|  |  |  |
| --- | --- | --- |
| IsetFinal_mai_2005%20copier2 | **Ministère de l'Enseignement Supérieur, de la Recherche Scientifique Direction Générale des Études Technologies** | 220px-Logo_reseaux_iset.svg.png |
| **Institut Supérieur des Études Technologiques de Djerba Département Technologies de l’informatique** | | |

|  |  |
| --- | --- |
| **Code projet** | **…………** |

**Rapport de**

**STAGE DE FIN D’ETUDES**

**Présenté en vue de l’obtention du diplôme de**

**Licence en Technologies de l’Informatique**

**Parcours : MDW**

**Participation dans le développement application mobile de prise de commande**

**Elaboré par :**

**Ben Daamer Oumayma**

**Hrich Saber**

**Encadré par :**

**Mr.Taher Haout**

**Effectué à :**

**Entreprise : TOUT EST LA**

**Encadreur : Mr. Moez ben otheman**

**Année universitaire : 2021/2022**

Sommaire

[Introduction générale 1](#_Toc105940188)

[Chapitre 1 : Présentation générale 2](#_Toc105940189)

[Introduction 2](#_Toc105940190)

[1.1 Cadre de travail 2](#_Toc105940191)

[1.2 Présentation de l’organisme d’accueil 2](#_Toc105940192)

[1.2.1 Service de l’organisme d’accueil 3](#_Toc105940193)

[1.2.2 Histoire de l’organisme d’accueil 3](#_Toc105940194)

[1.3 Présentation de l’application 4](#_Toc105940195)

[1.3.1 Etude de l’existant 4](#_Toc105940196)

[1.3.2 Solution proposée 4](#_Toc105940197)

[Conclusion 4](#_Toc105940198)

[Chapitre 2 : Etude conceptuel 5](#_Toc105940199)

[Introduction 5](#_Toc105940200)

[2.1 Maquet de l’application 5](#_Toc105940201)

[2.2 Conception UML 6](#_Toc105940202)

[2.2.1 Diagramme de cas d’utilisation 6](#_Toc105940203)

[2.2.2 Diagramme de classe 9](#_Toc105940204)

[2.2.3 Diagramme de séquence 10](#_Toc105940205)

[2.2.4 Diagramme d’activité 13](#_Toc105940206)

[Conclusion 13](#_Toc105940208)

[Chapitre 3: Réalisation 14](#_Toc105940209)

[Introduction 14](#_Toc105940210)

[3.1 Environnements de travaille 14](#_Toc105940211)

[3.1.1 Environnement matériel 14](#_Toc105940212)

[3.1.2 Environnement logiciel 14](#_Toc105940213)

[3.1.3 Technologies utilisé 17](#_Toc105940214)

[3.2 Tache réalise 18](#_Toc105940215)

[3.2.1 Création du logo 18](#_Toc105940216)

[3.2.2 Création de base de données 19](#_Toc105940217)

[3.2.3 Réalisation API 21](#_Toc105940218)

[3.2.4 Configuration d’IIS pour héberger un api .net core 24](#_Toc105940219)

[3.2.5 Réalisation du l’application ionic 26](#_Toc105940220)

[3.3 Présentation des interfaces d’application 29](#_Toc105940221)

[Conclusion 33](#_Toc105940224)

[Conclusion générale 34](#_Toc105940225)

[Bibliographie & Nétographique 35](#_Toc105940226)

[Annexe 36](#_Toc105940227)

List des tableaux

[Tableau 1: Description de cas d'utilisation d'agent commercial 8](#_Toc105939781)

[Tableau 2: Description de cas d'utilisation d'administrateur 9](#_Toc105939782)

[Tableau 3: Environnements matériels 14](#_Toc105939783)

[Tableau 4: Description des classes 37](#_Toc105939784)

[Tableau 5: Les attributs du l’utilisateur 38](#_Toc105939785)

[Tableau 6: Les attributs du rôle 38](#_Toc105939786)

[Tableau 7: Les attributs du l’article 38](#_Toc105939787)

[Tableau 8: Les attributs du la famille d'article 38](#_Toc105939788)

[Tableau 9: Les attributs du la ligne de commande 39](#_Toc105939789)

[Tableau 10: Les attributs du commande 39](#_Toc105939790)

[Tableau 11: Les attributs du client 39](#_Toc105939791)

Tableau des figures

[Figure 1: logo TOUT EST LA 2](#_Toc105940142)

[Figure 2: Localisation TOUT ST LA 3](#_Toc105940143)

[Figure 3: Structure de notre application 5](#_Toc105940144)

