
	<p>République Tunisienne</p> <p>Ministère de l'enseignement supérieure De la recherche scientifique</p> <p>Université de Jendouba</p> <p>Institut supérieur des langues appliquées Et d'informatique de Béja</p> <p>Département Informatique</p>	
---	--	---

Rapport De Projet De Fin d'Année

Présenté en vue de l'obtention de :

Licence National En Science De L'informatique

Spécialité : Génie Logiciel et système d'information (GLSI)

Sujet :

Conception et développement d'une
application web E-commerce

Elaboré par : Donia Gharbi

Encadré par : Ahmed Naffeti (entreprise)

Société d'accueil : Bee coders

L'année universitaire : 2022/2023

Dédicaces

Je dédie ce travail à :

Ma famille, en particulier mes parents et mes sœurs, pour leur amour, leur confiance, leurs conseils, et leur soutien inconditionnel.

Un immense merci pour tout personne qui m'a encouragé et m'aidé.

Je tiens tout particulièrement à remercier moi-même pour le travail que j'ai fait au fil des années et pour tous mes sacrifices pour réaliser tout ce là.

Merci d'être toujours là pour moi



Remerciements

Au terme de ce stage je tiens à exprimer mes respects mes connaissances et mes sincères remerciements aux personnels du BeeCoders. Je tiens à vous exprimer ma profonde gratitude et mon immense respect à Monsieur Ahmed Naffeti, mon encadrant professionnel au sein de BeeCoders, pour l'encadrement de mon travail et pour son soutien durant toute la période de mon stage.

Ainsi ceux qui m'ont aidé durant cette période pour effectuer mon stage dans les meilleures conditions, et je tiens également à remercier à mes formateurs pour les formations enrichissantes.

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail ici l'expression de nos sincères gratitude.

Sommaire

Introduction générale.....	1
Chapitre 1 : Présentation du cadre de projet.....	2
Introduction.....	2
I. Présentation de l'entreprise.....	2
II. Etude de l'existant.....	3
1. Description de l'existant.....	3
2. Critique de l'existant.....	4
3. Présentation de notre solution.....	4
III. Méthodologie adoptée.....	5
1. Choix de la méthode.....	5
2. Le langage de modélisation.....	6
Conclusion.....	6
Chapitre 2 : Analyse et spécification des besoins.....	7
Introduction.....	7
I. Présentation des acteurs.....	7
II. Spécifications des besoins fonctionnels.....	7
1. Présentation des cas d'utilisation des acteurs.....	7
2. Les diagrammes de cas d'utilisation.....	8
III. Spécifications des besoins non fonctionnels.....	12
Conclusion.....	12
Chapitre 3 : Conception.....	13
Introduction.....	13
I. Architecture du système.....	13
1. Architecture physique.....	13
2. Architecture logique.....	13
II. Modélisation statique.....	14
1. Dictionnaire de données.....	14
2. Règles de gestions.....	15
3. Diagramme de classe.....	17
III. Conception dynamique.....	17
1. Diagramme de séquences.....	17
Conclusion.....	20
Chapitre 4 : Réalisation.....	21
Introduction.....	21
I. Architecture adoptée.....	21
1. Environnement matériel.....	21
2. Environnement logiciel.....	21
3. Langages de programmation.....	23
4. Les principales interfaces.....	24
Conclusion.....	25
Conclusion générale.....	26

Liste des figures

Figure 1 : Logo de l'entreprise Bee coders	2
Figure 2 Page d'accueil de dabchy.tn.....	3
Figure 3 Page d'accueil de tayara.tn	3
Figure 4 Cycle de vie en cascade	5
Figure 5 Langage de modélisation UML.....	6
Figure 6 Diagramme de cas d'utilisation relatif à l'acteur visiteur.....	8
Figure 7 Diagramme de cas d'utilisation relatif à l'acteur client.....	9
Figure 8 Diagramme de cas d'utilisation relatif à l'acteur propriétaire magasin	10
Figure 9 Diagramme de cas d'utilisation relatif à l'acteur livreur	10
Figure 10 : Diagramme de cas d'utilisation relatif à l'acteur administrateur	11
Figure 11 les besoins non fonctionnels.....	12
Figure 12 : Architecture 3tiers.....	13
Figure 13 : Architecture d'angular	13
Figure 14 : diagramme de classe	17
Figure 15. Diagramme de séquence du cas d'utilisation « Chercher produit »	18
Figure 16: Diagramme de séquence du cas d'utilisation « s'authentifier » pour client	18
Figure 17 : Diagramme de séquence du cas d'utilisation « Modifier utilisateur » pour administrateur ...	19
Figure 18 : Diagramme de séquence du cas d'utilisation « Valider livraison » pour Livreur	19
Figure 19 : Diagramme de séquence du cas d'utilisation « Ajouter produit » pour Propriétaire de la boutique.....	20
Figure 20 : Environnement Matériel.....	21
Figure 21 : L'éditeur de texte visual studio code	21
Figure 22: Logo de framework angular.....	22
Figure 23: Logo de Framework node.js.....	22
Figure 24 : Logo de MySql	22
Figure 25 : Logo Xampp	22
Figure 26 : Logo de Postman	23
Figure 27 : Logo Html.....	23
Figure 28: Logo Css	23
Figure 29 : Logo Javascript.....	23
Figure 30 : Logo Typescript.....	24
Figure 31: Interface Tableau de bord pour l'administrateur	24
Figure 32 : Interface gestion des utilisateurs pour administrateur.....	24
Figure 33 : interface compte pour client	24
Figure 34 interface créer boutique pour utilisateur.....	25

Les tableaux

Tableau 1: Fiche signalétique de Bee Coders	2
Tableau 2 : Les points faibles et points forts de l'existant	4
Tableau 3 : Les points faibles et points forts de la solution proposée	5
Tableau 4 : Liste des acteurs et leurs cas d'utilisations	8
Tableau 5 : Description textuelle de l'inscription	9
Tableau 6 : Description textuelle de consulter commandes pour client	9
Tableau 7 : Description textuelle de l'ajout de produit pour propriétaire magasin..	10
Tableau 8: Description textuelle de Consulter commande non livrées pour livreur	11
Tableau 9 : Description textuelle de l'ajout d'un utilisateur pour administrateur	11
Tableau 10 : Dictionnaire de données	15

Introduction générale

De nos jours, en Tunisie, le commerce électronique n'a pas encore assez développé depuis la pandémie du covid 19 désormais, le commerce électronique est devenu l'alternative pour le tunisien qui s'est trouvé plongé dans une période de confinement inédit et un couvre-feu limitant sa circulation. Il a donc été obligé de changer son comportement et de tourner peu à peu vers les achats en ligne. Ainsi, certains veulent travailler et commercer son propre projet à la maison et commencer la vente en ligne d'où l'apparition de la notion des boutiques virtuelles ce qui facilite aux personnes la départ de leur projet et ainsi gagner plus de l'argent. En plus, parmi les difficultés qui envisagent les propriétaires de projet le problème de marketing rien ne peut nier que l'apparition dans des sites bien référencés et de grande popularité peut nous aider à augmenter le nombre des clients et gagner leur confiance. Finalement, on peut citer le problème de livraison car elle aussi présente un obstacle devant certains au niveau des échanges des besoins ainsi pour être dominant à une surface spécifique.

Le présent rapport, qui expose ce travail, est composé de trois chapitres structurés comme suit :

- Dans le premier chapitre, nous allons présenter le cadre et l'objectif de ce projet, l'étude de l'existant, de son critique et des solutions proposées.
- Le deuxième chapitre sera consacré à l'analyse des besoins fonctionnels et non fonctionnels et à la conception de ce projet, nous allons présenter quelques diagrammes
- Dans le troisième chapitre, je vais étudier l'implémentation de l'application, en décrivant l'environnement matériel et logiciel, et je vais donner un aperçu sur les interfaces réalisées
- Le dernier chapitre, je vais étudier l'implémentation de l'application, en décrivant l'environnement matériel et logiciel, et je vais donner un aperçu sur les interfaces réalisées.
- Nous terminerons ce rapport avec une conclusion générale qui contiendra notre avis sur cette expérience ainsi que nos perspectives sur le projet.

Chapitre 1 : Présentation du cadre de projet

Introduction

Dans le chapitre présent, nous découvrirons le cadre général de notre projet en suivant ces étapes :

- Nous présentons l'entreprise d'accueil brièvement
- Nous étudierons l'existant du projet.
- Nous ferons part de notre méthodologie à adopter.

I. Présentation de l'entreprise d'accueil

Notre stage de fin d'année s'est déroulé au sein de l'entreprise « Bee Coders ». C'est un bureau informatique professionnel en développement des sites web, des applications mobiles, des logiciels, IoT, etc. C'est une société qui est à l'écoute des besoins du marché.



