



Faculté de Sciences et Techniques

M1 Informatique

Projet : Réalisation d'un site de pâtisserie

Document d'analyse des besoins

Réalisé par :

SANNY Abdul-Qadir

NIBAREKE Thérance

ZOUBIR Amina

Tajri Ikram

Garces Cala Maria Fernanda

MOMBOULI Trinité

Diallo Boubacar Biro

Encadré par: C. ROUDET et L. ABROUK

Année universitaire : 2022 – 2023

Table des matières

1	<i>Glossaire.....</i>	3
2	<i>Introduction.....</i>	4
3	<i>Modèle de cycle de vie</i>	4
3.1	Cycle de vie du projet	5
4	<i>Diagramme Use Case</i>	6
4.1	Définition.....	6
4.2	Composition du diagramme de cas	6
4.3	Spécification des acteurs du système	7
5	<i>Diagramme de cas d'utilisation</i>	8
6	<i>Documentation de cas d'utilisation</i>	8
6.1	Cas d'utilisation associe au visiteur	8
6.1.1	Visualiser produit	8
6.1.2	Rechercher produit.....	9
6.1.3	Gérer panier.....	9
6.2	Cas d'utilisation associe au client	10
6.2.1	Gérer commande client.....	10
6.2.2	Voir commande	10
6.2.3	Valider commande	10
6.2.4	Annuler commande	11
6.2.5	Effectuer paiement.....	11
6.3	Cas d'utilisation associe à l'administrateur	12
6.3.1	Gérer clients.....	12
6.3.2	Gérer commandes	13
6.3.3	Gérer produits	13
6.3.4	Ajouter produit	13
6.3.5	Modifier produit	14
6.3.6	Supprimer produit	14
6.4	Cas d'utilisation associe à l'administrateur et le client	15
6.4.1	Authentification.....	15

6.4.2	Se déconnecter	15
7	Références.....	16

1 Glossaire

- **Product Owner** : personne responsable du produit et il représente les utilisateurs finaux, ses principales fonctions sont : définir la liste des fonctionnalités du produit et les prioriser selon leur importance et leur valeur ajoutée pour l'entreprise, choisir la date de livraisons et les valider avec l'équipe de développement, clarifier les besoins à l'équipe de développement si nécessaire, etc.
- **Scrum master** : personne responsable d'assurer que Scrum est bien appliquée et respectée. Aussi, il doit encourager l'équipe à apprendre et à progresser pour qu'elle soit fonctionnelle, productive et créative durant le projet.
- **Équipe Scrum** : elle regroupe tous les rôles habituellement nécessaires à un projet, à savoir l'architecte, le concepteur, le développeur, le testeur, etc. Son rôle principal est : transformer les besoins exprimés dans le Sprint Backlog en fonctionnalités utilisables et livrer régulièrement une version fonctionnelle du produit.
- **Product Backlog** : est une liste de besoins. Ils sont créés par le Product Owner et le client mais seulement priorisés par le Product Owner.
- **Sprint Backlog** : est une liste de tâches que l'équipe de développement souhaite accomplir au cours d'un sprint. Il est extrait du Product Backlog.
- **Sprint Retrospective** : est un événement qui est fait en fin de chaque sprint. L'équipe analyse ce qui s'est passé et ce qu'elle aimerait améliorer pour l'avenir.

2 Introduction

Ce document vise à faire connaître le plan de gestion du projet pour le développement d'un produit logiciel. Par conséquent, le cycle de vie qui sera utilisé et l'analyse des besoins sont expliqués.

Le logiciel à développer propose d'accompagner une pâtisserie afin de faciliter les processus qu'elle réalise et de réduire le temps de les faire. En conséquence, une plateforme sera offerte où le propriétaire pourra gérer ses produits (les différents types de pâtisseries offertes) et le client pourra accéder rapidement au service pour commander et se faire livrer dans les plus brefs délais.

