduction de la façon dont votre équipe livrera la vision décrite sur une feuille de route agile. À bien des égards, il s'agit d'une liste de tâches géante pour votre équipe de développement[8].

* **Le Sprint Backlog :** Le Backlog de sprint est une liste de tâches identifiées par l'équipe Scrum à accomplir pendant le sprint Scrum. Au cours de la réunion de planification du sprint, l'équipe sélectionne un certain nombre d'éléments du Backlog produit, généralement sous la forme d’user stories, et identifie les tâches nécessaires pour terminer chaque user story[9].
* **L’incrément :** L’incrément en scrum représente l’ensemble des éléments « done » du sprint en cours en plus de ceux déjà finalisé dans les sprints précédents.

1. **Elaboration et priorisation du Product Backlog**

Le Product Backlog est l’artefact le plus important de Scrum, il correspond à une liste priorisée des besoins et des exigences du client. Les éléments du Product Backlog, appelés User Stories, sont formulées en une ou deux phrases décrivant de manière claire et précise la fonctionnalité désirée par le client, généralement, dressé sous la forme d’un tableau (voir tableau 5).

Chaque user story a use priority,

H = high, M = medium, L = low

Tableau 2: product backlog ''user stories''

|  |  |  |  |
| --- | --- | --- | --- |
| ID | En tant que | Je souhaite | Priorités |
| 1 | Utilisateur | Être capable de créer un compte | H |
| 2 | Utilisateur | Authentifier via mon login et mot de passe | H |
| 3 | Utilisateur | Être capable de restaurer mon mot de passe | H |
| 4 | Utilisateur | Vérifier mon email | M |
| 5 | Administrateur | Gérer les sous catégories | H |
| 6 | Administrateur | Gérer les produit | H |
| 7 | Administrateur | Modifier les stocks de produit | H |
| 8 | Administrateur | Modifier les remises sur des produits | M |
| 9 | Administrateur | Consulter les utilisateur inscrit | M |
| 10 | Administrateur | Envoyer des courriers à certains utilisateurs | M |
| 11 | Administrateur | Publier et gérer les actualités | H |
| 12 | Administrateur | Suivi toutes les activités des utilisateurs | H |
| 13 | Administrateur | Suivi les statistiques | H |
| 14 | Utilisateur | Cherche pour un produit sans naviguer les catalogue | H |
| 15 | Utilisateur | Laisser un review pour un produit | H |
| 16 | Utilisateur | Filtrer les produits d’un catalogue | H |
| 17 | Utilisateur | Ajouter un produit à ma liste de souhaite | M |
| 18 | Utilisateur | Inscrire au actualités | L |
| 19 | Utilisateur | Ajouter un produit a mon panier | H |
| 20 | Utilisateur | Voir les produits récent | M |
| 21 | Utilisateur | Voir un carrousel qui contient des offres spécial | H |
| 22 | Administrateur | Gérer les produits qui ont des offres spécial | H |
| 23 | Utilisateur | Ajouter un produit à mon panier | H |
| 24 | Utilisateur | Gérer mon panier | H |
| 25 | Utilisateur | Changer la langue principale du site | L |
| 26 | Utilisateur | Je souhaite vérifier les ordres placés | H |
| 27 | Utilisateur | Choisir la méthode de payement | H |
| 29 | Utilisateur | Voir les historiques de mes ordres | H |
| 30 | Utilisateur | Acheter à nouveau un produit | H |
| 31 | Utilisateur | Suivi l’état d’ordres | H |
| 32 | Administrateur | Suivi et gérer l’état d’ordres | H |

1. **Planifications des sprints**

Après clarification des besoins de notre client et élaboration du Product Backlog, celui-ci est ventilé sur la période du stage selon des itérations dites sprints d’après Scrum. La durée permise pour un sprint est de 2 à 4 semaines au bout duquel un incrément potentiellement livrable sera présenté au client dans l’objectif d’avoir son feed-back~~.~~

Le tableau 3 présente le planning des sprints pour Med Espoir Shop.

Tableau 3 : planification des sprints

|  |  |  |  |
| --- | --- | --- | --- |
| SPRINT # | SPRINT NAME | Période | Taches |
| **Sprint 1** | Modules d’Inscription et Authentification | 1mars ->  14mars | * Création d’un compte * Authentification * Vérifier email * Restaurer mot de passe |
| **Sprint 2** | Développement des modules du back-office | 15mars -> 25avril | * Gestion des sous-catégories * Gestion d’un produit * Gestion des actualités. * Gestion des ordres. * Voir les log et activités des utilisateurs. * Développement les interfaces graphiques de notre backoffice. |
| **Sprint 3** | Gestion des catalogues et de recherche | 26avril-> 9mai | * Développer une interface pour le moteur de recherche intègre. * Développement les interface des catalogues |
| **Sprint 4** | Gestion du caddy | 10mai –> 23mai | * Gestion des promotions. * Gestion des paniers |
| **Sprint 5** | Gestion du module de paiement | 24mai -> 7juin | * Intégration d’un système de vente en ligne * Voire historique des ordres |

1. **Représentation schématique du PB par diagramme de cas d’utilisation**

Le diagramme des cas d’utilisation permet d’identifier toutes les fonctionnalités que doit fournir le système. Les figures 5 et 6 ci-dessous, représentent le diagramme des cas d’utilisation global de mon projet impliquant un internaute, un client et l’administrateur.

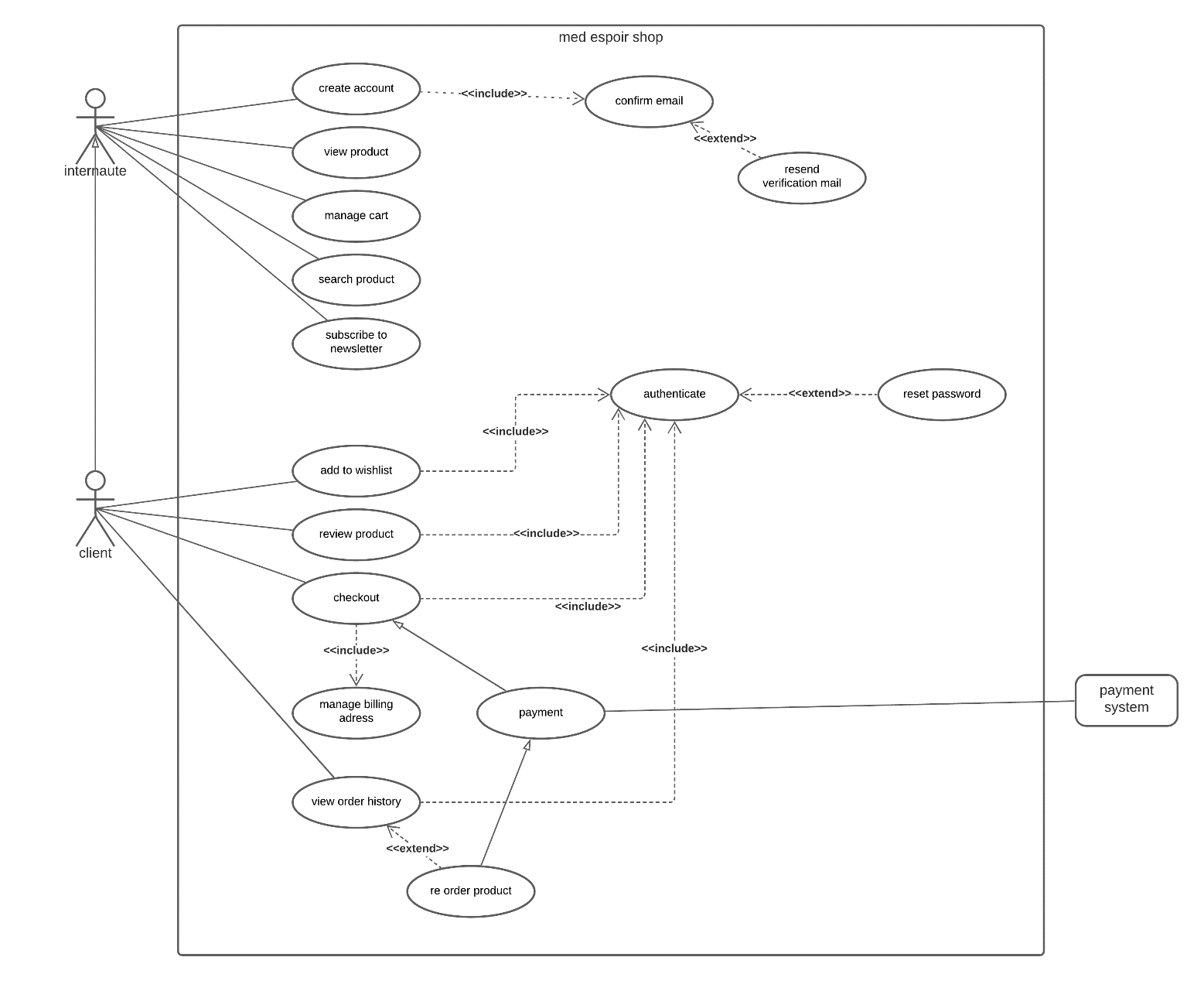
****

Figure 5: digramme de cas utilisation "client"

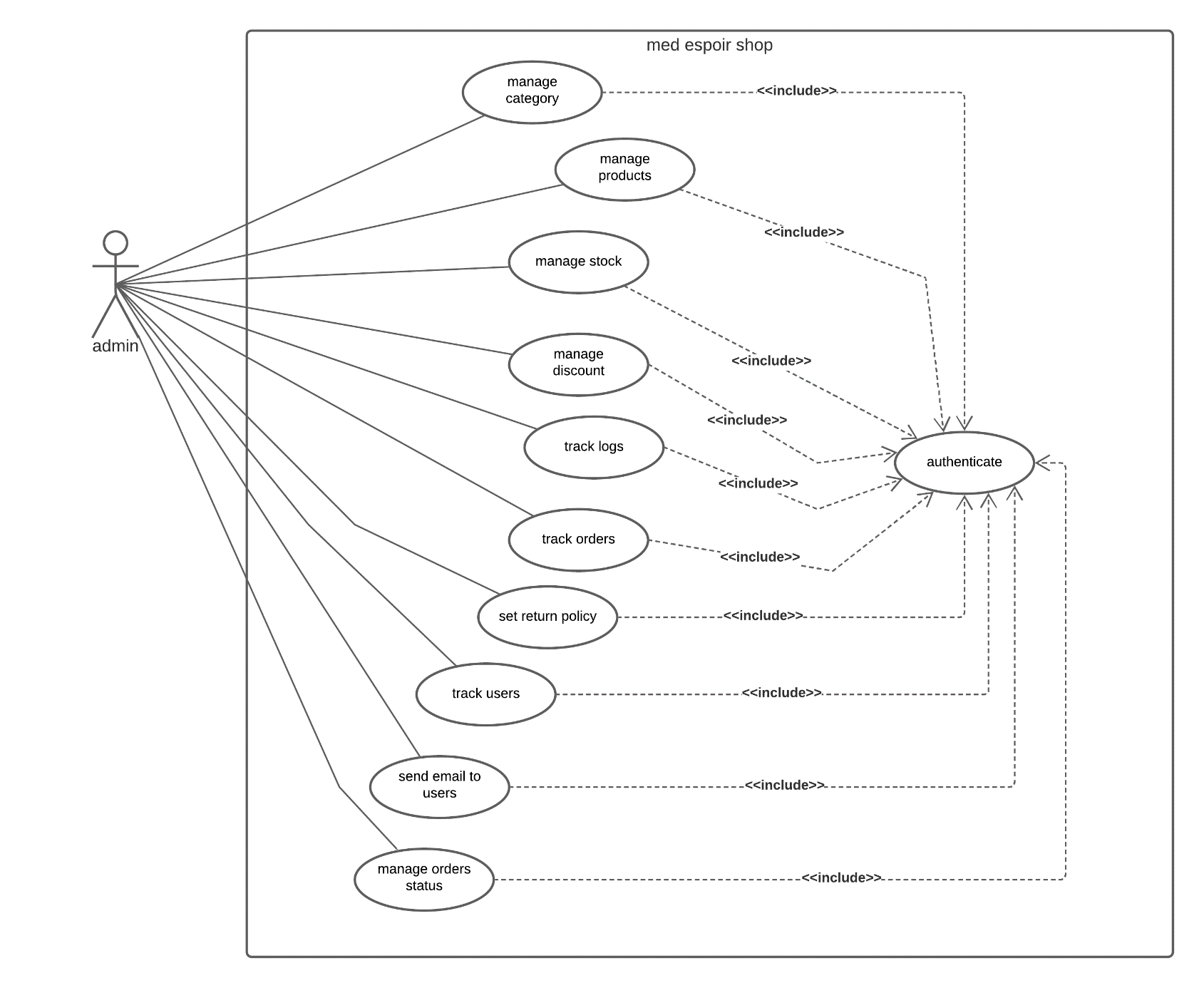
****

Figure 6: digramme de cas utilisation "admin"

1. **Architecture globale du projet**

Dans cette section nous allons présenter les architectures choisies et ses éléments, nous allons concentrer principalement sur l’architecture physique et l’architecture logique.

