

Université Vincennes–Saint-Denis

PARIS 8

Département UFR MITSIC

Option technologie de l'Hypermédia



Cahier des charges **-ORIENTAL-**



REALISER PAR

➤ BOUGOFFA Mohamed Amine

Sommaire

1.périmètre de projet	3
2.Contexte et objectif du projet.....	3
3.découpage du projet	3
4.Analyse des besoins	4
a. Processus unifié (UP)	
b. Unified Modeling Language(UML)	
5.Utilisation et application	4
6.Identification des acteurs	4
7.Identification des cas d'utilisation	5
8.Diagramme de contexte	6
9.Identification des messages échangés	7
10. Diagramme des cas d'utilisation	7
11. Diagramme des cas d'utilisation global.....	7
12. Présentation diagramme de classe	8
13. Diagramme de classe	8
14. Modèle relationnel	8
15. Conclusion	9
16. Environnement développement	9
17. Diagramme de déploiement.....	9

1- Périmètre du projet

Le projet www.oriental.com est une version hypertextuelle qui consiste à produire une version numérique pour une pub d'une pâtisserie et ateliers de gâteaux orientaux. Elle offre aux clients l'opportunité de découvrir l'originalité et la diversité que propose ce site.

En effet, notre site offre un accès immédiat grâce à une vitrine en ligne, dans lequel on trouve une variété de gâteaux orientaux.

Pour avoir accès au site, le client devra s'identifier par le biais d'une interface web qui lui permettra de voir les différentes variétés de gâteaux et d'en commander ou même d'apprendre via des tuto proposer.

2- Contexte et objectif du projet

Les pâtisseries orientales sont originaire des pays du Moyen-Orient ou du Maghreb. Les plus connues sont les suivantes : makrout, baklava, zlabiya, cornes de gazelle, chebakia...

Les pâtisseries orientales sont diverses et variées, mais on retrouve généralement les mêmes ingrédients : - miel - amandes - noix - dattes - graines de sésame.

Notre travail, donne l'opportunité à différentes populations de découvrir ces différentes variétés et saveurs de façon à soit se sera à l'intéressé (client) de passer directement commande en magasin sinon via le service de livraison.

à la différence des autres sites existant, notre réalisation du site permettra aux utilisateurs d'apprendre à réaliser par eux-mêmes les recettes à travers différents tuto mis à leurs dispositions.

3- découpage du projet :

A fin de montrer les différentes étapes suivies et traversées lors du développement de notre application web, nous allons illustrer ceci par un diagramme de Gantt comme suit :

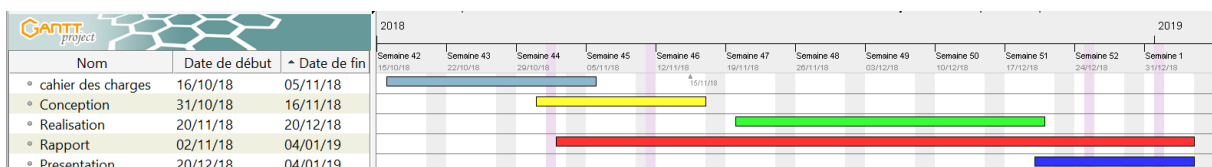


Figure 1: Diagramme de Gantt.

4-Analyse des besoins :

L'étape de l'analyse des besoins est la deuxième phase de cycle de vie du Processus Unifié et l'une des étapes les plus importantes à considérer ; en effet si les besoins sont mal spécifiés et exprimés, ou mal analysés, toute la suite devra être refaite, d'où l'importance accordée à cette activité.

Notre objectif dans cette étape est donc d'exprimer les besoins attendus du futur système à développer.

Dans ce qui suit nous allons définir le processus de développement suivi durant la conception de notre application et le langage avec lequel nous l'avons modélisé :

1. Processus unifié (UP): est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation. C'est un patron de processus pouvant être adaptée à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

2. Unified Modeling Language(UML): que l'on peut traduire par «*langage de modélisation unifié* » est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en terme de modélisation objet, à un tel point que sa connaissance est souvent nécessaire pour obtenir un poste de développeur objet.