Pour ce projet, il est prévu de réaliser une implémentation basée sur ce qui précède, qui offrira des fonctionnalités orientées pour répondre aux besoins des clients. Nous utiliserons le model MVC qui consistera en l'utilisation des *Services Web* et la mise en œuvre de *Back-end* et *Front-end*. En particulier, les fonctionnalités suivantes seront implémentées (pour plus de détails voir la [Section 6. Documentation de cas d'utilisation](#)) :

- Accéder à la plateforme
- Gestion de produits créés
- Gestion de profils clients
- Voir et recherche de produits disponibles
- Faire une commande
- Gestion de commandes passées par les clients

3 Modèle de cycle de vie

Afin d'identifier, de spécifier et d'associer les phases du projet, il est nécessaire d'adopter un cadre qui est utilisé pour la planification, la gestion et le contrôle des processus associés au cycle de développement logiciel [1]. Ces cadres sont connus sous le nom de modèles de cycle de vie et doivent être sélectionnés et adaptés en fonction des caractéristiques du projet. Les éléments suivants ont été pris en compte :

- Les spécifications du projet seront constantes parce qu'il n'y aura pas de propositions de changement par le client tout au long de l'exécution du projet.
- La gestion forte de l'interface graphique pour l'interaction avec l'utilisateur est obligatoire. Donc, il nécessite un modelé itératif et incrémental puisque cette partie du système est difficile à spécifier [2].

Par rapport aux méthodologies traditionnelles, le modèle en spirale de Boehm a été pris en compte parce qu'il s'agit d'un modèle itératif qui cherche à résoudre les problèmes présentés par le modèle en cascade [3]. Ce modèle a quatre phases : planification, évaluation, analyse des risques et développement, au cours desquelles des itérations sont effectuées avec le processus de gestion des risques comme axe principal afin de minimiser les problèmes associés aux prototypes générés à chaque itération [3]. Ces avantages pourraient atténuer les risques associés au développement de notre application. Cependant, la réussite de ce modèle est centrée sur le niveau d'expérience de l'équipe, qui doit être élevé, car si une mauvaise gestion des risques était effectuée, le projet en serait affecté. Par conséquent, ce modèle ne peut pas être considéré pour le projet.

D'un autre côté, pour le développement du projet, la **méthode agile Scrum**, est à l'opposé parce qu'elle permet un mécanisme systématisé de réduction des risques. De la même manière, le modèle peut être bénéficié en termes de flexibilité car la seule constante est le changement, qui est géré pendant le développement des itérations. Enfin, l'équipe Scrum est auto organisée et se concentre

sur la motivation parce que c'est possible de voir les résultats rapidement. Ces caractéristiques permettent un travail de qualité, qui viens de l'engagement de l'équipe pendant le développement du projet [4]. De même, Scrum, est une méthode agile qui aide à la planification itérative et incrémental du produit. Elle a petites itérations connues comme *sprints* qui résultent dans la livraison des fonctionnalités. Cette méthodologie définis des notions comme : *Product Owner*, *Scrum master*, *Equipe Scrum*, *Product Backlog*, *Sprint Backlog* et *Sprint Retrospective* (Voir [Section 1. Glossaire](#)). D'autre part, Scrum, est une méthode agile qui aide à la planification itérative et incrémental du produit. Elle a petites itérations connues comme *sprints* qui résultent dans la livraison des fonctionnalités. Aussi, en cette méthodologie, ils sont définis des notions comme : *Product Owner*, *Scrum master*, *Equipe Scrum*, *Product Backlog*, *Sprint Backlog* et *Sprint Retrospective* (Voir [Section 1. Glossaire](#)).

Compte tenu de ce qui précède, il a été décidé d'utiliser le **modèle de cycle de vie agile basé sur la méthodologie Scrum**.

3.1 La méthode agile

Afin de concevoir et développer notre application, nous avons opté pour le modèle de cycle de vie Agile. Ce choix revient au fait que la méthode agile plus de souplesse dans le cycle de vie du projet, et ayant une approche itérative et incrémentale qui est menée dans un esprit collaboratif.

La Figure 1. Cycle de vie agile affiche le processus général du cycle de vie agile. Il a 6 phases, cependant, en raison du périmètre du projet, ils sont considérés les étapes : **Idéation, Construction et Déploiement**.