* 1. **Architecture physique**

L’application Med Espoir-Shop se connecte à un serveur de base de données distant, via internet, afin de récupère les données. Ce qui nécessite à l’intégration d’un serveur web entre l’application client et le serveur base donnée.

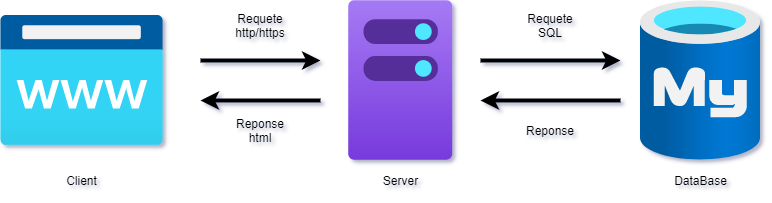
****

Figure 7: architecture physique

* 1. **Architecture logique**

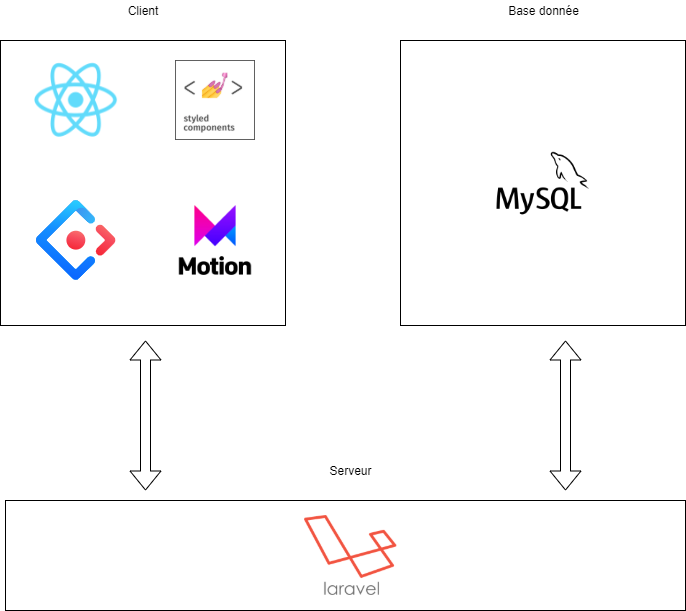
****La vue logique d’une architecture définit les principaux composants d’une architecture sans se soucier des détails physiques.

Figure 8: architecture logique

**Conclusion**

Après avoir décrire les besoins fonctionnels et techniques attendues de mon application et j’ai essayé d’exprimer le fonctionnement de notre système en se basant principalement sur les diagrammes de cas d’utilisation. Je peux ainsi entamer la prochaine étape qui consiste à présenter la phase de conception.

Après avoir décrire les besoins et les acteurs. J’ai réalisé une analyse de l’application en présentant la méthodologie de travail ainsi que le Backlog du produit.

J’ai détaillé de plus l’architecture et l’environnement matériel de notre application ainsi que le diagramme de classe et de cas d’utilisations global.

Dans le chapitre suivant nous avons présenté le module d’inscription et authentification de notre application (sprint 1) et montrer le processus de travail implique dans la réalisation de cet objectif.

**Chapitre 2**

**Sprint1 : Les Modules d’Inscription et Authentification de Med Espoir Shop**

**Introduction**

Le sprint est le noyau de scrum, c’est un intervalle de temps durant lequel un incrément du produit sera réalisé. Avant de commencer un sprint l’équipe doit définir le but envisagé et ce dernier ne doit pas être exprime en terme technique pour qu’il soit compréhensible par le Product Owner. Dans notre cas puisque l’authentification du client est nécessaire au début du passage d’une commande, suivi d’une commande est l’insertion des informations personnelle. Dans ce chapitre je présenterai le module d’inscription qui est une étape très importante par lequel le client obtient le droit de s’identifier.

1. **Backlog du premier sprint**

Le tableau 4 suivant représente le Backlog du premier sprint :

Tableau 4: backlog sprint1- authentification-

|  |  |  |  |
| --- | --- | --- | --- |
| *Id* | *en tant que* | *User stories* | *Taches* |
| 1 | Utilisateur | Je veux m’inscrire | * Développer les interface adéquates * Tester s’inscrire |
| 2 | Utilisateur | Je veux être capable de vérifier l’adresse email de mon compte. | * Développer les fonctionnalités. * Développer les interfaces adéquates. * Tester vérification email. |
| 3 | Utilisateur | Je veux être capable de m’authentifier via mon login et mot de passe. | * Développement des fonctionnalités. * Développement des interface adéquates. * Tester s’authentifier. |
| 4 | Utilisateur | Je veux être capable de restaurer mon mot de passe. | * Développer les interface adéquates. * Tester changer mot de passe. |
| 5 | Admin | Je souhaite interdire aux utilisateurs l’accès à certaines pages | * Développer les fonctionnalités nécessaires. * Tester Restreinte Access |

1. **Analyse**
   1. **Raffinement du cas utilisation du sprint 1**

Dans cette partie nous s’intéresse au raffinement des cas d’utilisation

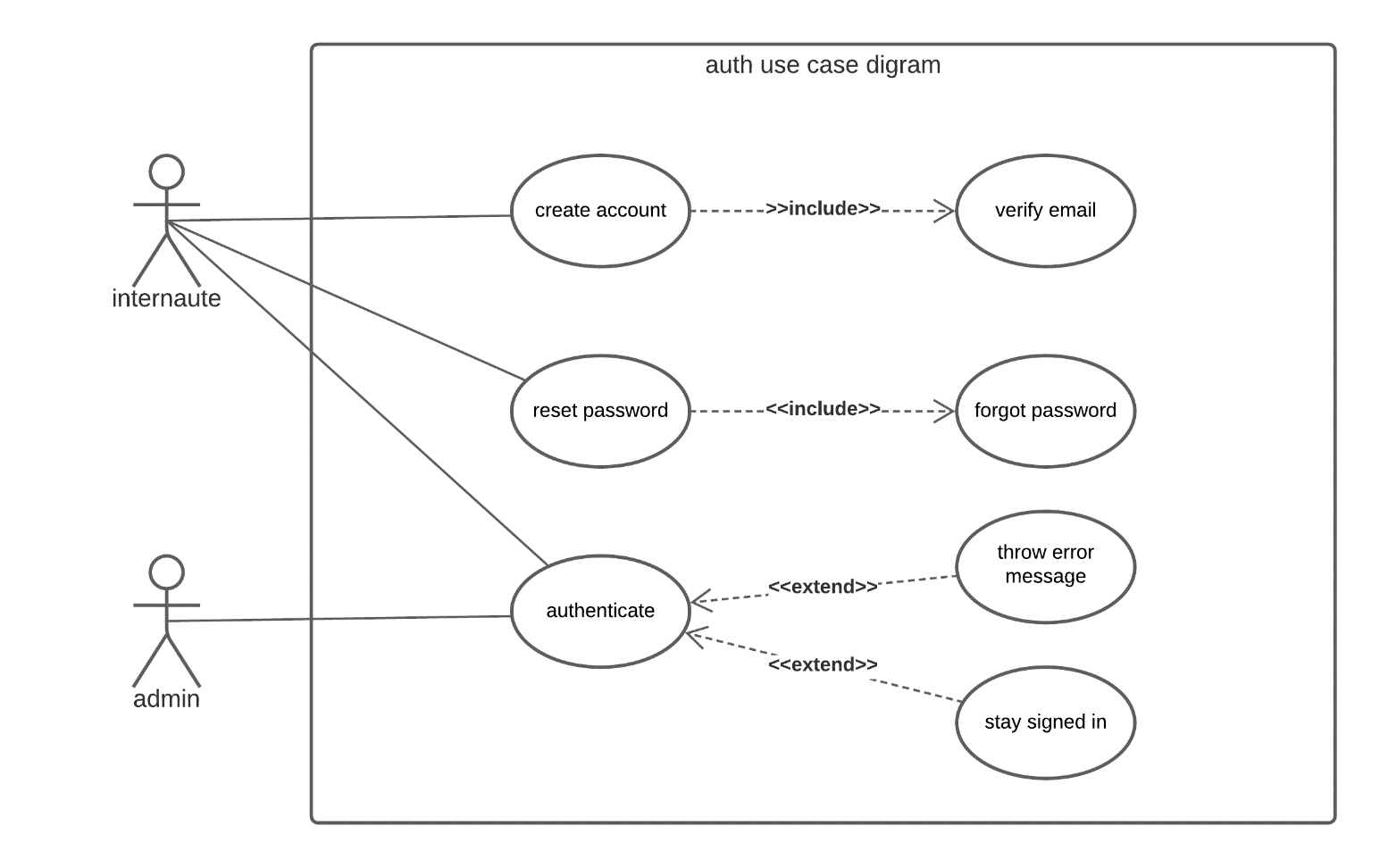


Figure 9: diagramme de cas utilisation Authentification

Le diagramme de cas d’utilisation est une représentation sans illustration ou explication, alors une description textuelle est nécessaire afin de décrire les interactions entre le système et ses acteurs suivant différents chemins qui sont les scenarios prévus du projet.