[Figure 4: cas d'utilisation globale 6](#_Toc105940145)

[Figure 5:Diagramme cas d'utilisation d'agent commercial 7](#_Toc105940146)

[Figure 6: Diagramme cas d'utilisation d'administrateur 8](#_Toc105940147)

[Figure 7: Diagramme de classe 9](#_Toc105940148)

[Figure 8: Diagramme de séquence d'ajout de commande 11](#_Toc105940149)

[Figure 9: Diagramme de séquence d'ajout d'article 12](#_Toc105940150)

[Figure 10: Diagramme d'activité « Passer une commande » Figure 11: Diagramme d'activité « ajouter article» 13](#_Toc105940151)

[Figure 12: Microsoft Visual Studio 15](#_Toc105940152)

[Figure 13: Visual studio code 15](#_Toc105940153)

[Figure 14: SQL Server 2012 15](#_Toc105940154)

[Figure 15: Start UML 16](#_Toc105940155)

[Figure 16: Postman 16](#_Toc105940156)

[Figure 17: Microsoft IIS 16](#_Toc105940157)

[Figure 18: C# 17](#_Toc105940158)

[Figure 19: ASP .NET 17](#_Toc105940159)

[Figure 20: ionic 17](#_Toc105940160)

[Figure 21: Angular 18](#_Toc105940161)

[Figure 22: logo de notre application 18](#_Toc105940162)

[Figure 23: Création de base de données 19](#_Toc105940163)

[Figure 24: Configuration de bas de donnée 19](#_Toc105940164)

[Figure 25: Création de tableau 20](#_Toc105940165)

[Figure 26: insérer les champs de base de données 20](#_Toc105940166)

[Figure 27: Schéma de base de donnés 21](#_Toc105940167)

[Figure 28:Exemple de requête Linq 22](#_Toc105940168)

[Figure 29: Fichier configuration de base de données Config.xml 23](#_Toc105940169)

[Figure 30: test de fonction 23](#_Toc105940170)

[Figure 31: L'emplacement d’IIS 24](#_Toc105940171)

[Figure 32: Activation de service d’IIS 24](#_Toc105940172)

[Figure 33: default web site d’IIS 25](#_Toc105940173)

[Figure 34: publier API 25](#_Toc105940174)

[Figure 35: Node Js 26](#_Toc105940175)

[Figure 36: la fonctionnalité de JWT 27](#_Toc105940176)

[Figure 37: les compositions de JWT 28](#_Toc105940177)

[Figure 38: page d'authentification Figure 39: page modifier mot de passe 29](#_Toc105940178)

[Figure 40: page tableau de bord Figure 41: menu d'application 30](#_Toc105940179)

[Figure 42: page de passage une commande 30](#_Toc105940180)

[Figure 43: formulaire d'ajout un article 31](#_Toc105940181)

[Figure 44: formulaire d'ajout famille article 31](#_Toc105940182)

[Figure 45: formulaire d'ajout client 32](#_Toc105940183)

[Figure 46: formulaire d'ajout un rôle 32](#_Toc105940184)

Remerciement

Au bout de nous travaillons de projet de fin d’étude, nous voulons tout d’abord exprimer sincères remerciements à **DIEU** qui nous donne la patience et le courage pour continuer enfin de réaliser ce travail.

Nous salu les membres du jury pour l'honneur qu'ils nous font en acceptant de juger ce modeste travail.

**A nos encadreurs**

Nous tient à exprimer nos gratitudes et présenter nos chaleureux remerciements à nos encadreurs Mr.Taher Haout et Moez ben otheman qui n'a pas cessé de nos prodiguer ses conseils et n'a épargné aucun effort pour contribuer à la réussite de ce travail et à mes professeurs, sans exception, pour leurs efforts afin de nos assurés une bonne formation à l'Institut Supérieur des Études Technologiques de Djerba.

**A nos chères familles**

Que nul remerciement ne puisse exprimer ce que nous leurs dois, pour leur bienveillance, leur affection et leur soutien... Trésors de bonté, de générosité et de tendresse, en témoignage de nos profonds amours et nos grandes reconnaissances « Que Dieu vous garde ».

Avec beaucoup d’égard, nous ne manquerons pas d’examiner notre grande reconnaissance à tous les enseignants et administrateurs de l'Institut Supérieur des Études Technologiques de Djerba.



Introduction générale

Les évolutions technologiques phénoménales et la crise provoquée par le virus COVID-19 ont fondamentalement changé le monde tel que nous le connaissons.

Le marché des applications mobiles connait un essor phénoménal et devient véritable manne financière pour les entreprises.