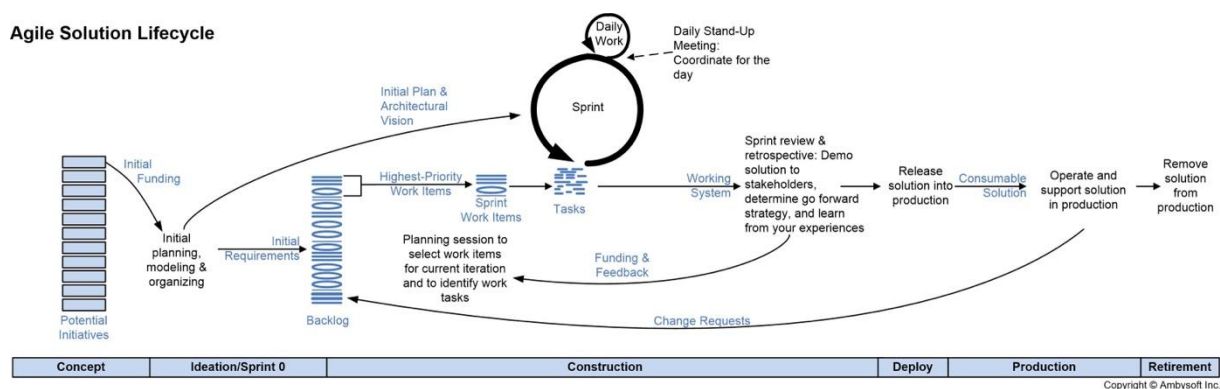


Figure 1. Cycle de vie agile

Pendant l'étape **d'Idéation**, les objectifs que l'équipe doit atteindre sont identifiés. Cela nécessite un travail acharné avec les *stakeholders* pour exposer initialement les besoins, l'architecture, la planification des versions et d'autres activités.

D'après cela, dans le projet, les cas d'utilisation abstraits des besoins, seront représentés dans le *Product Backlog*, qui sera affiné dans les étapes ultérieures. Aussi, dans cette phase, les besoins déjà définies sont également utilisées pour planifier une architecture et conception initiale du produit. Il est généralement organisée une réunion pour approuver et affiner les propositions résultantes.

D'autre part, dans la phase de **Construction**, se produit une série d'itérations impliquant le développement de l'application. Chaque *Sprint* comprendra les phases de développement logiciel : analyse, conception, développement, tests et livraison et visera également à produire un livrable.

Pour ce projet, chaque itération aura une durée d'une semaine et au moins une réunion SCRUM aura lieu tout au long de l'itération pour la solution des problèmes ou l'atténuation des risques. De la même manière, dans l'activité de planification du Sprint, le *Sprint Backlog* sera réalisé, qui indiquera les tâches à effectuer pour chaque membre de l'équipe lors de l'itération et aussi, les tâches seront estimées avec l'outil *Poker Planning*. À la fin de chaque itération, une révision de Sprint sera effectuée où ce qui a été appris sera évalué, les risques du projet seront réévalués et le *Product Backlog* et le *Release Backlog* seront mis à jour.

Finalement, dans la phase de **Déploiement**, le produit sera livré avec les tâches d'intégration et les tests système. Une révision de la documentation du système sera également incluse.

4 Diagramme de cas d'utilisation (Use Case)

4.1 Définition

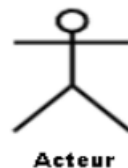
Les rôles des diagrammes de cas d'utilisation sont de recueillir, d'analyser et d'organiser les besoins ainsi que de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML pour la conception d'un système.

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Ainsi ces cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

Il ne faut jamais négliger cette première étape pour produire un site web conforme aux attentes des utilisateurs ciblés. Pour élaborer les cas d'utilisation, il faut se fonder sur des entretiens avec les utilisateurs.

4.2 Composition du diagramme de cas

Le diagramme de cas se compose de trois éléments principaux :



Un Acteur : c'est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.

Un cas d'utilisation : c'est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service. Il représente par une ellipse.



Contenant le nom du cas (un verbe à l’infinitif), et optionnellement, au-dessus du nom, un stéréotype.

Les relations : Trois types de relations sont pris en charge par la norme UML et sont graphiquement représentées par des types particuliers de ces relations. Les relations indiquent que le cas d'utilisation source présente les mêmes conditions d'exécution que le cas issu. Une relation simple entre un acteur et une utilisation est un trait simple.