* + 1. **Cas « inscription »**

Tableau 5 : description textuelle inscription

|  |  |
| --- | --- |
| Acteur : client | |
| Pre condition | **Post condition** |
| L’utilisateur doit avoir une connexion internet | Lorsque l’inscription se termine, un compte utilisateur sera créer |
| Scenario nominal | |
| 1. L’utilisateur clique sur le bouton de création un compte 2. L’application web affiche une interface pour la création du compte 3. L’utilisateur remplir le formulaire correctement 4. Le système vérifie les champs saisis 5. Le compte sera ajoute à la base de données 6. Un courrier électronique sera envoyé au client 7. Le client sera redirecteur vers l’interface home | |
| Scenario alternatif | |
| 1. Un message d’erreur est affiché si l’un des champs est vide ou invalide | |

* + 1. **Cas « authentification »**

Tableau 6 : description textuelle authentification

|  |  |
| --- | --- |
| Acteur : tous les utilisateurs d’application | |
| Pre condition | **Post condition** |
| Le client déjà possède un des identifiants pour se connecter | Une session ouverte contient l’utilisateur authentifie |
| Scenario nominal | |
| 1. Le client se dirige vers l’interface d’authentification 2. Le client saisit son login et son mot de passe correctement 3. L’utilisateur a le choix d’activer « keep me signed in » 4. Le système vérifie le login et le mot de passe 5. Le système va redirecteur l’utilisateur selon son rôle 6. L’utilisateur a l’accès a des certaines routes selon son rôle | |
| Scenario alternatif | |
| 1. Le système affiche un message d’erreur si les donnes ne sont pas validées. 2. Une page 404 s’affiche si l’utilisateur n’est pas authentifié ou autorise | |

* + 1. **Cas « restaure mot de passe »**

Tableau 7: description textuelle de restaurer mot de passe

|  |  |
| --- | --- |
| Acteur : client | |
| Pre condition | **Post condition** |
| L’utilisateur a un compte mais oublier le mot de passe pour compléter l’authentification | Le client est authentifié avec son nouvelle mot de passe |
| Scenario nominal | |
| 1. L’utilisateur a oublié son mot de passe 2. L’utilisateur clique sur le bouton restore mot de passe 3. Le client sera redirigé vers une Interface pour entrer son email 4. L’utilisateur entre son email correctement 5. Un courrier électronique qui contient un lien pour restaurer le mot de passe 6. Le lien ouvre une interface pour entrer le nouveau mot de passe 7. Le mot de passe a été change 8. L’utilisateur sera automatiquement authentifié lors de changement de mot de passe | |
| Scenario alternatif | |
| 1. Un message d’erreur indique que l’email n’est pas valide 2. Le champs mot de passe n’est pas valide | |

* + 1. **Cas « vérifier email »**

Tableau 8: description textuelle de vérifier email

|  |  |
| --- | --- |
| Acteur : client | |
| Pre condition | **Post condition** |
| L’utilisateur a des restreint sur la commande car son email n’est pas vérifié | Un compte avec un email vérifier pour bien identifier le client |
| Scenario nominal | |
| 1. Lors de création de compte un courrier électronique de vérification sera envoyé 2. Le client clique sur le lien dans l’email 3. L’utilisateur est redirigé vers home est son email a était vérifier | |
| Scenario alternatif | |
| 1. L’utilisateur peux renvoyer le message de vérification est sera valide pour 60min | |

1. **Conception**

Au niveau de cette partie du chapitre, nous avons exposer le diagramme des classes de conception et les diagrammes de séquence détaillés qui clarifie l’interaction entre l’utilisateur et le système.

* 1. **Diagramme de classes du sprint 1**

Une présentation graphique du diagramme des classes du sprint 1 dans figure 10 :

Diagram

Description automatically generated

Figure 10: diagramme de classe sprint 1

* 1. **Diagrammes des séquences**

Les diagrammes de séquences sont la représentation graphique des [interactions](https://fr.wikipedia.org/wiki/Unified_Modeling_Language) entre les [acteurs](https://fr.wikipedia.org/wiki/Acteur_(UML)) et le système selon un ordre chronologique dans la formulation [Unified Modeling Language](https://fr.wikipedia.org/wiki/Unified_Modeling_Language)[10].

* + 1. **Cas « inscription »**

La figure 11 présente le diagramme de séquence du cas utilisation « inscrire » :

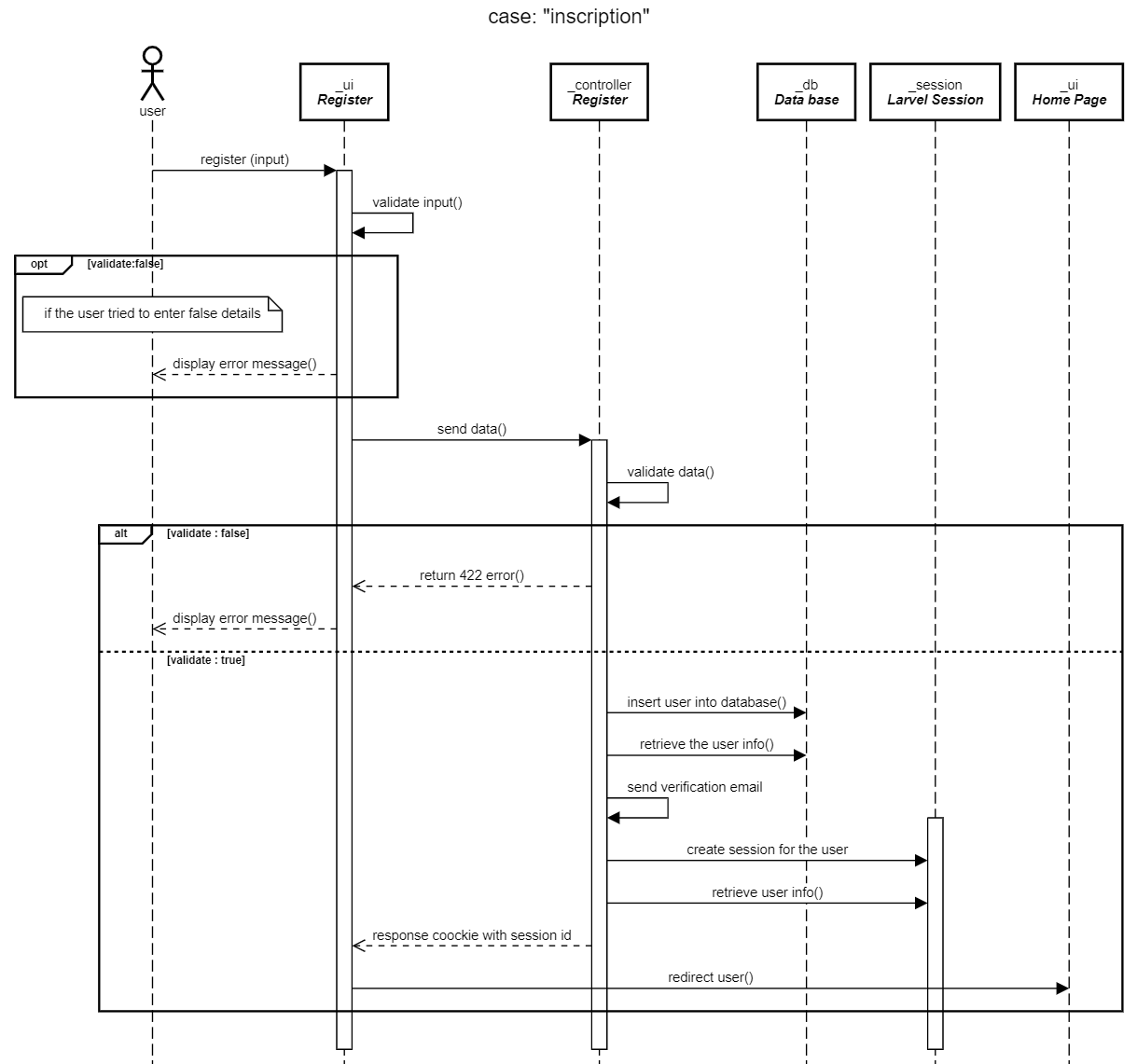
****

Figure 11: diagramme de séquence du cas utilisation s'inscrire

* + 1. **Cas « authentifier »**

La figure 12 présenter le diagramme de séquence du cas utilisation « authentifier » :

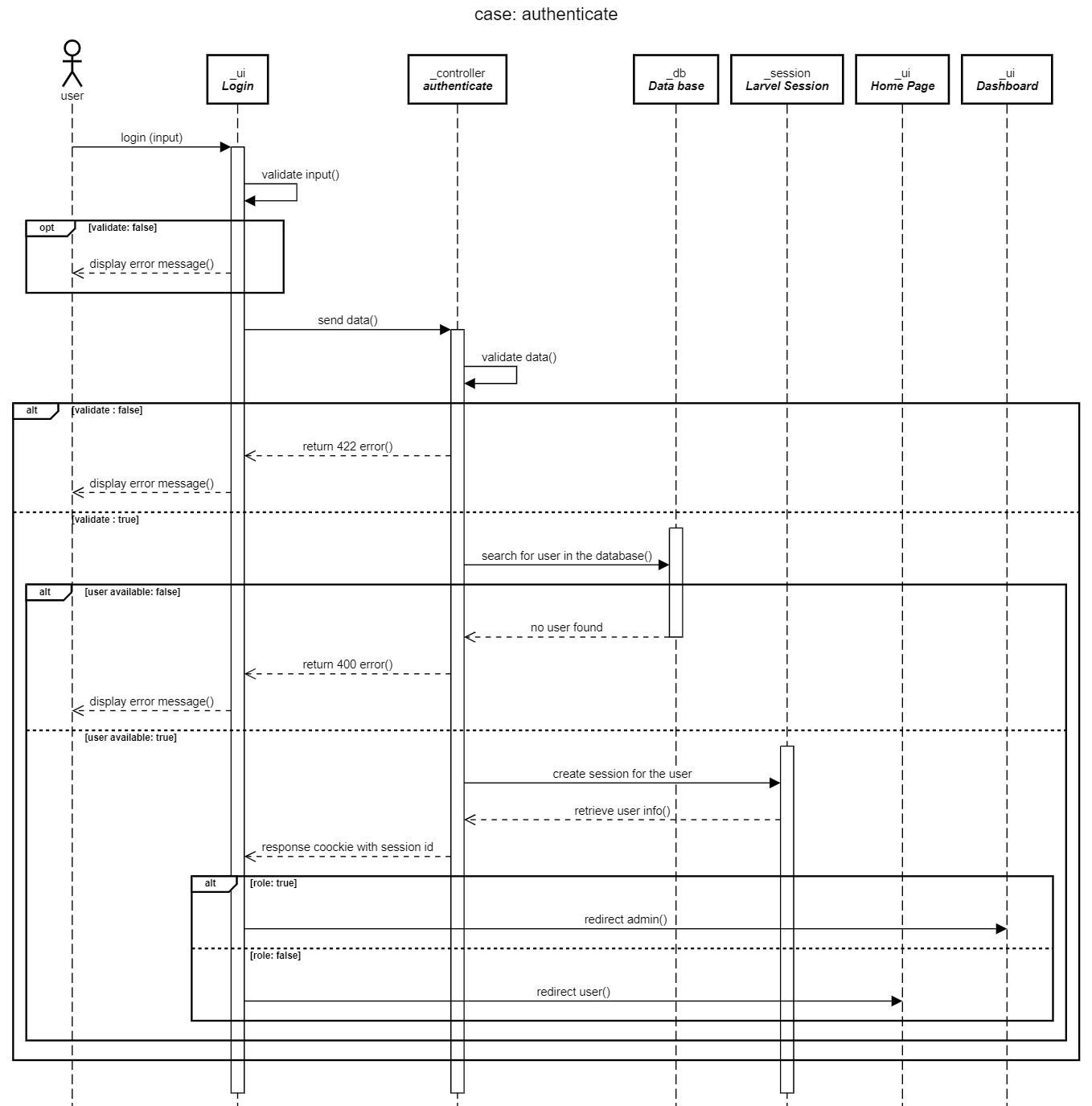


Figure 12: diagramme de séquence du cas utilisation authentifier

* + 1. **Cas « restaurer mot de passe »**

La figure 13 présenter le diagramme de séquence du cas utilisation « restaurer mot de passe » :