Le langage UML fournit 13 diagrammes, qui sont classés selon deux types de vue :

- Diagrammes structurels ou diagrammes statiques (UML structure) dans lesquels nous avons choisis d'utiliser le diagramme de classe (class diagram).
- Diagrammes comportementaux ou diagrammes dynamiques (UML behavior) dans lesquels nous avons choisis d'utiliser le diagramme de cas d'utilisation (Use case diagram), diagramme de séquence (séquence diagram), diagramme de contexte et le diagramme de séquence détaillé.

5-Utilisations et application :

La phase d'analyse a pour objectif de décrire les interactions entre les acteurs et le système, au moyen de diagrammes qui facilitent la compréhension des scénarios. Afin de modéliser notre application web nous allons utiliser certains diagrammes pour la spécification des besoins.

6- Identification des acteurs

Le système à concevoir est régi par un site web qui sera accessible via le réseau intranet permettant ainsi un accès aux différents services souhaitant solliciter l'administration. Un acteur est une entité externe qui interagit avec le système (opérateur, centre distant, autre système...). En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin.

De ce fait, les acteurs interagissant avec l'application sont :

- Administrateur : l'administrateur est le superviseur qui gère toutes les commandes et les listes des annonces (gâteaux).

- Client : le client qui gère ces commandes (réserver ou annuler des commandes) ainsi visualiser la liste des annonces .

7-Identification des cas d'utilisation

Dans ce qui suit, nous allons identifier les différents cas d'utilisation significatifs de notre système. Les cas d'utilisation décrivent les tâches fonctionnelles effectuées par les utilisateurs de l'application.

Acteurs	Cas d'utilisation		Message
Administrateur	S'authentifier	Connexion	Emis : Demande d'accès au système
		Déconnexion	Reçu : Affichage de l'espace d'accueil.
	Gérer les clients	Supprimer	Emis : Demande de mises à jour des clients.
		Ajouter	Reçu : Affichage de l'espace d'ajout, de suppression des clients.
	Gérer les commandes	Ajouter	Emis : Demande de mises à jour des commandes.
		Modifier	Reçu : Affichage de l'espace d'ajout, de modification ou de suppression des commandes.
		Supprimer	

Client	Gérer les commandes	Ajouter	Emis : Demande de mises à jour des commandes.
		Modifier	Reçu : Affichage de l'espace d'ajout, de modification ou de suppression des commandes.
		Supprimer	
	Visualiser	Emis : Demande de visualiser la liste des gâteaux. Reçu : Affichage de l'espace liste des gâteaux .	

Tableau 1: Identification des cas d'utilisations

8- Diagramme de contexte

La description des différents acteurs permet de décrire le système à développer par un premier diagramme appelé « diagramme de contexte ». Ce dernier permet de présenter l'interaction des différents acteurs avec le système.

Le diagramme de contexte associé à notre système est représenté ci-dessous

9-Identification des messages échangés

Afin de mieux comprendre l'interaction de chaque acteur(administrateur) avec le système, nous énumérons ci-dessous les différentes communications :

Messages système	Messages acteur
M1 : Interface d'authentification M3 : Interface de gestion M5 : Interface de mise à jour adéquate M11:Afficher la liste des annonces	M2 : Authentification M4 : Gérer les commandes M6 : Gérer les clients M12 : Visualiser les annonces

Tableau 2 L'Identification des messages échangés

10- Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent ce qu'un système fait, du point de vue d'un observateur externe. L'accent est mis sur ce qu'un système fait, plutôt que sur la façon dont il le fait.