4.3 Spécification des acteurs de notre système

Connaître les acteurs qui interagissent avec le système est essentiel pour comprendre son fonctionnement et les cas d’utilisation établis. Sur le Tableau 1. Spécification des acteurs, figure une description de chaque acteur avec les actions qu’il peut exécuter.

Tableau 1. Spécification des acteurs

Acteur	Description
Visiteur	Utilisateur inconnu du site qui a pour rôle de visiter le site en cherchant un produit. Il peut voir tous les produits et les ajouter au panier.
Client	Utilisateur ayant déjà créer un compte sur le site. En plus d’avoir les capacités du visiteur, il peut suivre le processus d’achat des produits. Cela irait de la gestion des commandes à son paiement.
Administrateur	Personne qui travaille dans la pâtisserie, on l’appelle généralement « le webmaster ». Il connaît le fonctionnement du site web et de l’entreprise. Il assure le dynamisme du site et veille sur les mises à jour des produits, de leurs prix, de leurs disponibilités, de la gestion des paiements et la gestion des livraisons. Aussi, faire autres actions de gestion telles que la modification de clients.

5 Diagramme de cas d'utilisation

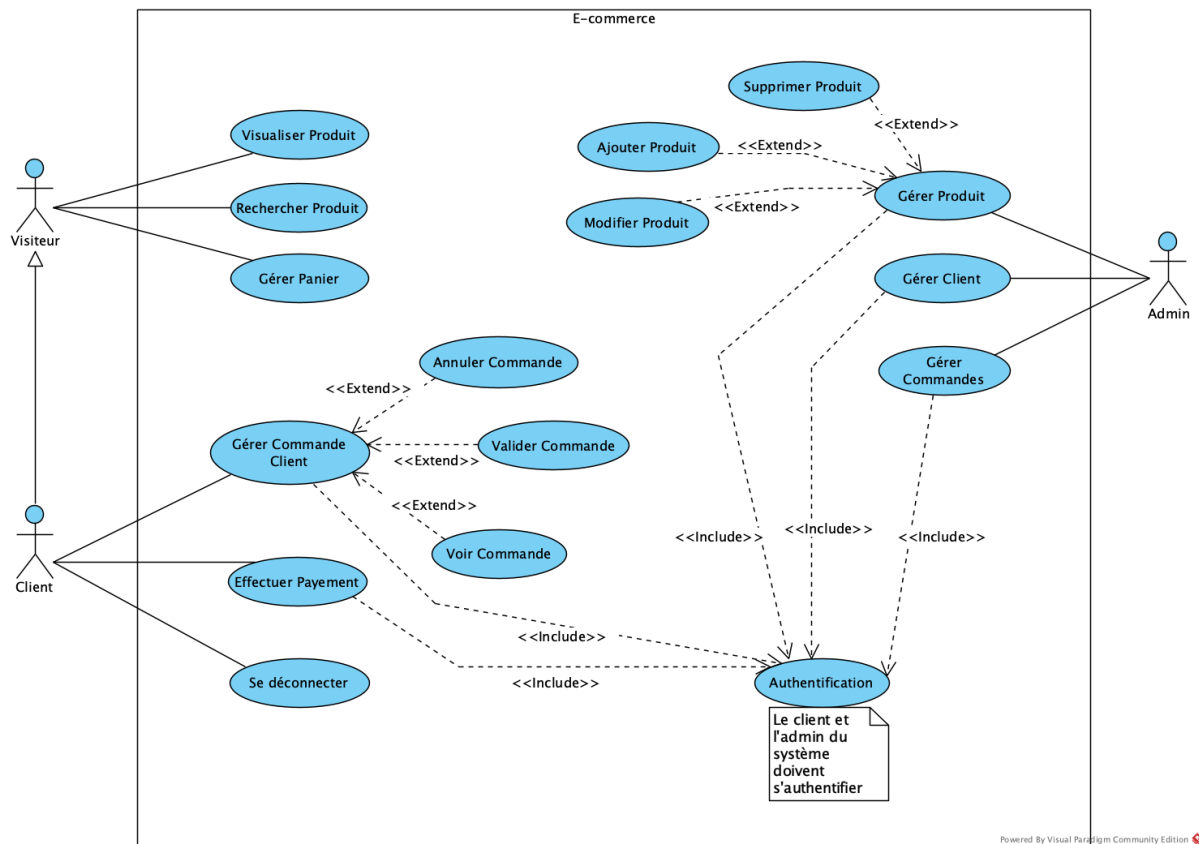


Figure 2. Diagramme de cas d'utilisation

6 Documentation des cas d'utilisation

6.1 Cas d'utilisation associé au visiteur

6.1.1 Visualiser produit

Nom	Visualiser produit	Version	1.0
Responsable	Équipe conception		
Résumé	Le visiteur regarde toutes les informations sur un produit sélectionné telles que la description, l'image, le prix		
Acteurs principaux	Visiteur (En raison de l'héritage, le client interagit également avec ce cas d'utilisation)		
Date de création	18/10/2022	Date de mise à jour	
Préconditions			
Scénario nominal			
<ol style="list-style-type: none"> 1. Le visiteur clique sur un produit 2. Le système affiche l'information du produit 3. Le client regarde l'information 			
Scénarii alternatifs			

Au point 2. Le système n'affiche pas toutes les informations

Postconditions

6.1.2 Rechercher produit

Nom	Rechercher produit	Version	1.0
Responsable	Équipe conception		
Résumé	Le visiteur veut rechercher un produit sur le site		
Acteurs principaux	Visiteur (En raison de l'héritage, le client interagit également avec ce cas d'utilisation)		
Date de création	18/10/2022	Date de mise à jour	
Préconditions			
Scénario nominal			
1. Le client saisit un mot clé sur le champ de recherche 2. Le système recherche le produit 3. Le système affiche les produits trouvés			