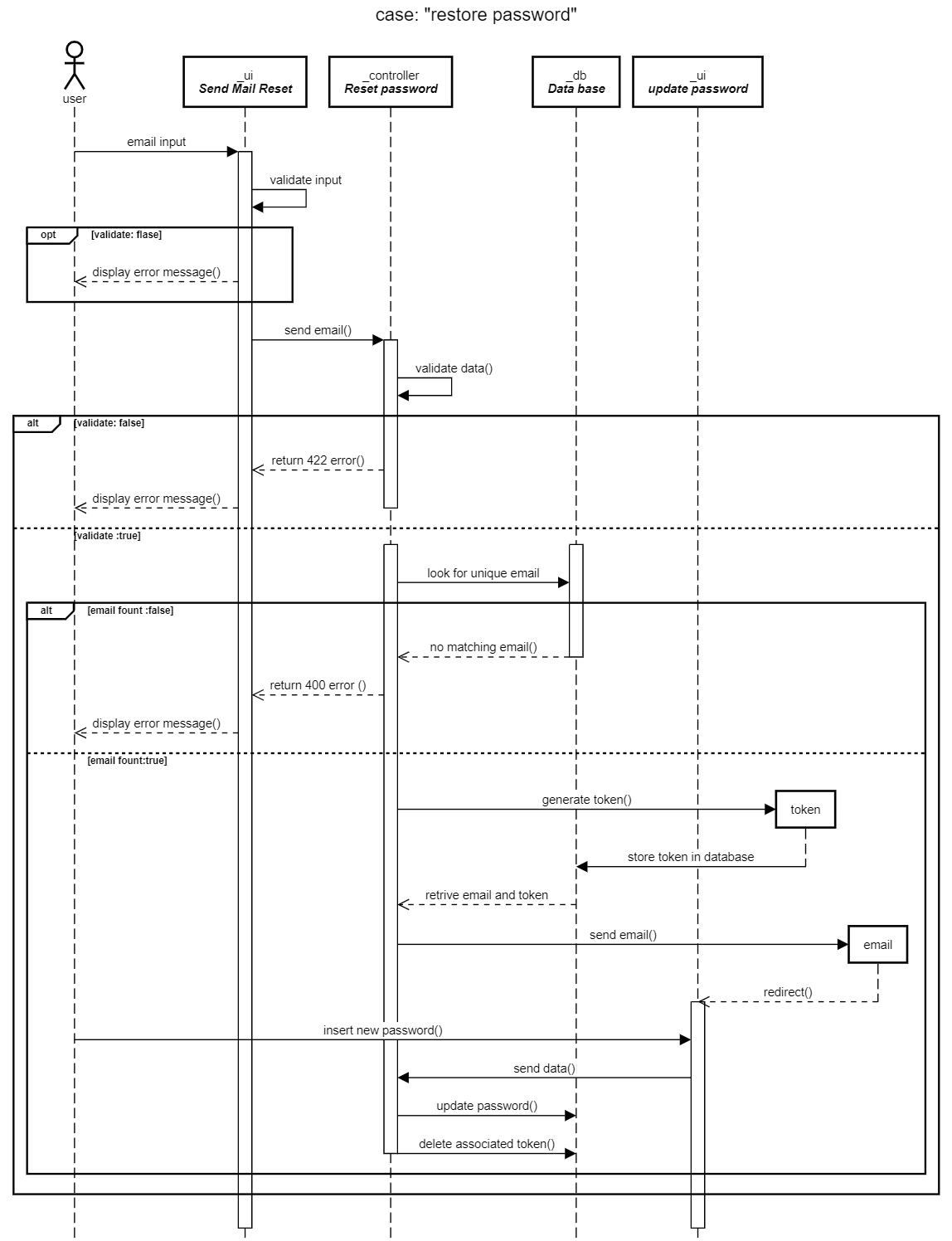


Figure 13: diagramme de séquence du cas utilisation restaurer mot de passe

1. **Environnement de développement** 
   1. **Technologies**

Tableau 9: les technologies de web

|  |  |
| --- | --- |
| Technologies | Description |
| Node JS 14 LTS | Node JS est une plateforme logicielle libre en JavaScript orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8, la librairie libuv pour sa boucle d’évènements, et implémente sous License MIT les spécifications CommonJS. |
| React JS 17 | React est une bibliothèque JavaScript efficace, flexible et déclarative, maintenue par Facebook et communauté indépendante de développeurs et d’entreprise. React permet de composer des interfaces client riches et complexes utilisant de petite pièce isolée appelées composants. |
| Laravel 8 | Laravel est un Framework web libre écrit en PHP respectant le principe mvc et entièrement développer en programmation orientée Objet. Laravel est distribué sous License MIT. |

* 1. **Environnement logiciel**

Dans cette partie, je vais citer les outils que j’ai utilisé tout au long de mon projet dans le tableau ci-dessous.

Tableau 10: les logiciels

|  |  |
| --- | --- |
| Outil | Description |
| NPM 6.14 | NPM est la gestionnaire de package officiel de node.JS. Depuis la version 0.6.3 de node JS, npm fait partie de l’environnement et est donc automatiquement installe par défaut.  NPM fonctionne avec un terminal et gère les dépendances pour une application |
| Composer 2 | Composer est la gestionnaire de package de PHP  Composer fonctionne avec une terminale et gère les dépendances pour une application. |
| XAMPP | Un ensemble de logiciels permettant de mettre en place un serveur web local. |
| Vs Code | Visual Studio Code est un éditeur de code extensible développé par Microsoft avec « électron JS » pour Windows, Mac Linux.  Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de syntaxe, la complétion intelligente du code et l’intégration du GIT. |
| Thunder client | Une extension Vs code et un fork de Postman qui nous permet d’envoyer et recevoir les requête http pour tester l’API |
| GitHub | GitHub est centré vers l’aspect social du développement est base sur GIT, il offre nombreuses fonctionnalités. La possibilité de gestion les versions, les page wiki, l’intégration continues et la livraison continue. |
| LucidChart | Un logiciel de digramme en ligne pour créer des diagrammes de processus, UML et organigrammes. |

* 1. **Environnement matériel**

Nous allons parler dans cette partie de la machine utilisée pour faire la réalisation de projet, le tableau ci-dessous contient les caractéristiques de cette machine.

Tableau 11: pc specs

|  |  |
| --- | --- |
| Série | LEGION Y540 |
| Marque | Lenovo |
| RAM | 16 Go |
| Disque Dur | 256 Go SSD, 1To HDD |
| CPU | Intel core I7-9750H |
| Carte Graphique | Nvidia GeForce GTX 1650 |
| Système exploitation | Windows 10 |

1. **Sprint review**

Il s’agit d’une démo de deux heures devant le client étant donné que notre sprint est de deux semaines. Toutes les interfaces sont dument testées. Un test désigne une procédure de vérification partielle d’un système. Lest tests de vérifications visent à vérifier que ce système réagit de la façon prévue par les développeurs et conforme aux besoins du client.

* 1. **Cas « authentification »**

Graphical user interface, application

Description automatically generated

Figure 14: cas authentification version web

A picture containing text, screenshot, monitor, black

Description automatically generated

Figure 15: cas authentification version responsive -mobile-

* 1. **Cas « inscription »**

Graphical user interface, application

Description automatically generated

Figure 16: cas inscription version web

Graphical user interface, application

Description automatically generated

Figure 17: cas inscription version responsive -mobile-

* 1. **Cas « mot de pass oublier »**

Dans cette cas quand l’utilisateur clicke sur le lien de restaurer mote de pass dans son courier, il sera rederiger vers ce lien ou il ya un token et son email dans l’**url**



Figure 18: cas restaurer mot de passe version web

A screenshot of a computer

Description automatically generated with medium confidence

Figure 19: cas restaurer mot de passe version responsive -mobile-

**Conclusion**

A la fin de cet sprint, nous avons produit un increment ayant les fonctionalite neccesasaie pour repondre a l’objectif de notre sprint. Les utlisateur peuvent créer des comptes, se connecter, verifier son email et restaurer le mot de passe au cas il l’oublie.

Dans le chapitre qui suit, nous allons concentrer sur les developpment du module back-office qui nous permet une gestion des informations et contenu do notre application.

# **Chapitre 3**

# **Sprint 2 : Developpement des modules du Back-Office de MedEspoir Shop**

**Introduction**

En appliquant le meme principe que le chapitre precedent nous avons decide que le but du deuxieme sprint est le suivant « developpement des modules de back-office ».

Le module back-office ou tableau de bord est un outil qui nous permet d’organise, stocke et affiche les infomarions importantes de notre application.

1. **backlog du deuxieme sprint**

Le tableau 4 suivant représente le Backlog du deuxième sprint :

Tableau 12: backlog sprint 2-dashboard-

|  |  |  |  |
| --- | --- | --- | --- |
| ID | En tant que | User sotries | Taches |
| 1 | Administrateur | Je peux faire la gestion des sous-catégorie | * Développer les fonctionnalités adéquates pour sous-catégories. * Développer les interfaces |
|  | Administrateur | Je peux faire la gestion des produits | * Développer les fonctionnalités crud pour les produits * Développer les interfaces |
|  | Administrateur | Je peux modifier les stocks des produits | * Développer les fonctionnalités adéquates * Développer les interfaces |
|  | Administrateur | Je peux modifier les remises sur des produits | * Développer la fonctionnalité de modification remise |
|  | Administrateur | Je peux définir si le client peut retourner le produit on non | * Développer la fonctionnalité adéquates * Développer les interfaces |
|  | Administrateur | Je peux envoyer des courriers au client | * Développer la fonctionnalité de mailing * Développer les interfaces |
|  | Administrateur | Je veux être capable de publier des actualités | * Développer les fonctionnalités adéquates * Développer les interfaces |
|  | Administrateur | Je veux suivi les activités des utilisateur | * Développer les fonctionnalités adéquates * Développer les interfaces |

1. **Analyse**
   1. **Raffinement du cas utilisation du sprint 2**

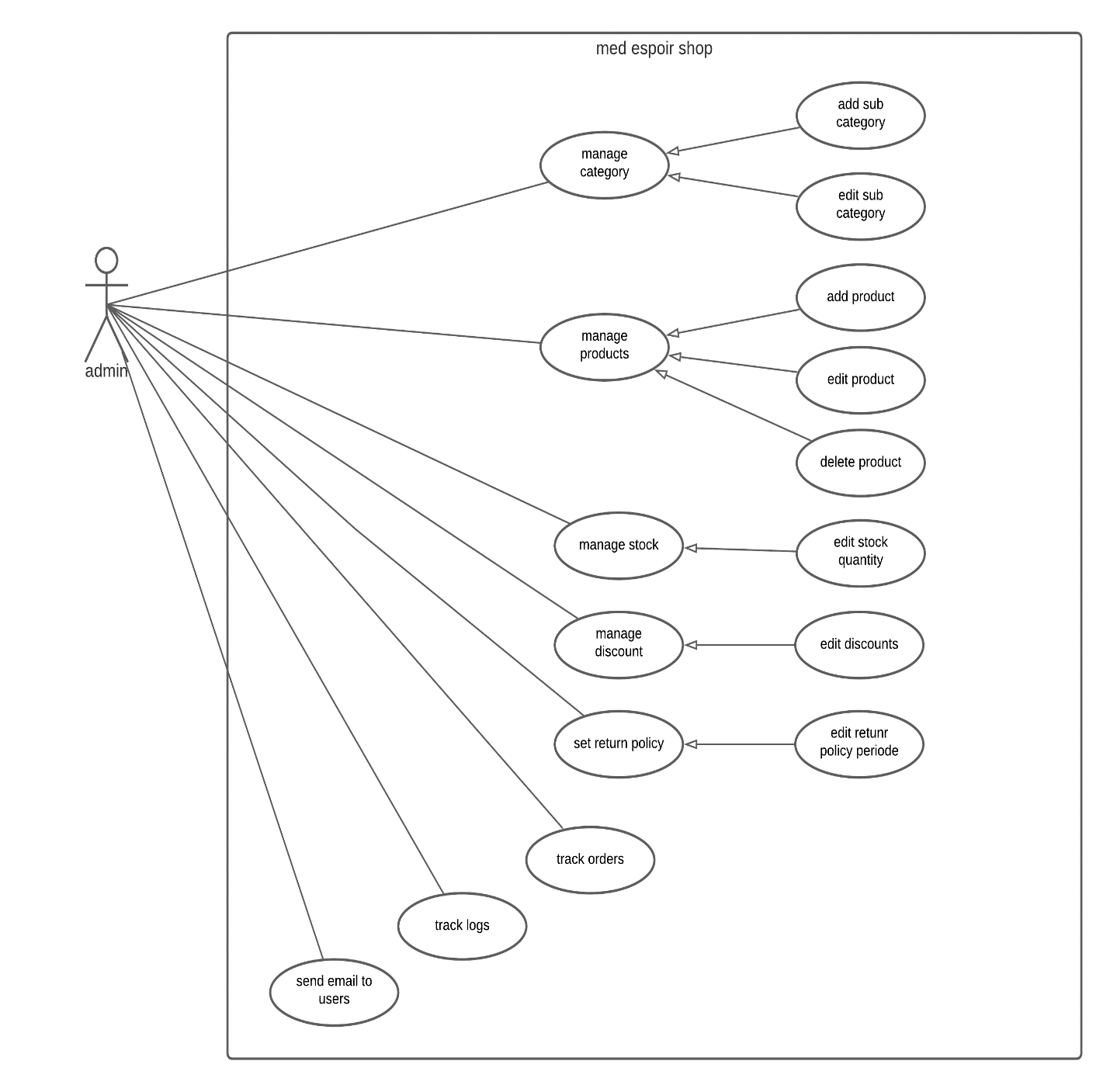
****

Figure 20: diagramme de cas utilisation sprint 2

**Remarque** : certains cas d’utilisation sont identiques et ont le même fonctionnement. Afin d’éviter la redondance, nous avons choisi de ne pas détailler tous les cas.