11-Diagramme de cas d'utilisation global

Pour mieux illustrer les fonctionnalités de l'ensemble des acteurs, nous avons établi un diagramme de récapitulation :

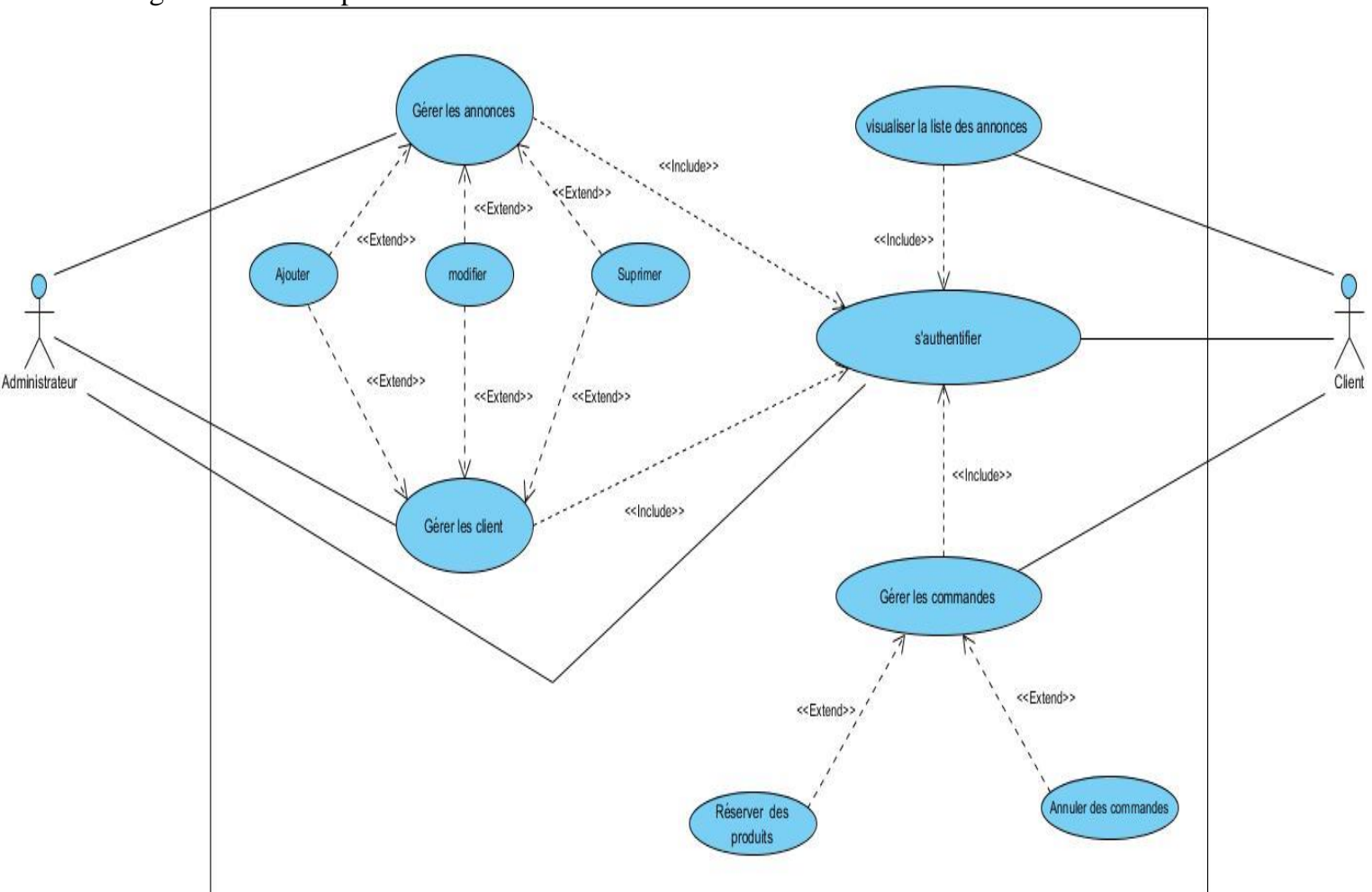


Figure 2: Diagramme de cas d'utilisation global

12-Présentation de diagramme de classe :

Le diagramme de classes est généralement considéré comme le plus important dans la conception d'une application, il constitue l'un des pivots essentiels de la modélisation. Ce diagramme appartient au modèle statique et permet de définir la structure interne de notre système, car il fait abstraction des aspects temporels et dynamiques. Cette représentation est centrée sur les concepts de classe et association.