Scénarii alternatifs			
Au point 3. Le système ne trouve pas des produits avec le mot clé. Le système affiche un message qu'il n'y a pas aucun résultat.			
Postconditions			

6.1.3 Gérer panier

Nom	Gérer panier	Version	1.0
Responsable	Équipe conception		
Résumé	Le client souhaite ajouter ou supprimer un produit au panier		
Acteurs principaux	Visiteur (En raison de l'héritage, le client interagit également avec ce cas d'utilisation)		
Date de création	18/10/2022	Date de mise à jour	
Préconditions			
Scénario nominal			
1. Le client clique sur « ajouter au panier » 2. Le système met à jour le panier 3. Le client clique sur le panier 4. Le système affiche le panier 5. Le client souhaite supprimer le produit. Il clique sur « supprimer le produit ». 6. Le système met à jour le panier			
Scénarii alternatifs			
Au point 5. Le client souhaite changer la quantité du produit			
Postconditions			
• Le système enregistre toutes les mises à jour			

6.2 Cas d'utilisation associe au client

6.2.1 Gérer commande client

Nom	Gérer commande client	Version	1.0
Responsable	Équipe conception		
Résumé	Après le client ai fait une commande, il souhaite voir toutes les commandes passées		
Acteurs principaux	Client		
Date de création	18/10/2022	Date de mise à jour	
Préconditions <ul style="list-style-type: none">Le client doit être connecté			
Scénario nominal			
<ol style="list-style-type: none">Le client veut voir ses commandes, donc il clique sur l'option dans le menuLe système affiche la liste des commandesLe client interagit avec l'information. S'il veut voir une commande <<extends>> Voir commande. S'il veut annuler une commande <<extends>> Annuler commande.			
Scénarii alternatifs			
Au point 2. Le système n'affiche les informations souhaitées			
Postconditions			

6.2.2 Voir commande

Nom	Voir commande	Version	1.0
Responsable	Équipe conception		
Résumé	Le client souhaite voir une commande et le système affiche l'information correspondante		
Acteurs principaux	Client		
Date de création	18/10/2022	Date de mise à jour	
Préconditions <ul style="list-style-type: none">Le client a déjà fait au moins une commandeLe client doit être connecté			
Scénario nominal			
<ol style="list-style-type: none">Le client clique sur la commande dont il souhaite voir l'informationLe système affiche l'information de la commandeLe client voit l'information de la commande			
Scénarii alternatifs			
Au point 2. Le système n'affiche pas l'information			
Postconditions			