* + 1. **Cas « gestion des produits »**

Ce cas permet à l’administrateur d’ajouter, supprimer, modifier un produit après l’ajout d’une sou catégorie, ce dernier a le même fonctionnement que le cas « gestion catégorie », « gestion des stocke », « gestion des remise », « définir politique de retour ».

* + - 1. **Ajouter un produit**

Tableau 13: description textuelle pour ajoute un produit

|  |  |
| --- | --- |
| Acteur : administrateur | |
| Pre condition | **Post condition** |
| Liste des produits affichée. | Lorsque l’ajout se termine un produit associer à une sous-catégorie sera ajouter à la liste. |
| Scenario nominal | |
| 1. L’admin clique sur le bouton ajouter. 2. Le système le dirige vers le formulaire adéquat. 3. L’admin remplit les informations d’un produit et l’associe a une sou catégorie. 4. Le système vérifie les champs saisis. 5. Le système affiche qu’un produit a été ajouter. 6. L’admin est dirigé vers la liste des sous-catégories. | |
| Scenario alternatif | |
| * 1. Si la sous-catégorie que l’admin désire n’est pas valable il doit le créer d’abord.   2. Un message d’errer si le champ obligatoire est vide ou invalide | |

* + - 1. **Modifier un produit**

Tableau 14: description textuelle pour modifier un produit

|  |  |
| --- | --- |
| Acteur : administrateur | |
| Pre condition | **Post condition** |
| Liste des produits affichée contient un ou plusieurs produits. | Le produit a été modifié avec succès. |
| Scenario nominal | |
| 1. L’admin cherche le produit qu’il désire modifie. 2. L’admin clique sur l’Icon de modification. 3. Le système le dirige vers le formulaire adéquat remplir par la valeur du produit à modifier. 4. L’admin change ce qu’il désire dans le champ. 5. L’admin vérifie les modifications en cliquant sur le bouton modifier. 6. Le système réaffiche la liste avec le produit modifier. | |
| Scenario alternatif | |
| * 1. Si la valeur de recherche est invalide aucun produit est affiché | |

* + - 1. **Supprimer un produit**

Tableau 15: description textuelle pour supprimer un produit

|  |  |
| --- | --- |
| Acteur : administrateur | |
| Pre condition | **Post condition** |
| Liste des produits affichée contient un ou plusieurs produits. | Le produit a été supprimé. |
| Scenario nominal | |
| 1. L’admin select ou cherche le produit qu’il désire supprimer. 2. L’admin clique sur l’Icon de suppression. 3. Le système réaffiche la liste sans le produit supprimé. | |
| Scenario alternatif | |
| * 1. Si la valeur de recherche est invalide aucun produit est affiché | |

* + 1. **Cas « contacter client »**

Ce cas permet à l’administrateur d’envoyer un courrier électronique au client.

Tableau 16: description textuelle pour contacter un client

|  |  |
| --- | --- |
| Acteur : administrateur | |
| Pre condition | **Post condition** |
| Listes des emails du client | Le courrier électronique est envoyé au client |
| Scenario nominal | |
| 1. L’admin sélection l’email du client qui désire lui contact à partir d’une liste. 2. L’admin rempli les champs du sujet et contenu 3. L’admin clique sur le bouton envoyer 4. Le système va envoyer le courrier au client | |
| Scenario alternatif | |
| * 1. Un message d’erreur s’il y’a un champ vide. | |

* + 1. **Cas « suivi activités »**

Ce cas permet à l’administrateur de consulter son activité et les changements qu’il a fait ainsi que les activités des clients.

Tableau 17: description textuelle pour le suivi des activités

|  |  |
| --- | --- |
| Acteur : administrateur | |
| Pre condition | **Post condition** |
| Aucune action a été faite.  La liste des activités et vide. | Une liste contient les activités et les changements faite par les utilisateurs. |
| Scenario nominal | |
| 1. Admin ajout un produit ou faire une action qui affecte la base de données. 2. Le système va enregistrer l’action. 3. L’admin peut consulter ses activités. | |
| Scenario alternatif | |
| 1. L’admin peut consulter les actions faites par autre utilisateur | |

1. **Conception**

Au niveau de cette partie du chapitre, nous avons exposer le diagramme des classes qui clarifie l’interaction entre les entités de notre application.

La figure 21 représente le diagramme de class du sprint 1

Diagram, schematic

Description automatically generatedFigure 21: diagrammes de class sprint 1

1. **Sprint Review**

Il s’agit d’une démo de quatre heures devant le client étant donné que notre sprint est de quatre semaines. Toutes les interfaces sont dument testées. De façon prévue par les développeurs et conforme aux besoins du client.

* 1. **Cas « gestion des produits »**

La figure ci-dessous représente la liste des produits

Graphical user interface, application

Description automatically generated

Figure 22: liste des produits

* **Ajout d’un produit**

Graphical user interface, application

Description automatically generated

Figure 23: cas ajouter un produit

* **Supprimer un produit**

Après la clique sur l’icône de suppression le système va retourner la même liste sans le produit supprimer

Table

Description automatically generated

Figure 24: cas supprimer un produit

* **Modifier un produit**

Graphical user interface, text, application, Teams

Description automatically generated

Figure 25: cas modifier un produit

1. **Sprint rétrospective**

Dans cet incrément et pendant le développement de notre api, la méthode **put** qui ’est responsable à modifier la donne existante retourne le code 201 mais les donnes ne sont pas modifiées.

Après quelque essaie de debugging est des recherches sur la documentation official, GitHub est stack over flow, nous avons résolu le problème.

Le problème était que PHP ne comprend pas les inputs dans **méthode** **put** à partir de « multipart/form-data » donc dans notre « **requeste headers** » nous avons changé le   
‘content-type’ en 'application/x-www-form-urlencoded;charset=utf-8' et avec l’aider du package npm qui s’appelle qs (query string), nous serons capables des faire des modifications.

**Conclusion**

nous avons produit un increment ayant les fonctionalite neccesasaie pour repondre a l’objectif de notre sprint.

Dans le chapitre suivant, nous allons concentrer sur les developpment du module front-office qui contient trois sprint commencant par la gestion des catalogue est de recherche.

# **Chapitre 4 : Sprint 3,4,5 : Developpement des Modules du Front-Office de MedEspoir Shop**

**Introduction**

Ce release contient trois sprints comme nous l’avons déjà mentionné dans le chapitre précédent. A la fin de ce chapitre nous devons avoir une application qui répond à la majorité des besoins .

Dans ce chapitre nous présenterons les sprints trois, quatre et Cinque, ou le troisième sprint est réservé pour la gestion du catalogue et module de recherche, le quatrième sprint sera pour la gestion du caddy et module des promotions, le cinquième sprint est dédié pour l’intégration de modules de paiement.

1. **Sprint 3 « Gestion du catalogue et module de recherche »**

En achevant ce sprint, le client doit avoir des produits organisés par ses catalogue et chaque catalogue permet le filtrage du produit selon le prix et les sous-catégories aussi il peut faire un rechercher généralise directement sans l’aide de filtrer toutes les sous catégories.

* 1. **Backlog du troisième sprint :**

Tableau 18: backlog sprint 3-gestion catalogue et recherche-

|  |  |  |  |
| --- | --- | --- | --- |
| Id | En tant que | User stories | Taches |
|  | Utilisateur | Voir les produits classifier selon un catalogue | * Traiter les fonctionnements nécessaires. * Développer les interface adéquate de chaque catalogue. |
|  | Utilisateur | Fileter les produits d’un catalogue par sous-catégories. | * Développer les fonctionnements de filtrage. |
|  | Utilisateur | Ajouter un produit a ma liste de souhaite | * Traiter les fonctionnements de liste de souhaite. |
|  | Utilisateur | Chercher pour un produit avec le moteur de recherche généralise | * Traiter les fonctionnements nécessaires. * Développer l’interface de recherche. |
|  | Utilisateur | Laisser des commentaires et feedback sur un produit | * Traiter les fonctionnements nécessaires pour Product review * Développer les interfaces adéquates. |
|  | Utilisateur | Inscrire aux actualités | * Traiter les fonctions adéquates |

* 1. **Analyse**
     1. **Raffinement du cas utilisation du sprint 3**

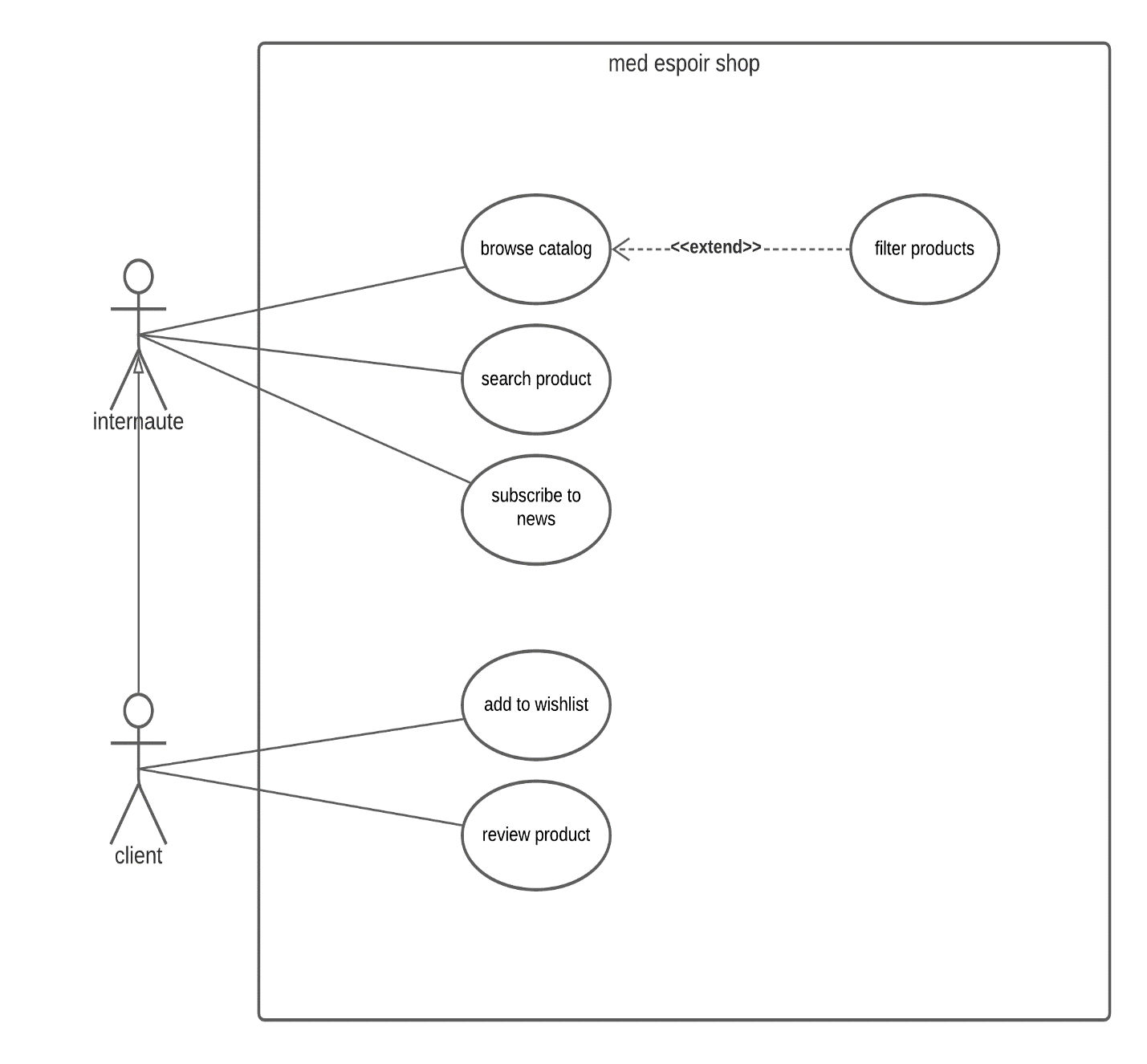
****

Figure 26: diagramme de cas utilisation sprint 3

* **Cas « filtre produit »**

Ce cas serve l’expérience utilisateurs en regroupant de façon logique le contenu, et pour limiter les nombres de catégories, par grouper les produits similaires sous le même catalogue.