13-Diagramme de classe

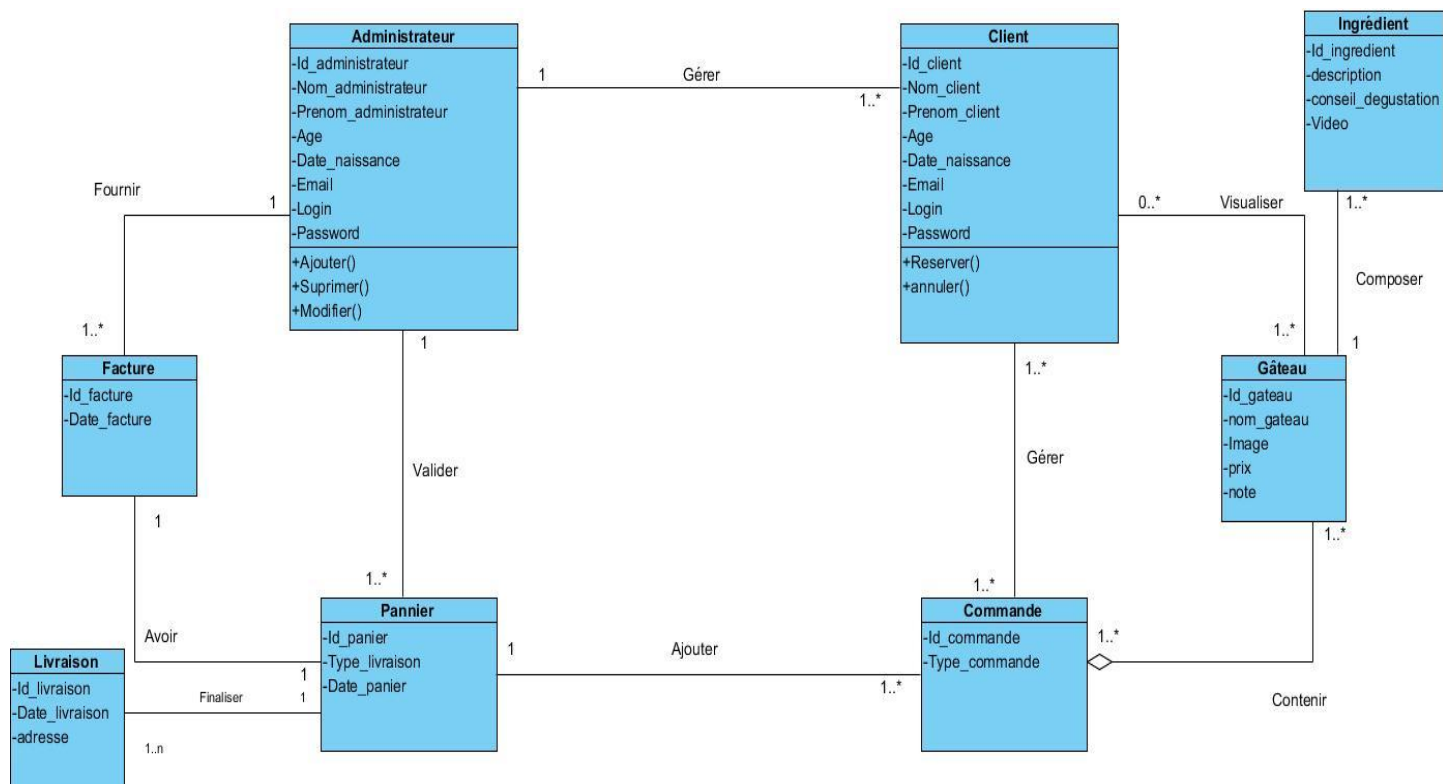


Figure 3 -Diagramme de classe

Présentation des classes et de leurs attributs

Classe	Attributs	Définition de l'attribut	type	Taille
Administrateur	Id_administrateur	L'identifiant de l'admin	Numérique	30
	Nom_administrateur	Le nom de l'administrateur	Alphanumérique	30
	Prenom_administrat	Le prenom de l'administrateur	Alphanumérique	30
	Age	L'age de l'administrateur	Numérique	3