6.2.3 Valider commande

Nom	Valider commande	Version	1.0
Responsable	Équipe conception		
Résumé	Le client valide sa commande		
Acteurs principaux	Client		
Date de création	18/10/2022	Date de mise à jour	

Préconditions	
<ul style="list-style-type: none"> Le client a déjà ajouté au moins un produit au panier Le client doit être connecté 	
Scénario nominal	
<ol style="list-style-type: none"> Le client clique sur le panier Le système affiche le panier Le client vérifie sa commande Le client clique sur « valider la commande » Le client choisi son mode de paiement 	
Scénarii alternatifs	
Au point 4. Le client ne valide pas sa commande et continue d'utiliser l'application	
Postconditions	
<ul style="list-style-type: none"> Le système affiche la page de récapitulatif de la commande 	

6.2.4 Annuler commande

Nom	Annuler commande	Version	1.0
Responsable	Équipe conception		
Résumé	Le client souhaite annuler une commande et le système met à jour la liste des commandes		
Acteurs principaux	Client		
Date de création	18/10/2022	Date de mise à jour	
Préconditions			
<ul style="list-style-type: none"> Le client a déjà fait au moins une commande Le client doit être connecté Le délai règlementaire pour annuler une commande n'est pas encore dépassé 			
Scénario nominal			
<ol style="list-style-type: none"> Le client clique sur la commande dont il souhaite voir l'information Le système affiche l'information de la commande Le client clique sur le bouton d'annuler la commande Le système affiche un modal de confirmation qui demande au client s'il est sûr d'annuler la commande Le client confirme l'annulation de la commande Le système change l'état de la commande vers annulé Le système met à jour la liste 			
Scénarii alternatifs			
Au point 5 : <ol style="list-style-type: none"> Le client ne confirme pas l'annulation Le système ferme le modal Le système n'effectue aucune opération sur la commande 			
Postconditions			
<ul style="list-style-type: none"> Le système enregistre l'annulation de la commande 			

6.2.5 Effectuer paiement

Nom	Effectuer payement	Version	1.0
Responsable	Équipe conception		
Résumé	Après avoir valider la commande, le client dois procéder au paiement.		

Acteurs principaux	Client		
Date de création	18/10/2022	Date de mise à jour	
Préconditions <ul style="list-style-type: none"> Le client a déjà validé la commande 			
Scénario nominal			
1. Le système affiche les moyens de paiement : A la livraison et carte bancaire 2. Le client choisit l'option : A la livraison 3. Le système valide la commande			
Scénarii alternatifs			
Au point 3. <ol style="list-style-type: none"> Le client choisit l'option : Carte Le système affiche une nouvelle page pour mettre l'information de la carte et procéder au paiement Le système valide l'information Le système valide la commande 			
Postconditions <ul style="list-style-type: none"> Le système enregistre une commande 			

6.3 Cas d'utilisation associé à l'administrateur

6.3.1 Gérer clients

Nom	Gérer client	Version	1.0
Responsable	Équipe conception		
Résumé	L'administrateur souhaite modifier ou voir la liste des clients.		
Acteurs principaux	Administrateur		
Date de création	18/10/2022	Date de mise à jour	
Préconditions <ul style="list-style-type: none"> L'administrateur doit être connecté 			
Scénario nominal			
1. L'administrateur clique sur l'onglet « clients » 2. Le système affiche la liste des clients 3. L'administrateur interagit avec l'information dans la vue 4. L'administrateur clique sur le bouton « Modifier » dans un client 5. Le système affiche l'information du client 6. L'administrateur modifie les champs qu'il veut (uniquement les champs modifiables) 7. L'administrateur confirme les modifications 8. Le système valide les modifications 9. Le système enregistre les modifications			
Scénarii alternatifs			
Au point 8. Modifications invalides. Le système affiche un message d'erreur Au point 9. Le système n'enregistre pas les modifications. Il affiche un message d'erreur, récupérer les informations précédentes et ferme le processus			
Postconditions <ul style="list-style-type: none"> Les modifications du client sont enregistrées 			