Figure 1 : Logo de l'entreprise Bee coders

Bee Coders a été créée en 2020.

Nom de l'entreprise	Bee Coders
Date de fondation	2020
Secteurs d'activité	Activités informatique
Siège sociale	Parc Technologique El Ghazela ,Ariana 2088
Adresse e-mail	Admin@beecoders.tn
Téléphone	+216 58 840 064

Tableau 1: Fiche signalétique de Bee Coders

Les Secteurs d'activité :

[1] Bee Coders c'est une boîte de développement et de services informatiques, spécialisée dans :

- La création et développement web et mobile (Android et IOS),
- Le développement des logiciels sur-mesure,
- La conception des objets connectés et des services intelligents,

- Le développement d'un système embarqué sur mesure,
- Le design web,
- Le consulting IT,
- Les formations à distance.

II. Etude de l'existant

Dans cette partie nous allons expliquer la problématique et exposer la solution proposée

1. Description de l'existant

a) Dabchy

Dabchy est une plateforme de mise en relation entre des vendeurs et des acheteurs.

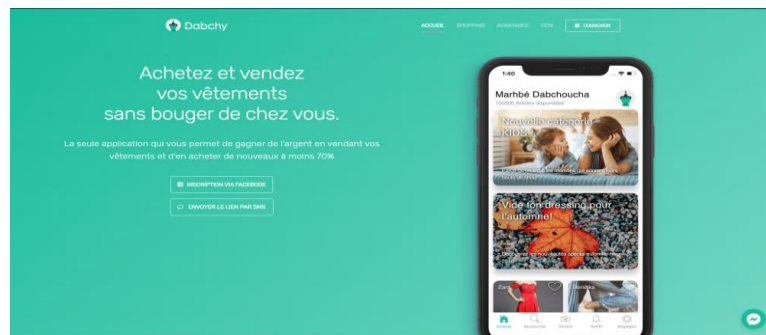


Figure 2 Page d'accueil de dabchy.tn

Description :

- ✓ Cette application existe en deux versions web et mobile elle présente des interfaces conviviales et lisibles à tous les utilisateurs, et offre plusieurs fonctionnalités à l'utilisateur (Se connecter, S'inscrire, Publier des produits, Commander des produits, Chercher des produits, Consulter les profils des autres vendeurs)

b) Tayara

Tayara est un site web qui assure une liaison entre des vendeurs et des acheteurs.

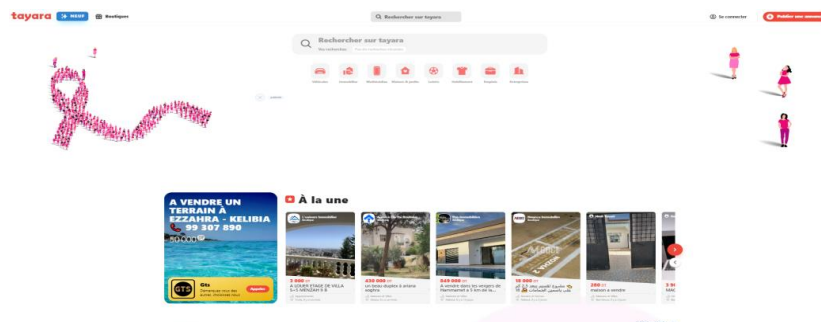


Figure 3 Page d'accueil de tayara.tn

Description :

- ✓ Ce site présente des interfaces conviviales et lisibles à tous les utilisateurs, et offre les fonctionnalités suivantes à l'utilisateur (Se connecter, S'inscrire, Publier des produits, Commander des produits, Publier des produits, Chercher des produits, Consulter les offres publiées, Créer et consulter des boutiques (Nouvelle fonctionnalités)).

2. Critique de l'existant

Dans cette partie, nous recensons tous les points faibles des 2 applications web citées précédemment.

Nom de l'application web	Points faibles	Points forts
Dabchy.tn	<ul style="list-style-type: none">-La confirmation d'un produit se fait parfois automatiquement.-La livraison est très lente-Les ventes se limitent seulement sur les vêtements-le client ne peut pas échanger des messages avec le vendeur seulement dans les commentaires.	<ul style="list-style-type: none">-Les pages web sont responsives.-le client peut évaluer un vendeur.-le site web est facile à utiliser.-le design du site est attirant.-Chaque vendeur a un grade selon le nombre des articles qui a vendus.
Tayara.tn	<ul style="list-style-type: none">-Le design du site est simple et classique.-Le client ne peut pas échanger des messages avec le vendeur.-Le site ne prend pas en considération la livraison.-les ventes ne sont pas totalement sécurisé.	<ul style="list-style-type: none">-les pages web sont responsives-Le client peut faire une recherche détaillée.-l'utilisateur peut créer une boutique.

Tableau 2 : Les points faibles et points forts de l'existant

3. Présentation de notre solution :

Notre solution est une application web améliore les fonctionnalités des applications existantes.

Points forts	Points faibles
<ul style="list-style-type: none">-Chaque utilisateur peut créer sa propre boutique.-Chaque produit doit être confirmé avant de débiter une livraison.-Assurer une livraison à domicile.-Le paiement sera en ligne ou lors de la réception.-Le client peut contacter le vendeur via message ou commentaire ou par téléphone.-Rapidité des services et de livraison	<ul style="list-style-type: none">-Les fonctionnalités ne sont pas totalement complètes, elles vont subir une amélioration au cours de temps et au niveau de paiement en ligne.

<ul style="list-style-type: none"> -L'utilisateur peut trouver une opportunité de travail. -l'application sera facile à utiliser. -les interfaces utilisateurs sont attractives. -Les clients fidèles auront des livraisons gratuites. -Chaque boutique peut être qualifiés et évolues par les utilisateurs pour assurer une transparence totale. -Le client peut faire retour une commande non satisfaisable et non conforme avec les détails citées à la description. -Le vendeur doit mentionner l'état de produit vendus obligatoirement avec des photos réelles de produits ainsi d'une photo des défauts s'ils existent. -Un utilisateur peut créer un ou plusieurs boutiques. 	
--	--

Tableau 3 : Les points faibles et points forts de la solution proposée

III. Méthodologie adoptée

Dans cette section, nous citerons la méthodologie qui sera utilisé dans le développement de cette application en justifiant la raison de notre choix.

1. Choix de la méthode

Nous choisissons la méthode du cycle de vie en cascade et ce choix se justifie par les arguments ci-dessous :

- L'idée de projet est claire et stable d'où nous n'avons pas besoins de changer les besoins en cours de développement.
- Chaque phase ne commence qu'une fois les résultats de la phase précédente ont été validés.
- Le point fort de cette approche est de garantir l'existence d'une documentation bien structurée.
- En cas d'erreurs, nous pouvons revenir à l'étape précédente.

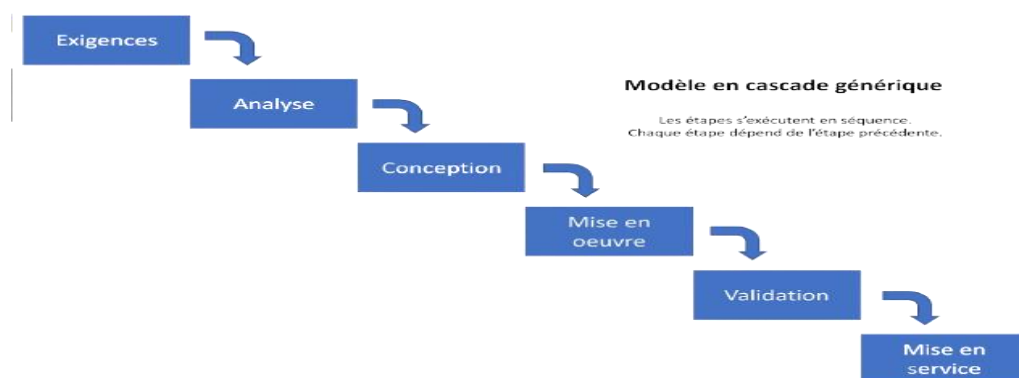


Figure 4 Cycle de vie en cascade

2. Le langage de modélisation

Pour pouvoir faciliter la création de l'application, nous utiliserons le langage de modélisation **UML**.



Figure 5 Langage de modélisation UML

Le langage de modélisation unifié (**UML**), est un langage de modélisation graphique à base de pictogrammes conçu une méthode de visualisation dans les domaines du développement logiciel et en conception orientée objet.

Conclusion

Dans ce chapitre, nous avons exposé le contexte général du projet. Nous avons présenté l'organisme d'accueil ainsi que la problématique et la solution proposée pour y remédier. Dans le chapitre suivant, il sera question de spécifier les besoins fonctionnels et non fonctionnels de notre application tout en élaborant les diagrammes nécessaires pour cette phase.