C’est dans cette optique que plusieurs entreprises telles que le supermarché et les hypermarchés n’ont pas hésité à exploiter les avancées technologiques pour offrir des services rapides à leur consommateur.

Le seul et unique objectif était la recherche perpétuelle des solutions pratique pour fidéliser le client et facilite leur commande.

L’idée ingénieuse née d’après la difficulté que l’on trouve dans la prise de commande enter le vendeur et le client, le remplissage et l’enregistrement des factures, l’obstination les détails de produit, etc.…, ce pour cela on a réalisé cette application pour défendre tous ces problèmes et facilite le travaille.

La réalisation de ce projet entre dans l’entreprise de super distribution une éventuelle intégration d’un système pareil par rapport à leur système d’information.

Dans ce contexte le rapport comporte trois chapitres organisés comme suite :

Le premier chapitre présente l’organisme d’accueil que la spécification des besoins et l’architecture adoptée.

Le deuxième chapitre se concentre sur le choix de l’architecture globale appliquée à la conception de l’application mobile.

Enfin nous expliquons les différentes étapes de la réalisation qui constitue le dernier volet de ce rapport et de l’objectif de ce stage.

Chapitre 1 : Présentation générale

Introduction

Ce chapitre consacré pour vous présentez le cadre de travaille, la présentation de l’organisme d’accueil et la présentation de nous application.

1. Cadre de travail

Ce stage s’inscrit dans la cadre d’un projet de fin d’étude pour l’obtention d’un diplôme de licence de la technologie d’informatique de l'Institut Supérieur des Études Technologiques de Djerba. Notre stage a effectué aux seins de la société TOUT EST LA.

Le sujet est intitulé « APPLICATION MOBILE DE PRISE DE COMMANDE ».

1. Présentation de l’organisme d’accueil

TOUT EST LA est née d'idées simples, mais de grandes ambitions avec un concept étudier et préparer par MR.MOEZ BEN OTHMEN gérant actuel de la société « TOUT EST LA ».



Figure : logo TOUT EST LA

Le centre commercial TOUT EST LA localisée dans Houmt Souk Djerba



Figure : Localisation TOUT ST LA

1. Service de l’organisme d’accueil

TOUT EST LA est basé sur une gamme étendue d'articles alimentation générale fruits légume poisson et autre comme décoration pour votre maison, mariage, anniversaire, les vases décoratifs, les vases de leurs, cadeaux pour hommes et femmes. TOUT EST LA le leader dans la vente des produits décoratifs et des accessoires de cuisine avec un large choix de produits. La notion de prix et très importante dans ce concept.

1. Histoire de l’organisme d’accueil

TOUT EST LA est une filiale de group BEN OTHMEN qui est un group familial fondé par ROMDANE BEN OTHMEN depuis presque 40 ans. Le fondateur est un homme autodidacte, il a commencé sa carrière par vendre des gilettes devant l’école primaire de son cartier, ensuite plusieurs autre activité comme taxi, chauffeur de bus, marchant ambulant. Par son initiative a commencé son premier projet de vente en gros touts produits. Vers la fin des années 1985, une nouvelle phase de diversification a été entamée, développant plusieurs nouveaux produits et des nouvelles activités telles que les produits naturels. Dans le même cadre en 1995 on commencé la notion du grand surface avec une supérette de 200 m carré. En 2010 le groupe a décidé de s’introduire dans le domaine immobilier. Fin2011, un nouveau bébé a vu le jour : TOUT EST LA Dans le même cadre une société d’importation est installé pour compléter l’activité de grand surface TOUT EST LA. Aussi bien le groupe a décidé d’investir dans le domaine d’industrie par une chaine de production des éponges à récurer sous le nom du « SPONGE LUX ».

1. Présentation de l’application

L’avantage de notre application est facile d’utilisation et regrouper les services possibles entre le client et vendeur.

1. Etude de l’existant

TOUT EST LA est un centre commercial qui achète et vend différents types de produits, c'est pourquoi il a beaucoup de commandes pour déterminer la relation entre le centre commercial et le client qui contient les produits et les quantités dont ils ont besoin.

C'est pourquoi vous souhaitez récupérer le prix du produit et calculer le total de cette commande.

Cette tâche de l'appliquer lors du passage d'une commande prend beaucoup de temps et retarde plus ou moins l'opération de commande, surtout lorsque la commande contient plusieurs produits et un grand nombre de produits.

1. Solution proposée

La solution proposée est de développer une application mobile permettant de gérer les systèmes d'information de l'entreprise via une interface API sécurisée.