Tableau 19: description textuelle de filtre produit

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| L’utilisateur | Une liste des produits selon l’option de filtre que l’utilisateur a sélectionné |
| Scenario nominal | |
| 1. L’utilisateur choisie un catalogue parmi le catalogue en top 2. Le système va afficher tous les produits de ce catalogue 3. L’utilisateur choisie le mode de filtrage qu’il désire par prix ou par sous-catégories 4. Le système affiche les produits correspondants. | |
| Scenario alternatif | |
| * 1. S’il n’est pas un produit associer au mode de filtrage le système afficher un message pour informer l’utilisateur | |

* **Cas « recherche générale »**

Ce cas permit à l’utilisateur d’affecter une recherche générale sur tous les produits par nom, catégorie et sous-catégories

Tableau 20: description textuelle de recherche générale

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| L’utilisateur entrain d’utiliser l’application | Une page s’affiche avec le produit sélectionné à partir de recherche |
| Scenario nominal | |
| 1. L’utilisateur clique sur le bouton de recherche dans la barre de navigation 2. Le system ouvre un modèle pour faire la recherche 3. L’utilisateur remplit le champ de recherche 4. L’utilisateur va afficher des suggestions qui ont des mots clés rassembles à la valeur de recherche. 5. L’utilisateur clique sur le produit désiré 6. Le système va diriger l’utilisateur a une page pour afficher toutes les informations du produit sélectionné | |
| Scenario alternatif | |
| * 1. Un message est affiché si le produit n’existe pas | |

* **Cas « ajouter au wishlist »**

Ce cas permet à l’utilisateur d’ajouter un produit a sa liste de souhaite

Tableau 21: description textuelle ajouter au wishlist

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| Utilisateur doit être authentifié. | Une liste qui contient les produits que l’utilisateur est intéressé à. |
| Scenario nominal | |
| 1. L’utilisateur clique sur l’icône de cœur pour ajouter un produit a la liste 2. Le système va créer une liste de souhaite et ajouter le produit sélectionner 3. L’utilisateur peut supprimer un produit de la liste | |
| Scenario alternatif | |
| * 1. Un message informatif si l’utilisateur va ajouter le même produit. | |

* **Cas « review produit »**

Ce cas permet à l’utilisateur de faire un ou plusieurs commentaires sur un produit.

Tableau 22: description textuelle review produit

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pres condition | **Post condition** |
| Utilisateur doit être authentifié | Un produit évalué avec des commentaires |
| Scenario nominal | |
| 1. L’utilisateur peut commenter sur un produit 2. L’utilisateur peut modifier ou supprimer ses commentaires 3. Le système affiche le produit avec son évaluation et commentaires | |

* 1. **Conception**
     1. **Diagramme de classes du sprint 3**

La figure 27 représente le diagramme de classe de sprint 2

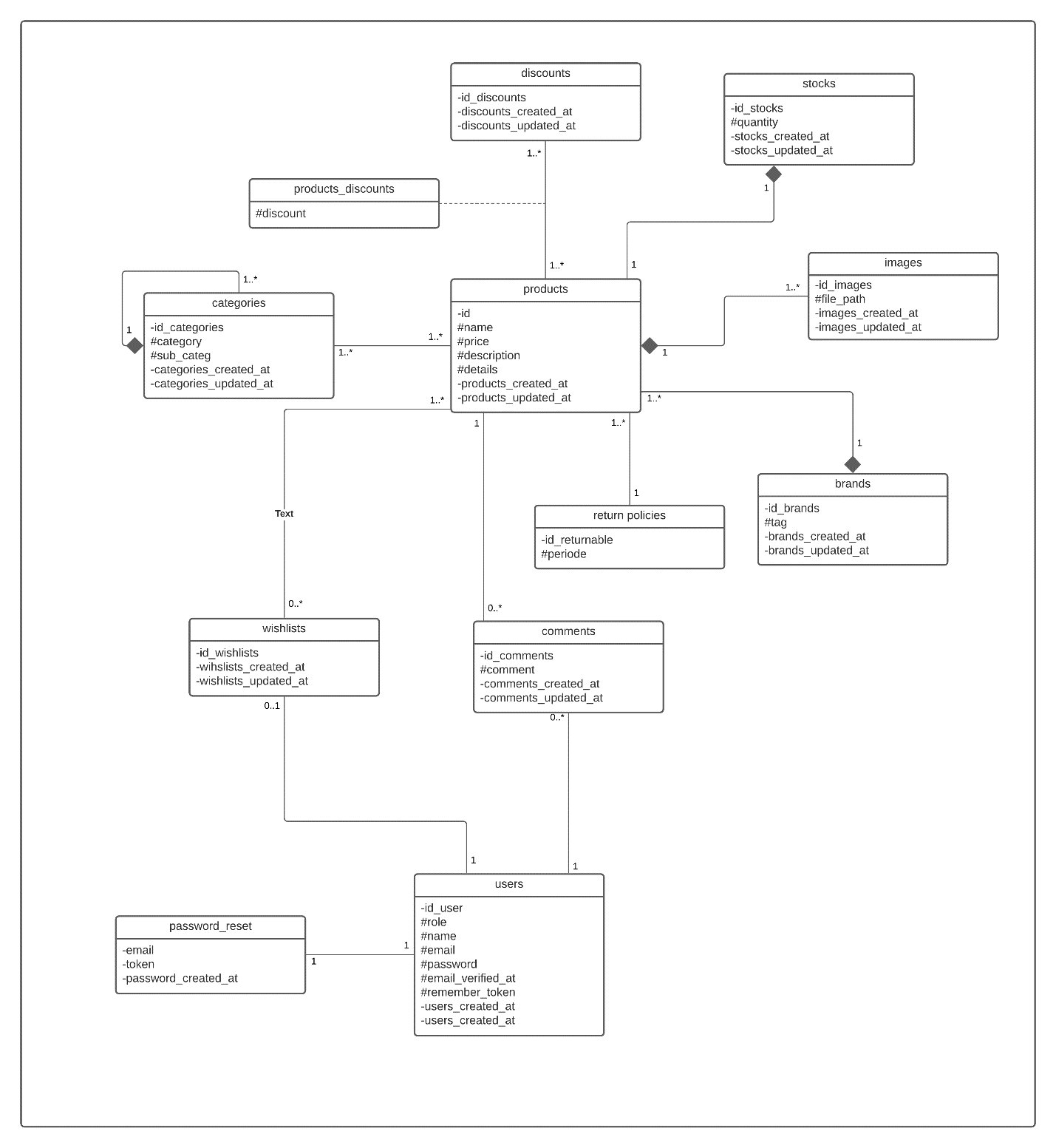


Figure 27: diagramme de classe sprint 3

* + 1. **Diagramme de séquence sprint 3**

La figure 28 représente comment la rechercher générale d’un produit est effectuer

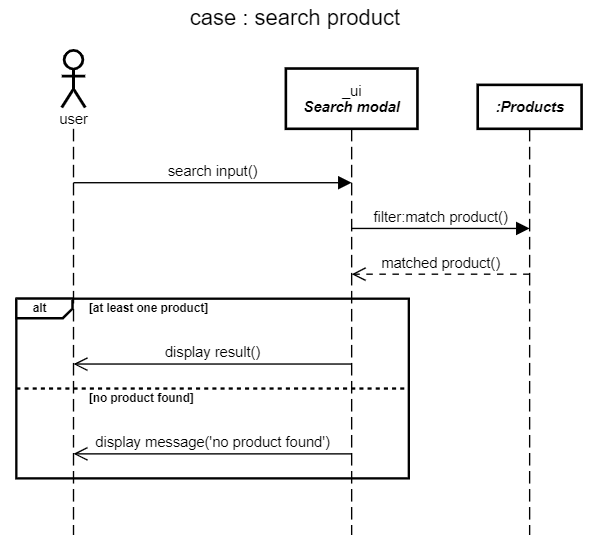


Figure 28: diagramme de séquence de recherche global

1. **Sprint review**

Il s’agit d’une démo de deux heures devant le client étant donné que notre sprint est de deux semaines. Toutes les interfaces sont dument testées. De façon prévue par les développeurs et conforme aux besoins du client.

* **Cas « filtre produit »**

Graphical user interface

Description automatically generated

Figure 29: filtre produit sous-catégorie

* **Cas « recherche générale »**

La figure 30 représente le résultat lorsque l’utilisateur chercher à un produit

Graphical user interface, application

Description automatically generated

Figure 30: cas recherche générale

Lorsqu’on clique sur le produit désire après effectue la recherche global le système nous dirige vers une page pour voir toutes les information ce produit.

Graphical user interface, application

Description automatically generated

Figure 31: detail du produit

* **Cas « review produit »**

Graphical user interface, application

Description automatically generated

Figure 32: commenter sur un produit

* **Cas « ajouter a wishlist »**

La figure 33 représente comment l’utilisateur va ajouter un produit a la liste de souhaite.

Graphical user interface, application, website

Description automatically generated

Figure 33: cas ajoute produit à wishlist

**Remarque** : l’utilisateur clique sur l’icône cœur pour ajouter ce produit a sa liste de souhaite.

1. **Sprint 4 « Gestion du caddy»**

Les caddies sont les bridges entre le shopping et l’achat, il est donc extrêmement important d’avoir le meilleur fonctionnement de caddie sur notre application web.

Pendant cet incrément, le client doit avoir une gestion de panier efficace, il peut ajouter, modifier est supprimer des produits avant de procéder à vérifier ses ordres.