Chapitre 2 : Analyse et spécification des besoins

Introduction

A la suite de l'étude de l'existant et la proposition d'une solution, nous découvrons ensemble le projet plus en détails coté fonctionnalité pour mettre ce dernier dans son contexte général.

Pour arriver à nos fins, nous présenterons dans ce présent chapitre les besoins fonctionnels, les besoins non fonctionnels et enfin les cas d'utilisation des différents acteurs de notre application web.

I. Présentation des acteurs

Dans cette partie du présent chapitre, nous nous intéressons aux besoins fonctionnels de notre application web ainsi que les besoins non fonctionnels.

Avant de présenter les fonctionnalités nous présentons les acteurs.

Description d'un visiteur :

- Un visiteur est toute personne qui visite notre application à la recherche d'un service ou un produit ou une opportunité de travail.

Description d'un client :

- Ce dernier peut chercher un produit ou un magasin selon la catégorie.
- Il peut éventuellement commander un produit et payer en ligne.

Description d'un propriétaire du magasin :

- Un propriétaire du magasin est toute personne qui peut créer une boutique
- Il peut consulter, publier, modifier, supprimer ces propres produits.

Description d'un livreur :

- Un livreur peut consulter les commandes non livrés et livrés ainsi il peut valider, retarder ou annuler la livraison.

Description d'un administrateur :

- Un administrateur est le profil de la personne qui administrera l'application web.

II. Spécification des besoins fonctionnels :

1. Présentation des cas d'utilisations des acteurs :

Notre future application doit permettre les actions suivantes pour chaque utilisateur :

Acteur	Cas d'utilisation
Visiteur	<ul style="list-style-type: none">• Consulter site• Créer boutique• Créer compte• Chercher une opportunité de travail

	<ul style="list-style-type: none"> • Chercher un produit ou magasin
Client	<ul style="list-style-type: none"> • Commander produits • Gérer panier • Gérer liste de favoris • Modifier profil
Propriétaire de magasin	<ul style="list-style-type: none"> • Consulter clients • Consulter commentaires • Gérer produits • Gérer profil • Contacter l'administrateur • Gérer commandes
Livreur	<ul style="list-style-type: none"> • Consulter commandes livrées • Gérer commandes non livrées • Gérer commandes en retour • Gérer profil
Administrateur	<ul style="list-style-type: none"> • Gérer profil • Gérer utilisateurs • Gérer livreurs • Gérer commandes • Gérer boutique • Gérer demandes d'emplois

Tableau 4 : Liste des acteurs et leurs cas d'utilisations

2. Les diagrammes de cas d'utilisation :

Dans cette partie, nous découvrirons ensemble les diagrammes de cas d'utilisation des acteurs les plus importants dans notre application web.

a) Visiteur :

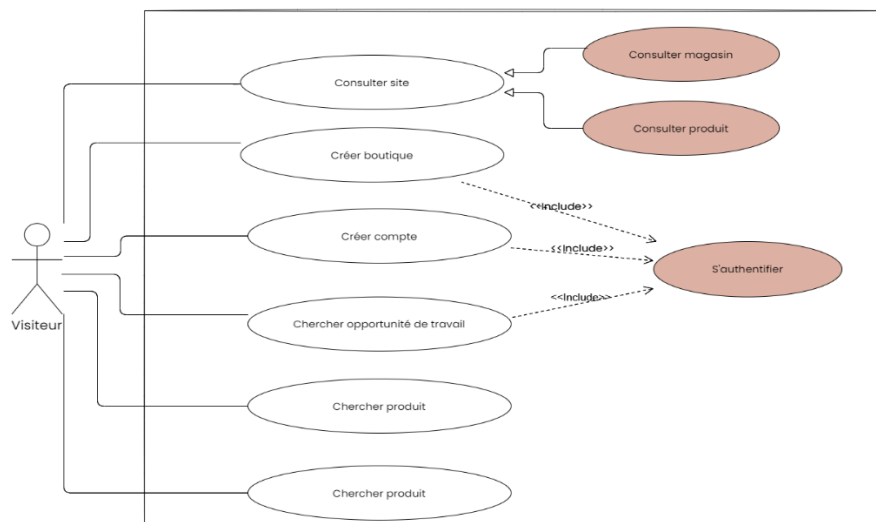


Figure 6 Diagramme de cas d'utilisation relatif à l'acteur visiteur

Description :

Ce diagramme nous décrit toutes les fonctionnalités qui sont disponibles à n'importe personne qui visite notre plateforme.

Cas n	1
-------	---

Acteur(s)	Visiteur
Objectif :	S'inscrire
Précondition (s):	Aucune
Postcondition(s)	Aucune
Scénario nominal :	<ol style="list-style-type: none"> 1. L'invité demande de s'inscrire 2. L'application affiche un formulaire d'inscription 3. L'invité remplit le formulaire 4. L'invité demande de créer son compte 5. L'application crée le compte 6. L'application affiche un pop-up de succès de l'inscription

Tableau 5 : Description textuelle de l'inscription

Visual Paradigm Online Free Edition

Client :

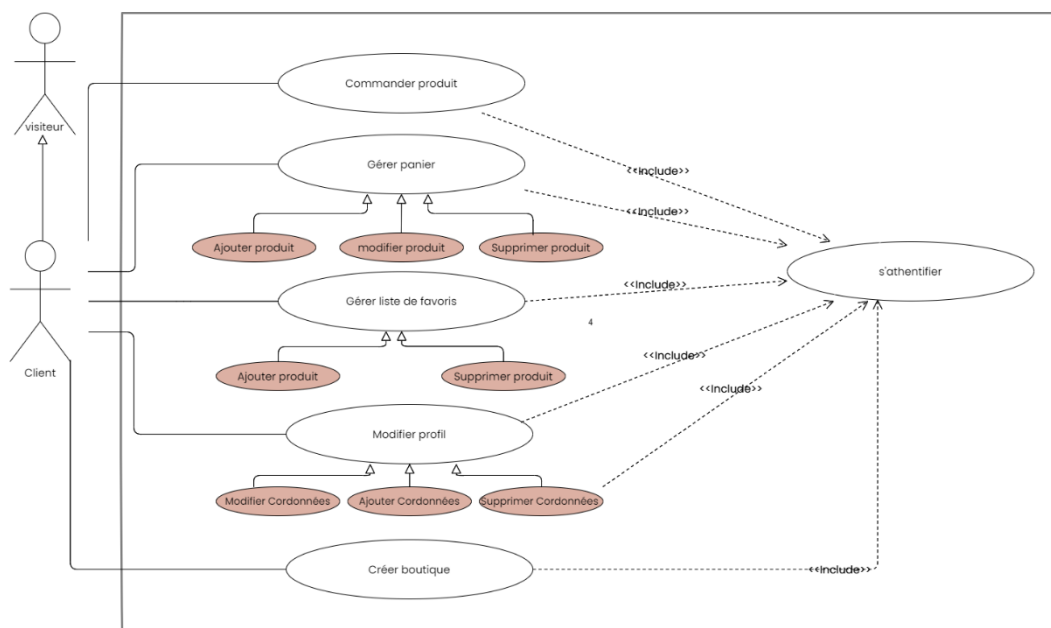


Figure 7 Diagramme de cas d'utilisation relatif à l'acteur client

Visual Paradigm Online Free Edition

Description :

Cas n :	1
Acteurs(s)	Client
Objectif :	Consulter commandes
Préconditions :	Être connecté
Postconditions :	Aucun
Scénario nominal :	<ol style="list-style-type: none"> 1. Le client demande d'afficher la liste des commandes dans le panier 2. L'application affiche la liste des commandes du client.