6.3.2 Gérer commandes

Nom	Gérer commandes	Version	1.0
Responsable	Équipe conception		
Résumé	L'administrateur souhaite voir la liste de commandes et modifier une commande		
Acteurs principaux	Administrateur		
Date de création	18/10/2022	Date de mise à jour	
Préconditions <ul style="list-style-type: none"> L'administrateur doit être connecté 			
Scénario nominal			
<ol style="list-style-type: none"> L'administrateur clique sur l'onglet des commandes Le système affiche la liste des commandes Le système affiche l'information de la commande L'administrateur modifie les champs qu'il veut. Par exemple : l'état de la commande. L'administrateur confirme les modifications Le système valide les modifications Le système enregistre les modifications 			
Scénarii alternatifs			
Au point 8. Modifications invalides. Le système affiche un message d'erreur Au point 9. Le système n'enregistre pas les modifications. Il affiche un message d'erreur, récupérer les informations précédentes et ferme le processus			
Postconditions <ul style="list-style-type: none"> Les modifications de la commande sont enregistrées 			

6.3.3 Gérer produits

Nom	Gérer produits	Version	1.0
Responsable	Équipe conception		
Résumé	L'administrateur souhaite ajouter, modifier, supprimer un produit ou voir la liste des produits		
Acteurs principaux	Administrateur		
Date de création	18/10/2022	Date de mise à jour	
Préconditions <ul style="list-style-type: none"> L'administrateur doit être connecté 			
Scénario nominal			
<ol style="list-style-type: none"> L'administrateur clique sur l'onglet « produits » Le système affiche la liste des produits ajoutés Le client interagit avec l'information dans la vue. S'il veut ajouter un produit. S'il veut modifier un produit. S'il veut supprimer un produit. 			
Scénarii alternatifs			
Au point 2. Le système affiche l'information incomplète			
Postconditions			

6.3.4 Ajouter produit

Nom	Ajouter produit	Version	1.0
Responsable	Équipe conception		
Résumé	L'administrateur souhaite ajouter un produit		
Acteurs principaux	Administrateur		

Date de création	18/10/2022	Date de mise à jour	
Préconditions			
<ul style="list-style-type: none"> L'administrateur doit être connecté 			
Scénario nominal			
<ol style="list-style-type: none"> L'administrateur clique sur « Ajouter un produit » Le système affiche un formulaire de création L'administrateur saisit l'information du produit L'administrateur confirme Le système enregistre le produit Le système affiche un message de création réussite 			
Scénarii alternatifs			
Au point 5. Le système ne peut pas enregistrer le produit. Il affiche un message d'erreur.			
Postconditions			
<ul style="list-style-type: none"> Le produit créé est enregistré 			

6.3.5 Modifier produit

Nom	Modifier produit	Version	1.0
Responsable	Équipe conception		
Résumé	L'administrateur souhaite modifier un produit		
Acteurs principaux	Administrateur		
Date de création	18/10/2022	Date de mise à jour	
Préconditions			
<ul style="list-style-type: none"> L'administrateur doit être connecté Il y a au moins un produit créé 			
Scénario nominal			
<ol style="list-style-type: none"> L'administrateur clique sur le bouton « Modifier » dans un produit Le système affiche l'information du produit L'administrateur modifie les champs qu'il veut L'administrateur confirme les modifications Le système valide les modifications Le système enregistre les modifications 			
Scénarii alternatifs			
Au point 5. Modifications invalides. Le système affiche un message d'erreur Au point 6. Le système n'enregistre pas les modifications. Il affiche un message d'erreur, récupérer les informations précédentes et ferme le processus			
Postconditions			
<ul style="list-style-type: none"> Les modifications du produit sont enregistrées 			

6.3.6 Supprimer produit

Nom	Supprimer produit	Version	1.0
Responsable	Équipe conception		
Résumé	L'administrateur souhaite supprimer un produit		
Acteurs principaux	Administrateur		
Date de création	18/10/2022	Date de mise à jour	

Préconditions
<ul style="list-style-type: none"> • L'administrateur doit être connecté • Il y a au moins un produit créé
Scénario nominal
<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton « Supprimer » dans un produit 2. Le système affiche un modal de confirmation 3. L'administrateur confirme l'action 4. Le système supprime le produit 5. Le système met à jour la liste de produits
Scénarii alternatifs
Au point 4. Le système ne peut pas supprimer le produit. Il affiche un message d'erreur, récupérer les informations précédentes et ferme le processus
Postconditions
<ul style="list-style-type: none"> • La liste des produits disponibles est mise à jour