* 1. **Backlog du quatrième sprint**

Tableau 23: backlog sprint 4 -gestion de caddy-

|  |  |  |  |
| --- | --- | --- | --- |
| ID | En tant que | User stories | Taches |
| 1 | Utilisateur | Je souhaite être capable d’ajouter des produits à mon panier | * Traiter le cas d’ajout un produit * Traiter le cas ou produit déjà existe. |
| 2 | Utilisateur | Je souhaite être capable de gérer panier | * Supprimer un produit de panier * Modifier la quantité de chaque produit |

* 1. **Analyse** 
     1. **Raffinement du cas utilisation du sprint 4**

La figure 34 représente le cas d’utilisation pour la gestion d’un caddy

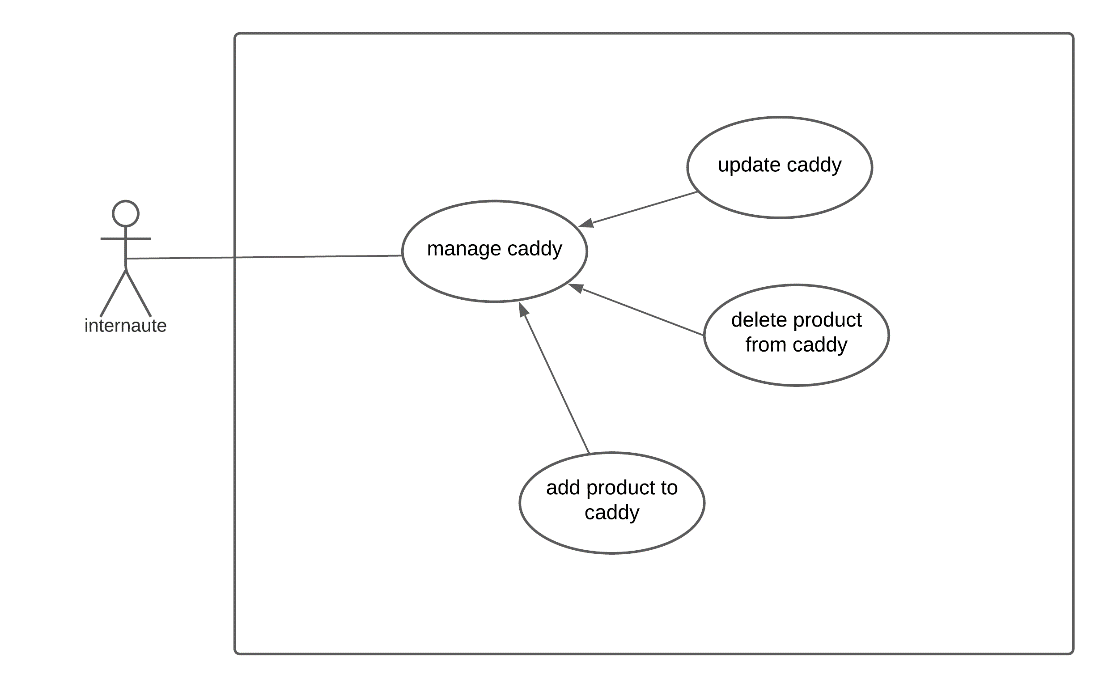


Figure 34: raffinement de cas utilisation du sprint 4

Le diagramme de cas d’utilisation est une représentation sans illustration ou explication, alors une description textuelle est nécessaire pour bien exprimer le fonctionnement de caddy.

* **Cas « ajout un produit  au panier»**

Tableau 24: description textuelle cas ajout produit au panier

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| Le caddy est vide | Le caddy contient les produits ajouté |
| Scenario nominal | |
| 1. L’utilisateur choisi un produit 2. L’utilisateur clique sur le bouton ajouter au panier 3. Le système vérifie que le produit procède un (id, prix) 4. Le système ajout le produit au panier 5. Le système affiche le prix et nombre total des éléments de caddy | |
| Scenario alternatif | |
| * 1. Si un produit existe déjà dans le caddy le système va incrémenter automatiquement la quantité de produit. | |

* **Cas « supprimer un produit de panier »**

Tableau 25: description textuelle cas supprimer produit de panier

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| Le caddy contient des produit | Un produit a été supprimer de panier |
| Scenario nominal | |
| 1. Le système vérifie l’état des produits 2. Le système afficher les produits valables dans le caddy 3. L’utilisateur choisi le produit qu’il désire supprimer 4. L’utilisateur clique sur le bouton supprimer 5. Le système va supprimer le produit et refaire le calcul montant | |
| Scenario alternatif | |
| * 1. Si un produit dans le caddy n’existe plus il sera automatiquement retiré de caddy | |

* **Cas « modifier quantité du produit »**

Tableau 26: description textuelle cas de modification de panier

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| Le caddy contient un produit au minimum | Changement des éléments des caddy |
| Scenario nominal | |
| 1. L’utilisateur se dirige vers panier 2. Le système affiche les produits dans le panier 3. L’utilisateur va augmenter ou diminuer la quantité pour un ou plusieurs produits 4. Le système va recalculer les montant total base sur les changements faite par l’utilisateur | |
| Scenario alternatif | |
| * 1. Si un la quantité d’un produit est égale à 0 le système va automatiquement supprimer le produit de caddy. | |

* 1. **Conception**

La figure 35 est un diagramme de séquence qui explique les interactions entre les utilisateurs de notre application et le système de gestion de caddy.

Ce diagramme explique tous les cas de gestion de caddy « ajout produit », « supprimer produit » et « modifier quantité du produit » qui est exprimé en deux étapes un pour l’augmentation de quantité et l’autre pour la diminution de quantité.

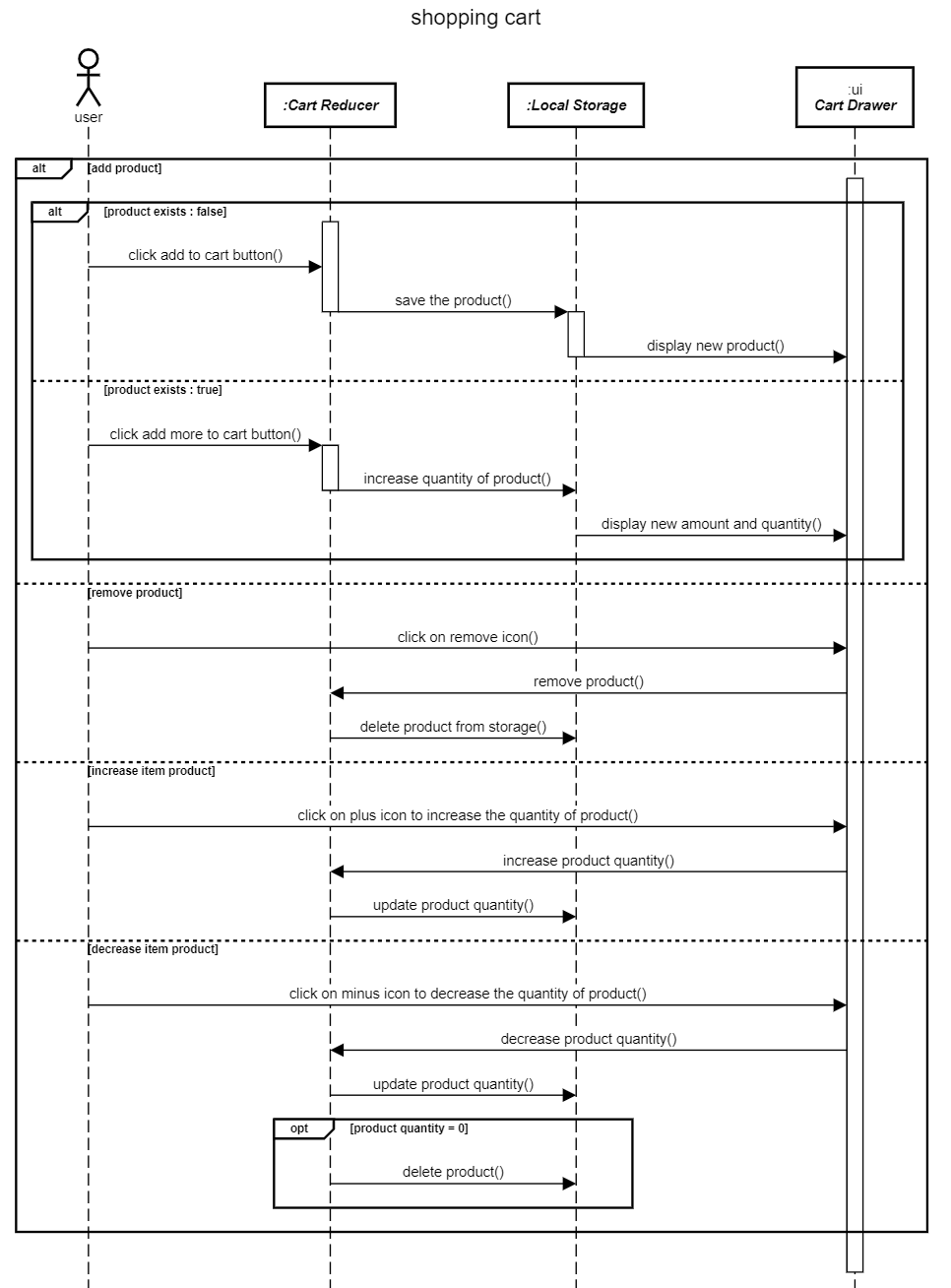


Figure 35: diagramme de séquence sprint 4 -gestion caddy-

* 1. **Sprint review**

Il s’agit d’une démo de deux heures devant le client étant donné que notre sprint est de deux semaines. Toutes les interfaces sont dument testées.

* **Cas « ajout un produit au panier »**

Dans ce cas le caddy est vide, et le bouton nous instruire de faire un ajout au panier.

Graphical user interface, application, Teams

Description automatically generated

Figure 36: cas ajout au panier -caddy vide-

Après la clique sur le bouton le produit a été ajouter au caddy est le bouton a été change et nous instruire maintenant d’insérer plus car le produit est déjà . Si on clique plus qu’un seul fois la quantité du produit va augmenter.

Graphical user interface, application, Teams

Description automatically generated

Figure 37: cas ajout au panier -produit existe-

Graphical user interface, application, Teams

Description automatically generated

Figure 38: cas ajout au panier -diffèrent produit-

Si nous choisisse un autre produit le bouton nous instruite d’ajouter au panier car ce produit n’existe pas dans le caddy

* **Cas « supprimer un produit de panier »**

Dans ce cas nous voulons de supprimer le dernier produit ajouté, if suffit de cliquer sur l’icône de corbeille, el le produit sera supprimé.

Graphical user interface, application

Description automatically generated

Figure 39: cas supprimer un produit de panier

* **Cas « modifier quantité de produit »**

Graphical user interface, application

Description automatically generated

Figure 40: cas modifier la quantité de produit

Comme si expliquer dans le diagramme de séquence la clique sur l’icône plus va augmenter les quantités de produit, et le moins va les diminuer , si la quantité est égale à zéro le produit sera supprimer de caddy.

A picture containing graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated

Figure 41: drawer du caddy -version mobile-

L’interface de caddy est responsive et facile à utiliser, nous avons un indicateur qui contient la quantité totale des tous les produits et un bouton qui indique le montant total à payer. Ce dernier et notre portail par la sprint suivante

1. **Sprint 5  « intégration des modules de paiement en ligne »**

Le paiement en ligne est tout simplement un échange d'argent par système électronique. Il s'agit des paiements que l'on réalise sur Internet ou via des réseaux de télécommunications, générés à partir soit d'un ordinateur, soit d'un téléphone mobile[11].

Dans ce sprint nous allons intégrer quelques modules de paiement en ligne, comme PayPal et Strippe qui offre des api flexible et facile à implémenter.