Tableau 6 : Description textuelle de consulter commandes pour client

Propriétaire magasin :

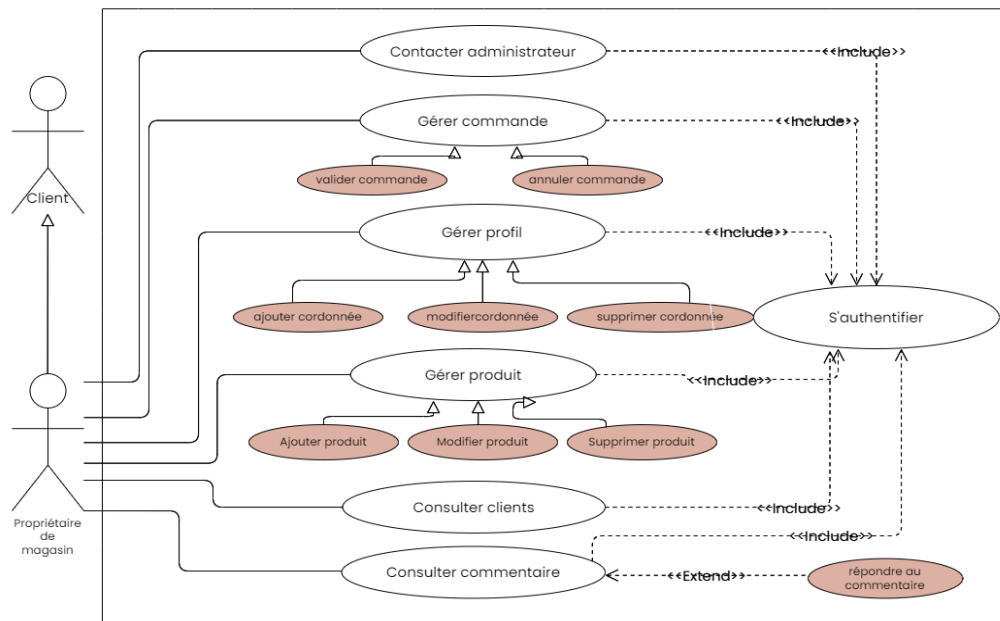


Figure 8 Diagramme de cas d'utilisation relatif à l'acteur propriétaire magasin

Description :

Cas n	1
Acteur (s)	Propriétaire du magasin
Objectif :	Ajouter produit
Préconditions :	Être connecté
Postconditions :	Aucune
Scénario nominal :	<p>1.Le propriétaire du magasin demande d'ajouter un produit.</p> <p>2. La plateforme affiche un formulaire.</p> <p>3.Ce dernier remplit le formulaire et demande d'effectuer l'ajout.</p> <p>4.L'application ajoute le produit et affiche la page précédente.</p>

Tableau 7 : Description textuelle de l'ajout de produit pour propriétaire magasin

Visual Paradigm Online Free Edition

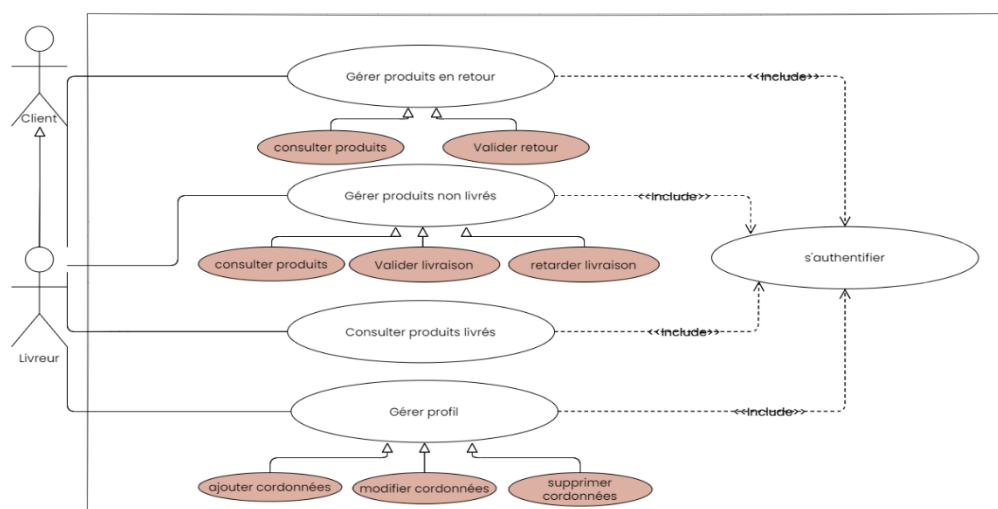
Livreur :

Figure 9 Diagramme de cas d'utilisation relatif à l'acteur livreur

Description :

Cas n	1
Acteurs(s)	Livreur
Objectif	Consulter les commandes non livrées
Préconditions :	Être connecté
Postconditions :	Aucun
Scénario nominal :	1.Le livreur demande de consulter les commandes livrées. 2. La plateforme affiche la liste des commandes.

Tableau 8: Description textuelle de Consulter commande non livrées pour livreur

Administrateur :

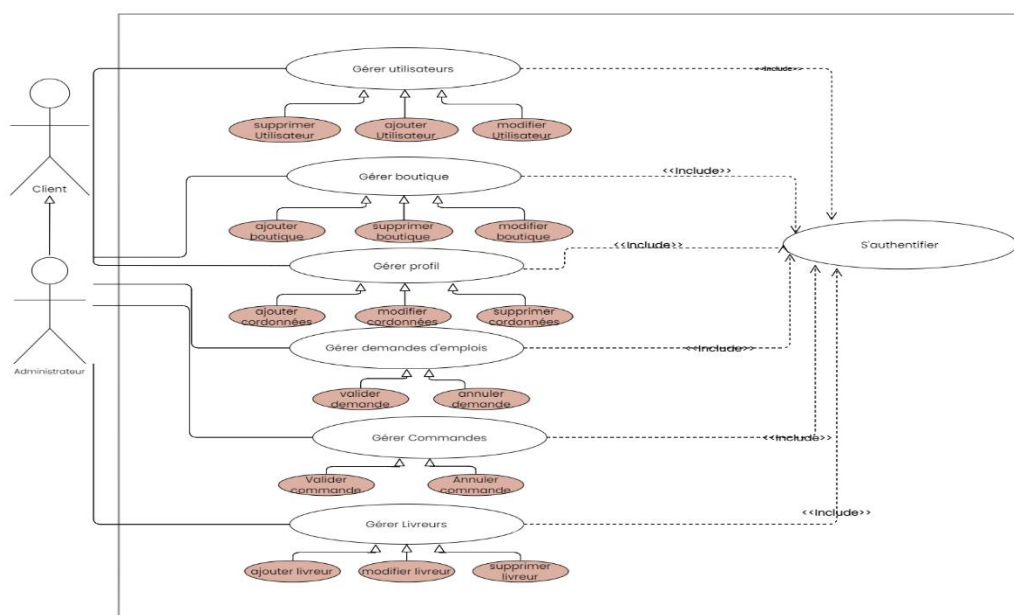


Figure 10 : Diagramme de cas d'utilisation relatif à l'acteur administrateur

Description :

Cas n	1
Acteur(s)	Administrateur
Objectif	Ajouter utilisateur
Préconditions :	Être connecté
Postconditions :	Utilisateur ajouté
Scénario nominal :	1.l'administrateur demande d'ajouter un produit. 2. La plateforme affiche un formulaire. 3.Ce dernier remplit le formulaire et demande d'effectuer l'ajout. 4.L'application ajoute l'utilisateur et affiche la page précédente.

Tableau 9 : Description textuelle de l'ajout d'un utilisateur pour administrateur

III. Spécification des besoins non fonctionnels

La qualité de notre logiciel est considérée comme le facteur le plus important dans notre projet afin d'avoir un produit parfait avec les meilleures fonctionnalités. Pour cela on se base sur la norme ISO 9126 pour définir toutes les règles à appliquer.

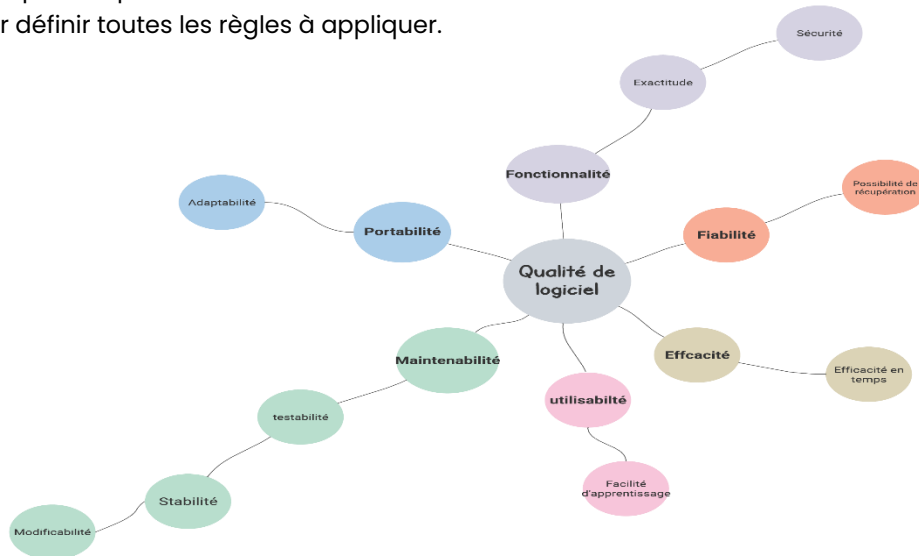


Figure 11 les besoins non fonctionnels

Fonctionnalité :

- Exactitude : Le résultat de notre projet doit être exact et convenable aux besoins spécifiés.
- Sécurité : L'application doit assurer la confidentialité des données et définir les niveaux d'accès possibles pour chaque utilisateur.