	Date_naissance	La date de naissance de l'admin	DATE	15
	Login	L'email de l'admin	Alphanumérique	15
	password	Le mot de passe de l'admin	Alphanumérique	15
Client	Id_Client	L'identifiant de le Client	Numérique	30
	Nom_Client	Le nom de le Client	Alphanumérique	30
	Prenom_Client	Le prenom de le Client	Alphanumérique	30
	Age	L'age de le Client	Numérique	3
	Date_naissance	La date de naissance de le Client	DATE	
	Login	L'email de le Client	Alphanumérique	15
	password	Le mot de passe de le Client	Alphanumérique	15
Gâteau	Id_gateau	L'identifiant de le gâteau	Numérique	30
	Nom_gateau	Le nom de le gâteau	Alphanumérique	30
	Prix	Le prix de le gâteau	Numérique	50
Commande	Id_commande	L'identifiant de le gâteau	Numérique	30
	Type_commande	Le nom de le gâteau	Alphanumérique	30
Facture	Id_facture	L'identifiant de le facture	Numérique	30
	Date_facture	La date de la facture	DATE	10
Pannier	Id_pannier	L'identifiant de le panier	Numérique	30
	Type_pannier	Le type de panier	Alphanumérique	30
	Date_pannier	La date de panier	DATE	10

Tableau 3Tableau présentatif des classes et leurs attributs

14-Modèle relationnel

A partir du modèle de classes, et pour obtenir le modèle relationnel, nous avons appliqué les règles de passage suivantes :

* règle 1 : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.

* règle 2 : les associations (0/1...*) (0/1...1) : il faut ajouter un attribut de type clé étrangère dans la relation de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

* règle 3 : les associations (0/1...*) (0/1...*), n-aire et classe d'association : la classe association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.

* règle 4 : Transformation de l'héritage (Décomposition ascendante) : Il faut supprimer la (les) relation(s) issue(s) de la (des) sous-classe(s) et faire migrer les attributs dans la relation issue de la surclasse

Administrateur(id_administrateur, prenom, nom, age, email, login, password , #Id_client, #Id_pannier, #Id_facture)

Client(Id_client, prenom, nom, age, mail, login, mot_passe, #Id_commande, #id_gateau)

Commande (Id_commande, type_commande)

Gâteau (Id_gateau, nom_gateau, prix, image, note , #Id_ingredient)

Pannier (Id_pannier, type_livraison, date_pannier, #Id_commande, #Id_livraison)

Facture (Id_facture, date_facture)

Ingrédient (Id_ingredient, description, video)

livraison (Id_livraison, date_livraison, adresse)

15-Conclusion

A l'issus de cette étape, nous avons conçu et documenté le code que nous devons produire, l'activité de la conception a facilité la compréhension de notre système, qui ébauche vers l'activité d'implémentation

16-Environnement de développement

Pour la réalisation de ce travail, nous avons eu recours aux environnement et outils de développement suivants :

✓ **Visual Studio Code** est un environnement de développement libre permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation (HTML5, CSS3 , JavaScript , PHP, Bootstrap).

✓ **Wamp server** est une plateforme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.

✓ **Mysql Server** est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

✓ **GanttProject** est un logiciel libre, permettant de modéliser sous forme de **Gantt** la planification des différentes tâches qui sont nécessaires à un projet. Chaque tâche pourra être définie par sa durée et par les ressources qui pourront lui être affectées.

✓ **Visual Paradigm For UML** : Permet la création des diagrammes UML et des modèles qui en sont à l'origine. Ceux-ci peuvent alors générer du code dans un langage de programmation déterminé. Il propose également la création d'autres types de diagrammes, comme celui-ci qui permet la modélisation des bases de données

17-Diagramme de déploiement

Le diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