6.4 Cas d'utilisation associé à l'administrateur et le client

6.4.1 Authentification

Nom	Authentification	Version	1.0
Responsable	Équipe conception		
Résumé	Un client s'authentifie pour utiliser les fonctionnalités du système.		
Acteurs principaux	Client, administrateur (pour des raisons pratiques, le terme utilisateur sera utilisé pour désigner les deux acteurs)		
Date de création	18/10/2022	Date de mise à jour	
Préconditions	<ul style="list-style-type: none"> • L'utilisateur doit posséder un compte. 		
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur se connecter. 2. Le système affiche le formulaire d'authentification. 3. L'utilisateur saisit le <i>username</i> et le mot de passe. 4. Le système valide l'information. 5. Le système se connecte et affiche la page d'accueil (en fonction du type de l'utilisateur). 		
Scénarii alternatifs	<p>Au point 3. L'utilisateur saisit un mauvais <i>username</i> ou mot de passe. Le système affiche un message d'erreur.</p> <p>Au point 3. L'utilisateur ne saisit pas un des champs de formulaire. Le système affiche un message d'erreur.</p> <p>Au point 4 :</p> <ol style="list-style-type: none"> a. Le système ne trouve pas le client. Le système affiche un message d'erreur. b. Le système revient au point 2 en affichant le formulaire. c. L'utilisateur a l'option de créer un compte ou de ressayer. <p>Au point 5. Le system ne se connecte pas. Le système affiche un message d'erreur.</p>		
Postconditions	<ul style="list-style-type: none"> • Le système enregistre la date et l'heure d'authentification d'utilisateur. 		

6.4.2 Se déconnecter

Nom	Se déconnecter	Version	1.0
Responsable	Équipe conception		

Résumé	Le client ferme sa session dans le système		
Acteurs principaux	Client, administrateur (pour des raisons pratiques, le terme utilisateur sera utilisé pour désigner les deux acteurs)		
Date de création	18/10/2022	Date de mise à jour	
Préconditions			
<ul style="list-style-type: none"> L'utilisateur doit être connecté. 			
Scénario nominal			
<ol style="list-style-type: none"> L'utilisateur clique sur se déconnecter Le système valide l'action Le système déconnecte le client 			
Scénarii alternatifs			
Au point 2. Le système ne valide pas l'action. Le système affiche un message d'erreur.			
Postconditions			
<ul style="list-style-type: none"> Le système enregistre la date et l'heure auxquelles l'utilisateur s'est déconnecté. 			

7 Conclusion

Le projet que nous serons amenés à développer vise à permettre à des internautes(clients) de commander des pâtisseries sur notre site internet. Le gérant de pâtisserie (admin ou webmaster) pourra, grâce à un back office gérer la liste des clients, des commandes et des produits à afficher. Après analyse de différentes possibilités, du contexte de notre projet et des ressources à notre disposition, nous avons opté pour un cycle de vie de développement logiciel basé sur la méthode agile scrum. Les cas d'utilisation principaux ont été analysés nous permettant de planter le décor pour l'analyse de spécification fonctionnelle.

8 Références

- [1] V. Rastogi, «Software Development Life Cycle Models Comparison, Consequences,» *International Journal of Computer Science and Information Technologies*, Vol. 6, pp. 168-172, 2015.
- [2] I. Sommerville, *Software Engineering* 9th Edition, Pearson, 2011.
- [3] N. Ruparelia, «Software Development Lifecycle Models,» *ACM SIGSOFT Software Engineering Notes*, vol. 35, n° 13, pp. 8-13, 2010.
- [4] N. Bhatta et M. Thite, «Agile approach to e-HRM project management,» chez *e-HRM: Digital Approaches, Directions & Applications*, Taylor & Francis, 2019.
- [5] «The Agile System Development Lifecycle (SDLC),» [En ligne]. Available: <http://www.ambysoft.com/essays/agileLifecycle.html>. [Accès le 15 Novembre 2022].