Maintenabilité :

- Stabilité : L'application doit être capable de maintenir la stabilité même lors d'une modification et prévoir le nombre d'erreurs empêchant le fonctionnement des services qui sont les bugs bloquant l'application.
- Modificabilité : Lors d'une modification bien spécifiée, l'application doit assurer son implémentation facilement grâce à l'architecture 3tiers qui assure une meilleure structuration du projet.
- Testabilité : Le produit doit être valide par rapport à la spécification des besoins.

Utilisabilité :

- Facilité d'apprentissage : Le portail doit être ergonomique et bien organisé afin de minimiser l'effort de l'utilisateur.

Efficacité :

- Efficacité en temps : Les requêtes des utilisateurs sont traitées plus vite grâce à l'architecture SPA single page application.

Portabilité :

- Adaptabilité : L'application doit être conforme à n'importe quel environnement de travail donc fonctionnelle sur n'importe quel type de système d'exploitation.

Fiabilité :

- Possibilité de récupération : En cas de défaillance, le produit doit récupérer les données et les services défaits en un intervalle de temps minimal.

Conclusion

Dans ce deuxième chapitre, Nous avons présenté les différentes fonctionnalités que nous réaliserons dans ce projet ainsi qu'une spécification non fonctionnelle. Dans le prochain chapitre on va découvrir l'architecture de ce projet ainsi que les différents modèles (classes).

Chapitre 3 : Conception

Introduction

À la suite de l'analyse des besoins et leurs spécifications, nous découvrirons dans ce présent chapitre tout d'abord l'architecture globale que nous utiliserons dans le développement de notre projet.

Ensuite, nous ferons part également le diagramme de classe de notre projet en détails, et finalement nous verrons quelques diagrammes de séquence suivant l'architecture choisie.

I. Architecture du système :

1. Architecture physique :

Pour réaliser une application web avec un client léger et bien sécuriser les données, nous avons opté pour l'architecture 3tiers.

Le rôle de ce modèle est clairement défini :

- La présentation des données (niveau 1 ou 1^{er} tiers) : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur.
- Le traitement métier des données (niveau 2 ou 2^{ème} tiers) : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- L'accès aux données persistantes (niveau 3 ou 3^{ème} tiers) : correspondant aux données qui est destinées à être conservées sur la durée, voir de manière définitive.



Figure 12 : Architecture 3tiers

Ce modèle d'architecture 3 tiers a pour objectif de répondre aux préoccupations suivantes :

- Allégement du poste de travail client ;
- Prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.) ;
- Meilleure répartition de la charge entre différents serveurs d'applications.

2. Architecture logique d'angular :

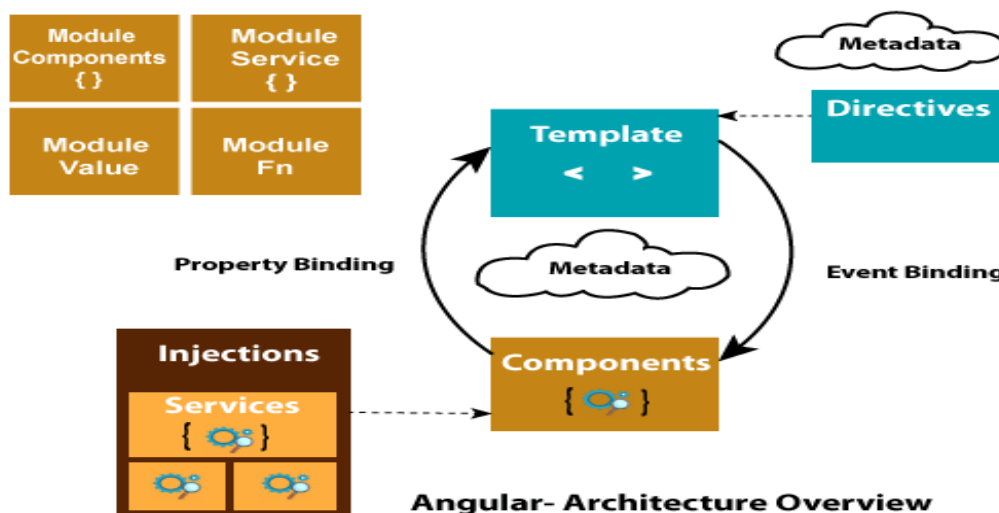


Figure 14 : architecture d'angular

Composants - Template :

Le composant est la partie principale d'angular. Il s'occupe d'une partie de la vue. L'interaction entre le composant et la vue se fait à travers différents types de binding. Un Template est le complément du composant. C'est la vue associée au composant. Elle représente le code HTML géré par le composant.

Binding :

Le Binding est le mécanisme qui permet de mapper des éléments du DOM avec des propriétés et des méthodes du composant. Le Binding permettra également de faire communiquer les composants.

Les métadatas :

Les Metadata sont des informations permettant de décrire les classes.

Librairies : ► Ensemble de modules JS qui contiennent un ensemble de fonctionnalités.

II. Modélisation statique :

1. Dictionnaire de données :

Le tableau ci-dessous contient toutes les classes constituant la base de données de l'application ainsi qu'une description de leurs attributs.

Table	Code	Libellé	Type
Compte	IdCompte role	Identificateur de compte Type de rôle compte	Int String
Utilisateur	IdUser	Identificateur de l'utilisateur	int
Demande d'emplois	idEmplois nom email numTel adresse: option	Identificateur de demande Nom de personne Email de personne Numéro de personne Adresse de personne Option choisis	Int String String Int String String
Administrateur	idProp nom num email adresse motPasse	Identificateur de l'administrateur Nom de l'administrateur Numéro de l'administrateur Email de l'administrateur Adresse de l'administrateur Mot de passe de l'administrateur	Int String Int String String String
liste de favoris	idListe idprod	Identificateur de la liste Identificateur des produits	Int int

Commande	idCom idB description boutique prix photo nomClient lieuBoutique adresseClient numTelProp numTelClient	Identificateur de commande Identificateur de boutique Description de commande Nom de boutique Prix de commande Photo de la commande Nom de client Lieu de boutique Adresse de client Numéro de propriétaire de boutique Numéro de client	Int Int String String Float String String String String String Int Int
Client	idClient nom email numTel adresse motPasse	Identificateur de client Nom de client Email de client Numéro de client Adresse de client Mot de passe de client	Int String String Int String String
Produit	Idprod nom nomPropri description catégorie boutique prix photo	Identificateur de produit Nom de produit Nom de propriétaire boutique Description de produit Catégorie de produit Nom de boutique Prix de produit Photo de produit	Int String String String String String Float String
Boutique	IdB nom nomPropri catégorie lieu motPasse	Identificateur de boutique Nom de boutique Nom de propriétaire de boutique Catégorie de boutique Lieu de boutique Mot de passe lié à ce boutique	Int String String String String String
Facture	IdFac description idClient prfac	Identificateur de facture Description de facture Identificateur de client Prix de facture	Int String Int Float
Livreur	IdLiv nom email numTel adresse motPasse	Identificateur de livreur Nom de livreur Email de livreur Numéro de livreur Adresse de livreur Mot de passe de livreur	Int String String Int String String
Propriétaire de magasin	IdProp nom nomBoutique numTel adresse motPasse	Identificateur de propriétaire de magasin Nom de propriétaire de magasin Nom de boutique Numéro de propriétaire Adresse de propriétaire Mot de passe de propriétaire	Int String String String Int String String
Panier	idPanier idprod	Identificateur de panier Identificateur des produits	Int Int
Catégorie	id nom	Identificateur de catégorie Nom de catégorie	Int String

Tableau 10 : Dictionnaire de données

2. Règles de gestions :

Le diagramme de classe étudié est basé sur les règles de gestion suivantes :

R1 : Un client a un et un seul compte.

R2 : Un compte appartient à un et un seul client.

R3 : Un client peut passer une ou plusieurs commandes.