L'application permet d'enregistrer les commandes des clients, ce qui aide à générer des commandes avec le prix des produits récupérés en scannant le code a barre de chaque produit, l'utilisateur peut également sélectionner le montant demandé d’une commande ce qui il peut ensuite le partager avec mail.

L’application permet aux l’utilisateur de gérer les articles, les familles d’article et les clients.

Conclusion

Au cours de ce chapitre, nous expliquons le contexte de travail d’une manière générale dans le but de bien définir les objectifs de l’application à réaliser et cela suivra la référence au cours du chapitre suivant l'étude conceptuelle. .

Chapitre 2 : Etude conceptuel

Introduction

Au cours de l’étude de l’existant et l’essai de réussir des demandes de nos enseignes.

Dans ce chapitre nos vous présentez la conception de notre application mobile nous utilisons le langage UML (Unified Modeling Language), mais tout d’abord on va obtenir la maquette de notre application.

1. Maquet de l’application

La présentation de la maquette d’une application mobile est très importante pour savoir à l’utilisateur les étapes pour bien d’utilisation de l’application et bien organiser les informations sur les différents écrans. Cette organisation permet une visualisation claire et facilite leur utilisation.

Figure : Structure de notre application

1. Conception UML

L’absence de modèle se faisait sentir. Le code était, en outre, faiblement documenté et nous n’étions pas à l’abri d’une perte de connaissances due au départ d’un développeur. UML nous a permis de décrire nos besoins et nos spécifications de façon très précise.

Nos codes sont toujours en phase avec le modèle, ce qui fait qu’un nouveau technicien n’a besoin que de connaître UML pour comprendre tout de suite la structure du programme.

* + 1. Diagramme de cas d’utilisation

Les diagrammes de cas d’utilisation sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d’un système logiciel.

Ils sont utiles pour des présentations auprès de la direction ou des acteurs d’un projet, mais pour le développement, les cas d’utilisation sont plus appropriés.

On commence par la présentation générale de cas d’utilisation pour vous donnez une vue globale de notre application

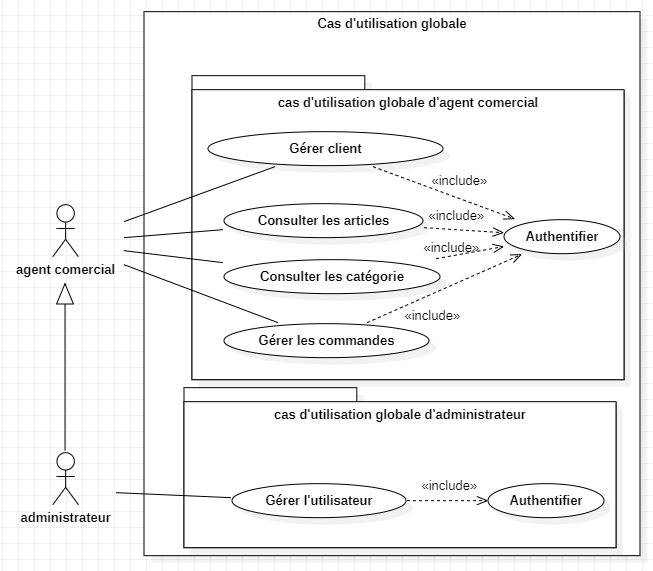


Figure : cas d'utilisation globale

Nous allons passer maintenant le diagramme de cas d’utilisation de chaque acteur avec un tableau descriptif.