* 1. **Backlog du cinquième sprint**

Tableau 27: backlog sprint 5-integration des modules de paiement en ligne

|  |  |  |  |
| --- | --- | --- | --- |
| Id | En tant que | User stories | Taches |
| 1 | Utilisateurs | Je souhaite vérifier les ordres placés |  |
| 2 | Utilisateur | Je souhaite choisir la méthode de paiement. | * Développer fonctionnement de paiement * Développer l’interface adéquates. |
| 3 | Utilisateur | Voir les historiques de mes ordres | * Développer les interface et fonctionnalités adéquates |
| 5 | Utilisateur | Suivi l’état d’ordres | * Développer les fonctionnements et les interface adéquates |
| 6 | Administrateur | Suivi et gérer l’état d’ordres | * Développer les fonctionnalités * Développer les interface adéquates |

* 1. **Analyse**
     1. **Raffinement du cas utilisation du sprint 5**

La figure 42 représente le raffinement du cas utilisation du sprint 5

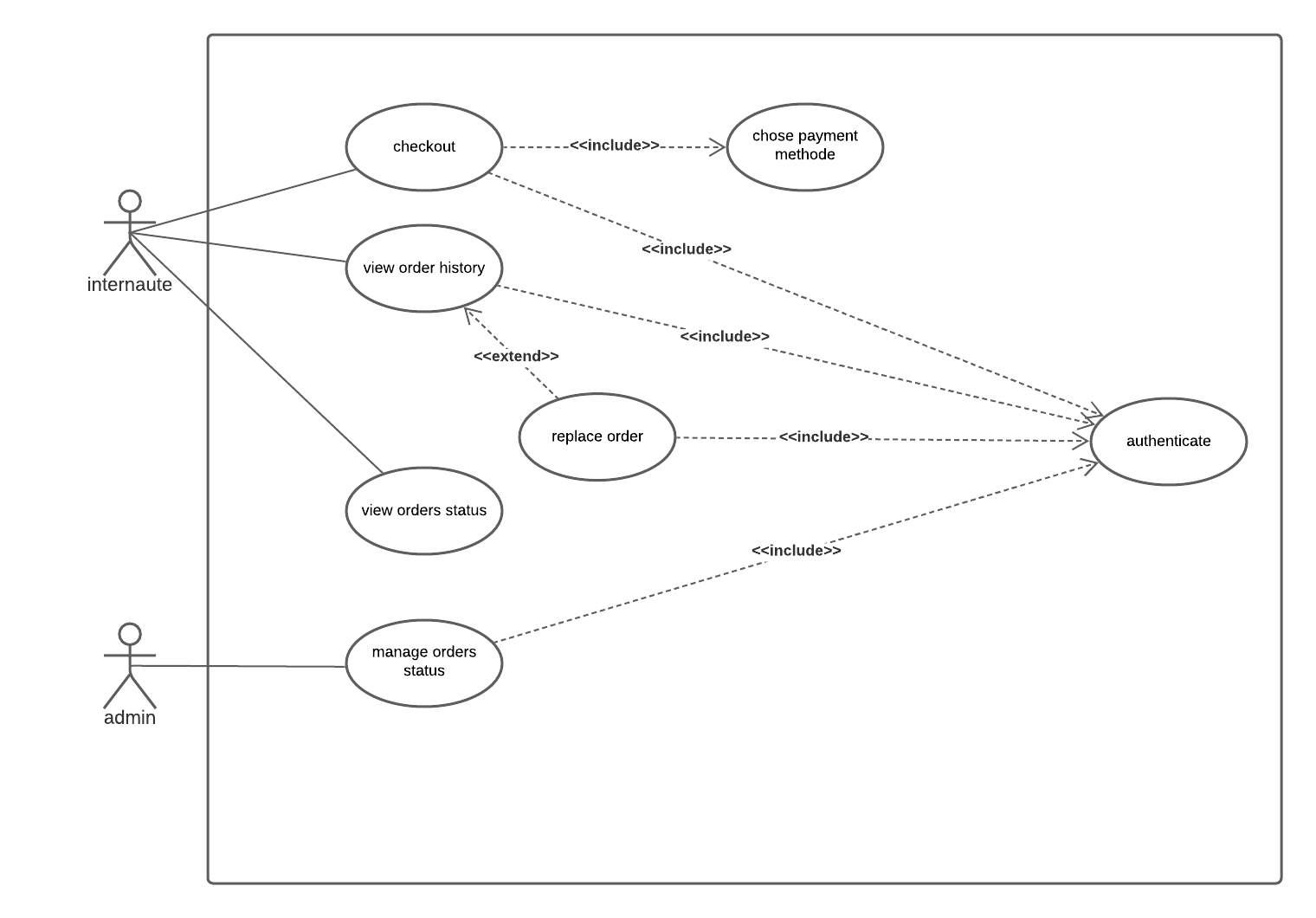


Figure 42:raffinement de cas utilisation du sprint 5

Pour bien exprimer ce diagramme de cas d’utilisation une description textuelle est nécessaire afin de décrire les interactions entre le système et ses acteurs

* **Cas « vérifier les ordres placés »**

Tableau 28: sprint 5 - description textuelle de vérifier les ordres placés

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| L’utilisateur doit être authentifier, et avoir au moins un produit dans son caddy | Le montant totale du caddy |
| Scenario nominal | |
| 1. L’utilisateur vérifie si les produits qu’il veut est valide 2. L’utilisateur confirme le caddy 3. Le système dirige l’utilisateur a page de paiement | |
| Scenario alternatif | |
| Même si l’utilisateur est connecté mais le caddy est vide il ne peut pas accéder au page de paiement | |

* **Cas « Choisir la méthode de paiement »**

Notre application fournit 3 méthodes de paiement , deux seront faites en ligne avec carde bancaire fournit par stipe ou un compte PayPal

Tableau 29: sprint 5-descriprion textuelle de choisir méthode de paiement

|  |  |
| --- | --- |
| Acteur : client | |
| Pre condition | **Post condition** |
| L’utilisateur doit être authentifier | L’utilisateur choisi la méthode de paiement et paye son montant. |
| Scenario nominal | |
| 1. L’utilisateur entre ses donnée et numéro du carte / ou utilise PayPal pour la rapidité de paiement 2. L’utilisateur paye son montant 3. Le système vérifie la transaction 4. L’utilisateur va recevoir un email contient son ordre | |
| Scenario alternatif | |
| 2.1. Un message d’erreur si la carte est invalide ou y’a pas assez de solde dans PayPal | |

* **Cas « voire les historiques et recommander l’ordres »**

Tableau 30: sprint 5 - description textuelle de voir historique est recommander

|  |  |
| --- | --- |
| Acteur : utilisateur | |
| Pre condition | **Post condition** |
| L’utilisateur doit être authentifier | Une liste contient les historiques d’achat de cet utilisateur |
| Scenario nominal | |
| 1. L’utilisateur Access a son profile 2. L’utilisateur peut voir les anciens ordres qu’Il l’a acheté 3. L’utilisateur peut recommander les anciens ordres | |
| Scenario alternatif | |
| L’utilisateur n’a pas encore fait un achat donc la liste des historiques est vide | |

* **Cas « suivi est géré état ordres »**

Ce cas est peux être gérer par l’utilisateur et l’administrateur

Tableau 31: sprint 5-description textuelle de client suivi son ordre

|  |  |
| --- | --- |
| Acteur : client | |
| Pre condition | **Post condition** |
| Un ordre a été enregistrer | Un statut indique l’état de livraison du ordres |
| Scenario nominal | |
| 1. L’utilisateurs dirige vers l’historique d’ordre pour inspecter ses derniers achats 2. L’utilisateurs peut annuler sa commande si le statut d’ordres est non reçu 3. Si l’utilisateurs a reçu son ordre il peut changer sa statue 4. Après le reçu d’ordres l’utilisateurs peux le retourner | |
| Scenario alternatif | |
| Le retour du produit n’est pas valide si la politique de retour est désactivée dans le jour de validation de la commande | |

Tableau 32: sprint 5-description textuelle d'administrateur suivi les ordres

|  |  |
| --- | --- |
| Acteur : administrateur | |
| Pre condition | **Post condition** |
| Un nouvel ordre a été créer | Un statut indique l’état de livraison du ordres |
| Scenario nominal | |
| 1. L’administrateur de site va inspecter l’état de chaque ordre 2. Le système va notifier l’administrateur s’il y’a des nouveaux ordres 3. L’administrateur peut suivi le revenu à partir de page statistique 4. L’administrateur peut annuler une demande 5. Lorsque l’administrateur annule une ordre un courrier sera envoyé à l’utilisateur | |

* 1. **Conception**
     1. **Diagramme de class du sprint 5**

La figure 43 présente le diagramme de classe de notre application

Diagram, schematic

Description automatically generated

Figure 43: diagramme de class sprint 5

* 1. **Sprint review**

Il s’agit d’une démo de deux heures devant le client étant donné que notre sprint est de deux semaines. Toutes les interfaces sont dument testées. Et le client peut payer en ligne.

* **Cas « choisie méthode de paiement**

Graphical user interface, text, application

Description automatically generated

Figure 44: cas "choisir la méthode de paiement"

Dans ce cas on le client va choisir la méthode de paiement compatible

* **Mode de paiement « Stripe »**

Graphical user interface, application

Description automatically generated

Figure 45: méthode de paiement stripe

* **Mode de paiement « PayPal »**

Graphical user interface, text, website

Description automatically generated

Figure 46: méthode de paiement paypal

Nous avons effectué un test de paiement avec paypal sandbox

Graphical user interface, text, application, email

Description automatically generated

Figure 47: transaction monétique avec paypal

**Conclusion**

Dans ce chapitre nous avons étudier le module de front office qui contient trois sprints, sprint trois pour la gestion de recherche et catalogue, sprint quatre pour la gestion des caddy et sprint Cinque pour l’intégration des modules de paiement en ligne.

# Conclusion Générale

Dans le cadre de notre projet de fin d’étude de la licence en technologie de l’informatique : développement web et multimédia au sien de la société MedEspoir. Nous avons conçu et développé un site de vente en ligne des produit « **MedEspoir Shop** » , nous avons essayé tout au long de notre travail de construire notre application incrément par incrément en utilisant la méthodologie Scrum.

Nous avons commencé dans un premier lieu par comprendre le contexte général de notre application web et identifier les différentes exigences de notre futur système. Nous avons préparé par la suite notre planning de travail en respectant les priorités de nos besoins suite a une discussion entre l’équipe du développement et le directeur du produit.

Malgré les contraintes de temps est les difficultés techniques que nous avons rencontrée qui se résument principalement dans la compréhension des nouveaux langages de développement et l’essais des nouvelles solutions techniques, nous avons réussi à réaliser la totalité de notre solution.

Le travail dans le cadre de ce projet fin d’étude, était d’une importance considérable dans la mesure où il nous a servi comme portail vers le monde Professional. Dans ce point de vue technique, l nous a permis de mettre en œuvre les acquis théoriques que nous avons appris tout au long de notre cursus universitaire et de les enrichir. Outre ce projet était aussi enrichissant pour les bonnes pratiques de la gestion de Project et codage.

Finalement notre travail ne s’arrête pas à ce niveau, en effet plusieurs fonctionnalités peuvent être ajoutées à notre application notamment l’application de plus de l’intelligence tel qu’un système de recommandation, ou l’implémentation d’une assistante virtuelle.

**Webographie**

[1] <https://www.leadingagile.com/2012/08/simple-cheat-sheet-to-sprint-planning-meeting/> [Accédé mai 2021]

[2] <https://www.visual-paradigm.com/scrum/what-is-sprint-review/> [*accédé mai 2021]*

[3] <https://www.mountaingoatsoftware.com/agile/scrum/meetings/sprint-retrospective#:~:text=The%20sprint%20retrospective%20is%20usually,which%20is%20usually%20quite%20sufficient> [*accédé mai 2021]*

[4 ] <https://www.visual-paradigm.com/scrum/daily-scrum-meeting-quick-guide/> [*accédé juin 2021]*

[5] [Product Owner (PO) / Product Manager - Qui est-ce ? (clementine.jobs)](https://www.clementine.jobs/fiches-metiers/metiers-marketing-publicite-digitale/product-owner/) [*accédé juin 2021]*

[ 6] [Certified ScrumMaster (CSM) Course | Scrum Alliance](https://www.scrumalliance.org/get-certified/scrum-master-track/certified-scrummaster) [*accédé june 2021]*

[7] [What is a Scrum Development Team?](https://www.scrum.org/resources/what-is-a-scrum-developer#:~:text=Learn%20About%20the%20Role%20of,with%20the%20domain%20of%20work.) [*accédé juin 2021]*

[8] [What is a Product Backlog? | Definition and Overview (productplan.com)](https://www.productplan.com/glossary/product-backlog/#:~:text=Definition%3A%20A%20product%20backlog%20lists,set%20forth%20in%20the%20roadmap.&text=Typical%20items%20in%20a%20product,bug%20fixes%2C%20and%20other%20tasks.) [*accédé juin 2021]*

[9] [Sprint Backlog and the Scrum Sprint (mountaingoatsoftware.com)](https://www.mountaingoatsoftware.com/agile/scrum/scrum-tools/sprint-backlog) [*accédé juin 2021]*

[10] [Diagramme de séquence — Wikipédia (wikipedia.org)](https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence) [*accédé juin 2021]*

[11] [Paiement en ligne - Définition du glossaire (ecommercemag.fr)](https://www.ecommercemag.fr/Definitions-Glossaire/Paiement-ligne-245342.htm) [*accédé juin 2021]*