R4 : Une commande peut être passée par un et un seul client.
R5 : Un client a un et un seul panier.
R6 : Un panier appartient à un et un seul client.
R7 : Un client a une seule liste de favoris.
R8 : Une liste de favoris appartient à un et un seul client.
R9 : Une liste de favoris contient un ou plusieurs produits.
R10 : Un produit appartient à un ou plusieurs listes de favoris.
R11 : Une commande possède une seule facture.
R12 : Une facture appartient à une seule commande.
R13 : Un client paie une ou plusieurs factures.
R14 : Une facture est payée par un et un seul client
R15 : Une commande contient un ou plusieurs produits.
R16 : Un produit peut être trouvé dans une ou plusieurs commandes.
R17 : Une boutique contient un ou plusieurs produits.
R18 : Un produit peut être trouvé dans une ou plusieurs boutiques.
R19 : Une boutique possède une seule catégorie.
R20 : Une catégorie possède une ou plusieurs boutiques.
R21 : Une boutique a un ou plusieurs clients.
R22 : Un client appartient à un ou plusieurs boutiques.
R23 : Un livreur possède un et un seul compte.
R24 : Un compte appartient à un et un seul livreur.
R25 : Un livreur livre une ou plusieurs commandes.
R26 : Une commande peut être livrée par un et un seul livreur.
R27 : Un utilisateur effectue une demande d'emplois
R28 : Une demande d'emplois est effectuée par un et un seul utilisateur.
R29 : L'administrateur gère un ou plusieurs comptes.
R30 : Un compte est géré par un et un seul administrateur.
R31 : L'administrateur gère une ou plusieurs boutiques.
R32 : Une boutique est gérée par un et un seul administrateur.
R33 : L'administrateur gère une ou plusieurs demandes d'emplois.
R34 : Une demande d'emplois est gérée par un et un seul administrateur.
R35 : Un administrateur a un et un seul compte.
R36 : Un compte appartient à un et un seul administrateur.
R37 : L'administrateur gère une ou plusieurs commandes.
R38 : Une commande est gérée par un et un seul administrateur
R39 : Le propriétaire de boutique possède une seule boutique
R40 : Une boutique appartient à un et un seul propriétaire.
R41 : Le propriétaire de boutique gère une ou plusieurs commandes
R42 : Une commande est gérée par un et un seul propriétaire

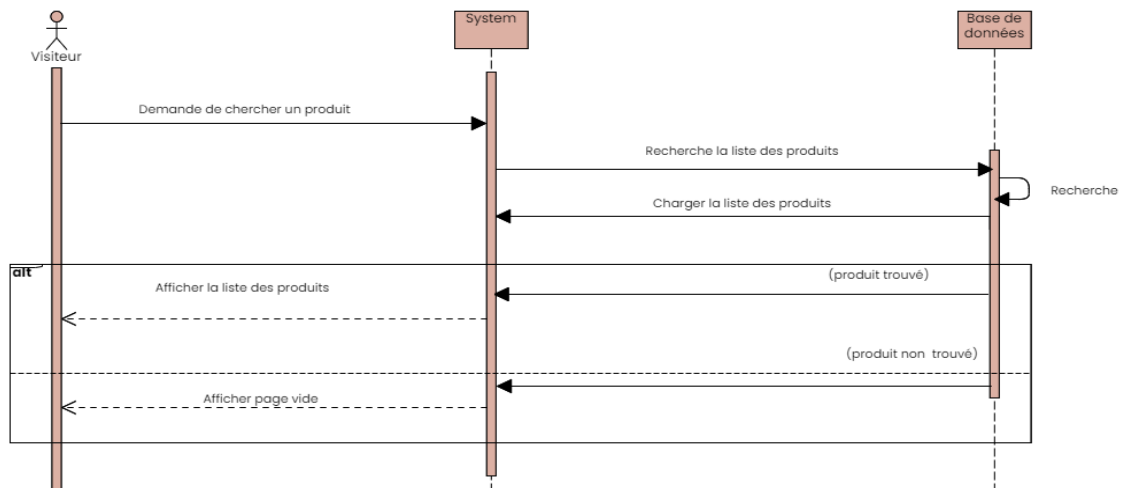
Visual Paradigm Online Free Edition



Description :

III. Conception dynamique :

a. Diagramme de séquence du cas d'utilisation « Chercher produit »



: Figure 16. Diagramme de séquence du cas d'utilisation « Chercher produit »

Description :

1. Le visiteur demande de chercher un produit.
2. Le système vérifie l'existence du produit dans la base de données.
3. Le système charge la liste des produits de la Base de données.
4. Si le produit existe déjà une liste sera affichée au visiteur.
5. Si le produit n'existe pas une liste vide sera affichée.

b. Diagramme de séquence du cas d'utilisation « s'authentifier » pour client

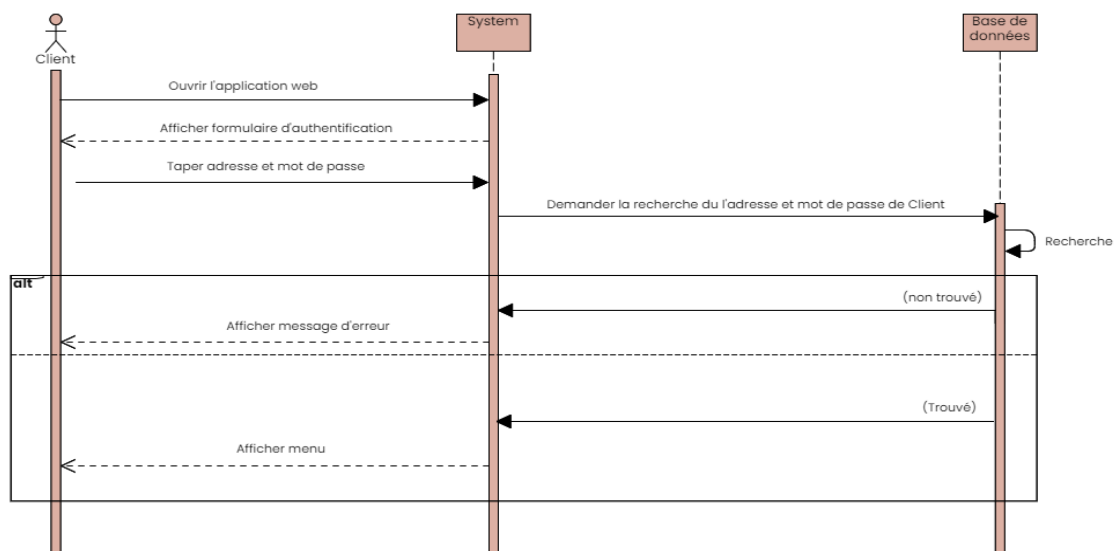


Figure 17: Diagramme de séquence du cas d'utilisation « s'authentifier » pour client

Description :

1. Le client ouvre l'application web.
2. Le système affiche la page d'authentification.
3. Le client fait entrer ces données dans le formulaire.
4. Le système vérifie les données entrées dans la base.

5. Si les données sont incorrectes le système affiche un message d'erreur.
7. Si les données sont correctes le système affiche le menu.

c. Diagramme de séquence du cas d'utilisation « Modifier utilisateur » pour administrateur

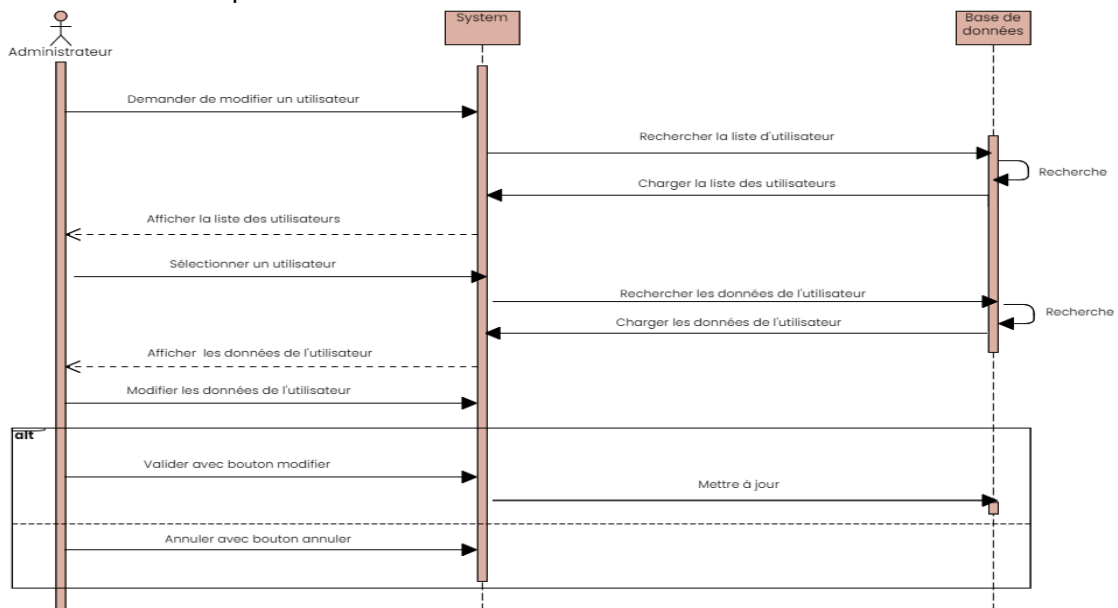


Figure 18 : Diagramme de séquence du cas d'utilisation « Modifier utilisateur » pour administrateur

Description :

1. L'administrateur choisit l'interface de la liste des utilisateurs.
2. Le système cherche la liste des utilisateurs dans la base de données.
3. Le système affiche la liste des utilisateurs à l'administrateur.
4. L'administrateur sélectionne un utilisateur.
5. Le système cherche les données de l'utilisateur dans la base de données.
7. Le système affiche les données de l'utilisateur à l'administrateur.
8. L'administrateur modifie les données.
9. Si l'administrateur valide la modification les données seront mis à jour dans la base.
10. Si l'administrateur ne valide pas la modification les données ne seront pas mis à jour dans la base.

d. Diagramme de séquence du cas d'utilisation « Valider livraison » pour Livreur

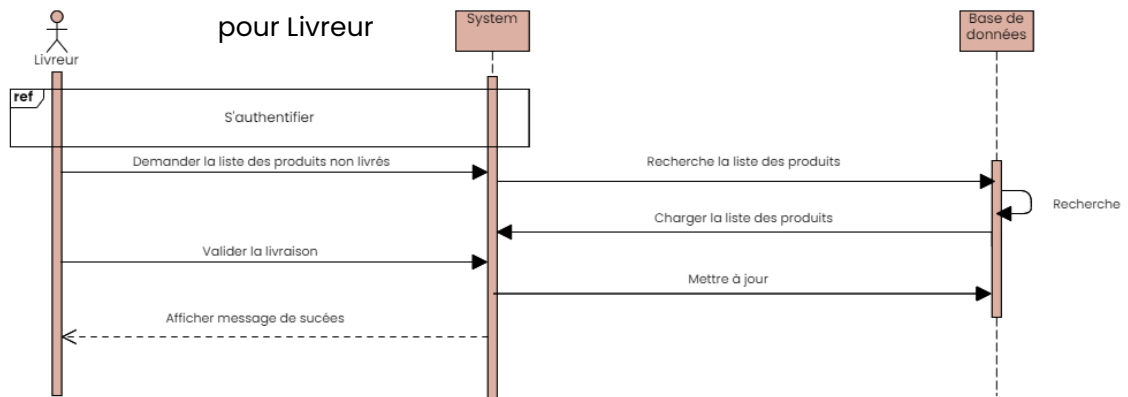


Figure 19 : Diagramme de séquence du cas d'utilisation « Valider livraison » pour Livreur

Description :

1. Le livreur se connecte à l'application web.
- 2.. Le livreur choisit l'interface de la liste des commandes non livrées.
3. Le système cherche la liste des commandes non livrées dans la base de données.
4. Le système affiche la liste des commandes non livrées au livreur.
5. Le livreur valide la livraison d'une commande.
6. Le système met à jour cette modification dans la base de données.
7. Le système affiche un message de succès au livreur.

e. Diagramme de séquence du cas d'utilisation « Ajouter produit » pour Propriétaire de la boutique.

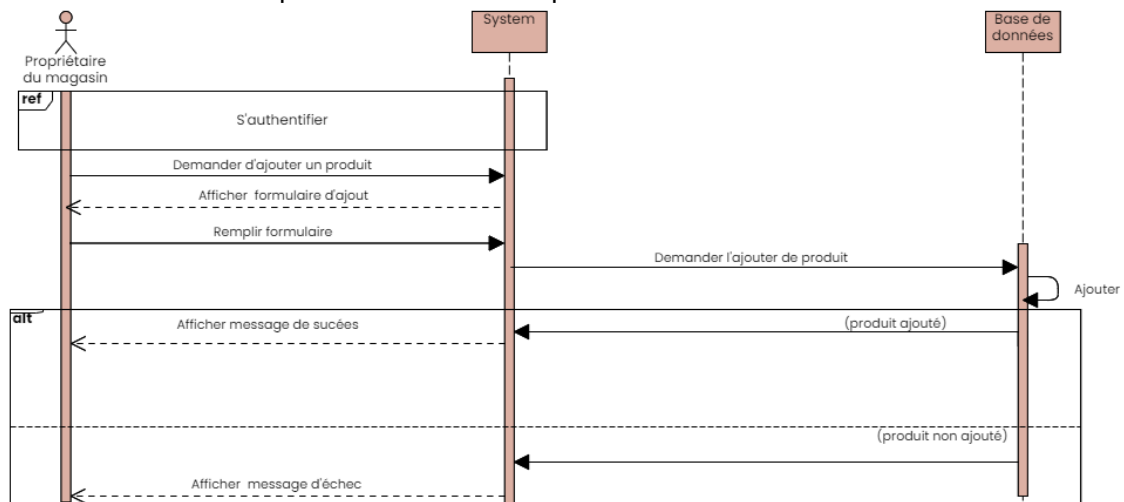


Figure 20 : Diagramme de séquence du cas d'utilisation « Ajouter produit » pour Propriétaire de la boutique.

Description :

1. Le propriétaire du magasin se connecte à l'application web.
- 2.. Le propriétaire du magasin demande d'ajouter un produit.
3. Le système affiche un formulaire d'ajout.
4. Le propriétaire du magasin remplit le formulaire.
5. Le système demande d'effectuer l'ajout de produit dans la base de données.
6. Si le produit est ajouté dans la base, un message de succès sera affiché.
7. Si le produit n'est pas ajouté dans la base, un message d'échec sera affiché.

Conclusion

Nous venons de terminer cette partie de conception, qui consiste à déterminer aussi bien l'architecture que nous avons adoptée ainsi que la modélisation technique par un diagramme de classe et dynamique par quelques diagrammes de séquence.

Chapitre 4 : Réalisation

Introduction

Dans ce dernier chapitre, nous présenterons le travail réalisé durant la phase d'implémentation de notre application.

L'architecture adoptée

Dans cette section, nous présenterons l'environnement de développement matériel et logiciel que nous avons utilisé pour réaliser notre application.

1. Environnement matériel :

Pour réaliser ce projet, nous avons utilisé un ordinateur qui comporte ses caractéristiques techniques suivantes :

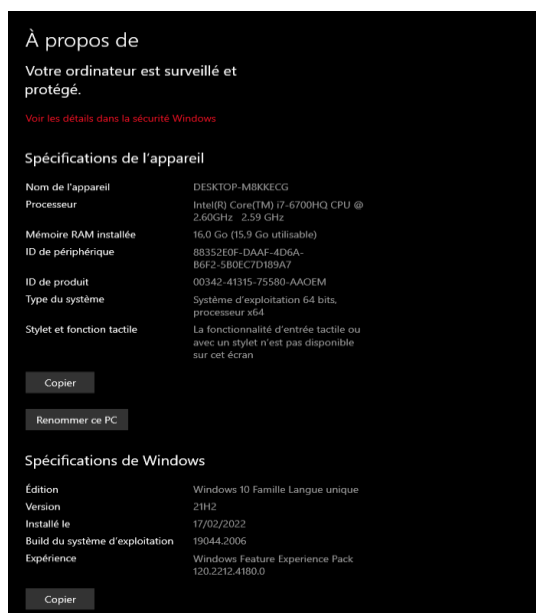


Figure 21 : Environnement Matériel

2. Environnement logiciel :

Pour développer notre application web, nous avons manié quelques environnements logiciels, que nous présentons dans ce qui suit :

a) Visual studio code



Figure 22 : L'éditeur de texte visual studio code

[2] Visual studio code est un éditeur de code léger développé par Microsoft mais il est puissant qui s'exécute sur votre bureau et il est disponible pour Windows, MacOS et linux.

b) angular



Figure 23: Logo de framework angular

[3] Angular est un framework (cadre de travail) Javascript

- Permet de développer des sites web, des applications web, des applications mobiles hybrides, de manière robuste et efficace.
- Supporte plusieurs langages : JS (ES5), TypeScript, Dart
- Développé par Google en 2009

c) nodeJs



Figure 24: Logo de Framework node.js

[4] Node.js est un Framework d'E / S asynchrones, non bloquant et basé sur des événements, qui utilise le moteur JavaScript V8 de Google. Il est open-source et multi-plateforme.

d) MySQL



Figure 25 : Logo de MySQL

[5] MySQL est un système de gestion de bases de données relationnelles (SGBDR) fonctionnant sous Windows et Linux. Il fait partie des logiciels de gestion de base des données les plus utilisées au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

e) Xampp



Figure 26 : Logo Xampp

[6]XAMPP est un projet open source à but non lucratif développé par Apache Friends. Son nom est un acronyme pour Cross-Platform (X), Apache, MySQL, PHP et Perl.

f) Postman



Figure 27 : Logo de Postman

[7]Postman permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en **Collections**.

3. Langues de programmation :

a) Html



Figure 28 : Logo Html

HTML signifie « HyperText Markup Langage » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure.

b) css



Figure 29: Logo Css

Css est l'un des langages principaux du Web ouvert et a été standardisé par le W3C. C'est le langage que nous utilisons pour styliser un document Html.

c) Javascript



Figure 30 : Logo Javascript

Javascript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

d) Typescript



Figure 31 : Logo TypeScript

TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript

Les principales interfaces



Figure 32: Interface Tableau de bord pour l'administrateur

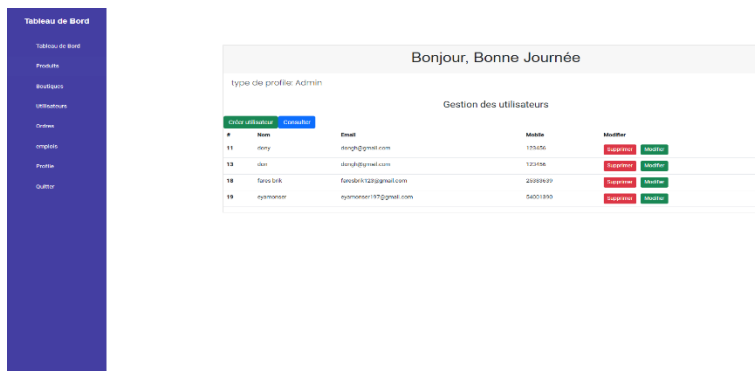


Figure 33 : Interface gestion des utilisateurs pour administrateur

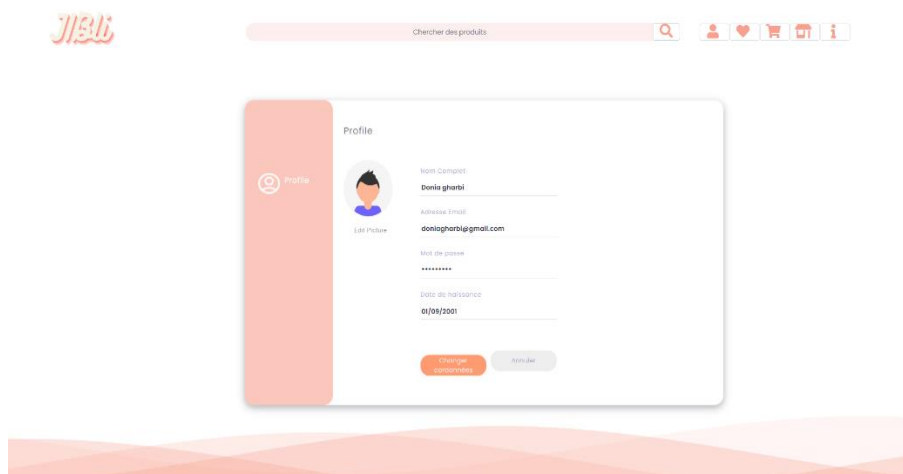


Figure 34 : interface compte pour client

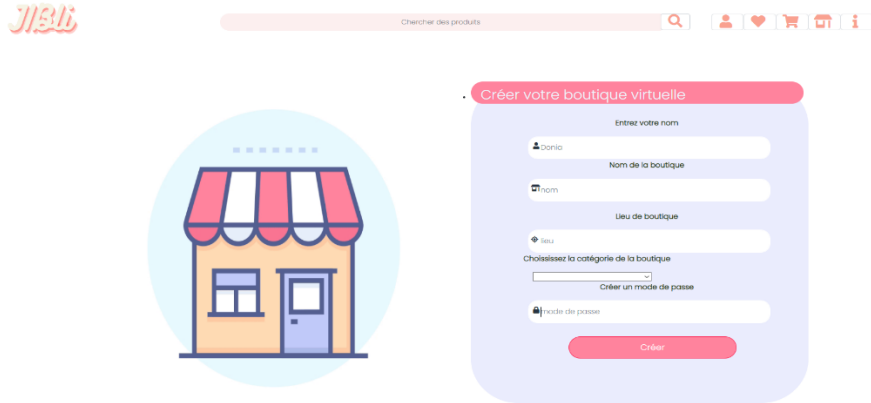


Figure 35 interface créer boutique pour utilisateur

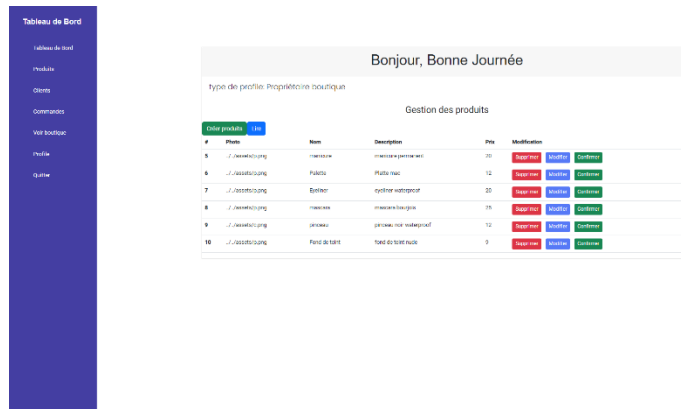


Figure 36: interface gérer produit pour propriétaire de magasin

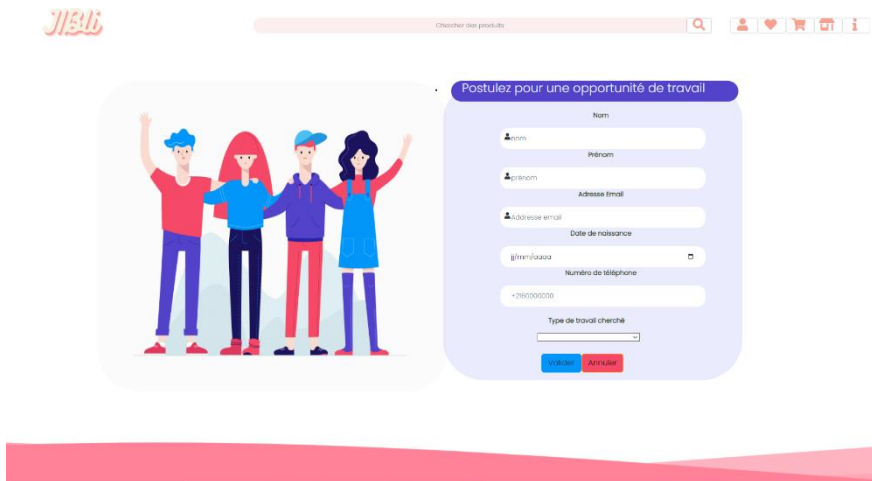


Figure 37 : interface de remplissage de formulaire pour une opportunité de travail

Conclusion

Dans ce dernier chapitre, nous avons vu l'environnement de développement que nous avons utilisé durant la réalisation de l'application web et puis comment l'application sera déployée. Finalement, nous avons présenté quelques interfaces de l'application web

Conclusion générale

Le travail s'articule de la conception et la réalisation d'une application web destinée à mettre en relation des vendeurs et des clients nommée « Jibli Delivery » au sein de la société Bee coders afin de clôturer notre licence de développement des systèmes informatique au sein de l'institut supérieur des langues appliquées et d'informatique Béja.

Ce rapport décrit le bilan de notre travail, nous avons procédé dans un premier temps à une présentation générale de projet pour bien comprendre la problématique et engendrer une solution optimale qui répond aux besoins des clients qui souhaitent commander ou vendre des produits en ligne, et aussi aux administrateurs de gérer la plateforme.

Par la suite, nous avons passé à la spécification des besoins qui nous a permis de définir les besoins fonctionnels et non fonctionnels de notre application web. Ensuite, nous avons passé à la conception de l'application web en suivant le cycle de vie en cascade. Puis, nous avons passé à la création des maquettes, et finalement nous avons terminé par la réalisation.

Ce travail était très intéressant puisqu'il nous a permis de découvrir un nouveau domaine de travail et de s'éloigner des projets traditionnels. Il nous a permis d'approfondir nos connaissances dans les bonnes pratiques de programmation et dans l'utilisation de nouvelles technologies telles que les Framework.

Pour conclure, notre travail ne s'arrête pas à ce stade. En effet, plusieurs fonctionnalités peuvent être ajoutées comme le paiement en ligne de la commande ou agrandir la plateforme pour la rendre supportant plusieurs pays ainsi de y intégrer un chatBot.

Bibliographie

- [1] «<https://www.beecoders.tn>,» [En ligne].
- [2] «<https://code.visualstudio.com>,» [En ligne].
- [3] N. JELASSI, Écrivain, *Présentation sur angular*. [Performance].
- [4] «<https://en.wikipedia.org/wiki/Node.js>,» [En ligne].
- [5] <https://www.data-bird.co/sql/mysql>. [En ligne].
- [6] «<https://blog.webnet.fr/>,» [En ligne].
- [7] <https://www.clubic.com/telecharger-fiche70674-xampp.html>. [En ligne].