* Agent commercial

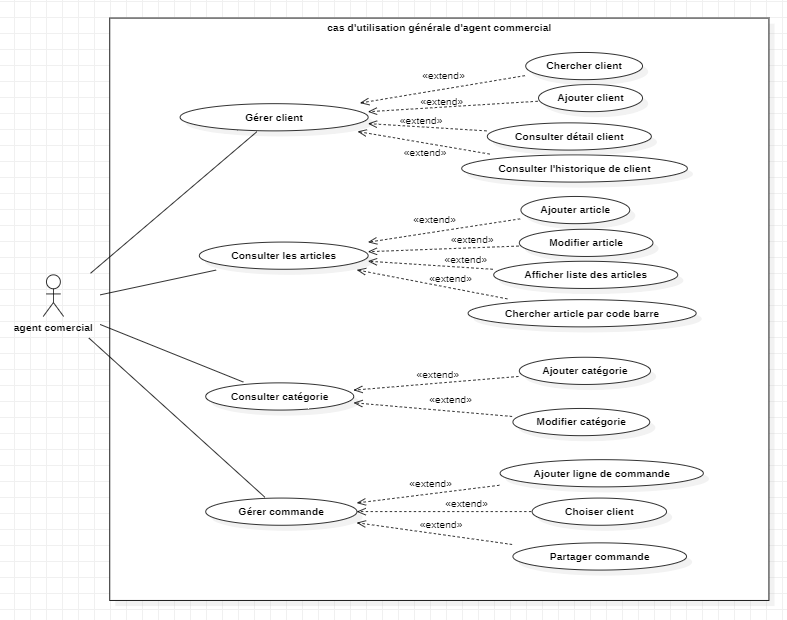


Figure :Diagramme cas d'utilisation d'agent commercial

|  |  |
| --- | --- |
| Description d’un cas d’utilisation | |
| Titre de cas d’utilisation : | Diagramme cas d’utilisation d’agent commercial |
| Objectif : | Les taches d’agent commercial |
| Acteur : | Agent commercial |
| Condition : | Authentifier |
| Description de l’enchainement | |
| Scénario alternative : | 1. L’agent commercial peut gérer les clients que ce soit ajouter, consulter ces détails, consulter ces histoires et chercher. 2. L’agent commercial peut consulter les articles que ce soit ajouter, modifier, afficher la liste et chercher par code a barre. 3. L’agent commercial peut consulter les catégories que ce soit ajouter et modifier. 4. L’agent commercial peut gérer les commande que ce soit ajouter et partager. |

Tableau : Description de cas d'utilisation d'agent commercial

* Administrateur

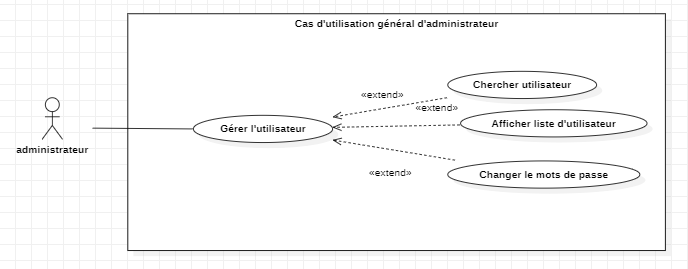


Figure : Diagramme cas d'utilisation d'administrateur

|  |  |
| --- | --- |
| Description d’un cas d’utilisation | |
| Titre de cas d’utilisation : | Diagramme cas d’utilisation d’administrateur |
| Objectif : | Les taches d’administrateur |
| Acteur : | Administrateur |
| Condition : | Authentifier |
| Description de l’enchainement | |
| Scénario alternative : | 1. L’administrateur peut gérer les utilisateurs cherchés, afficher et changer le mot de passe. |

Tableau : Description de cas d'utilisation d'administrateur

* + 1. Diagramme de classe

Le diagramme de classe est un schéma utilisé en génie logiciel pour présenter les classe les classe et les interfaces des systèmes utilisée ainsi que leur relation.

Ce diagramme fait partie de la partie statistique d’UML, ne s’intéressant pas aux aspects temporels et dynamique.

Le diagramme de classe du notre application mobile ce présent comme ces suites:

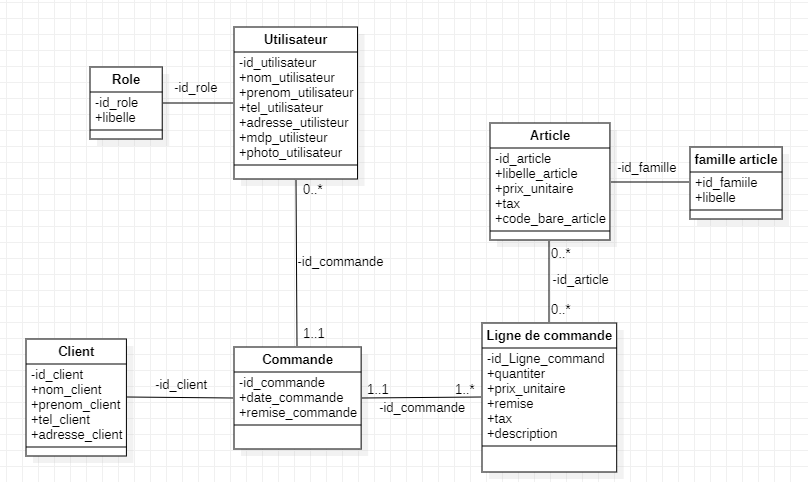


Figure : Diagramme de classe

Le diagramme de classe est centrée autour de la classe commande qui est associée a la classe client et a la classe ligne de commande.

Pour modéliser une commande client 3 classe intéressant qui sont commande, client et ligne de commande. Une commande et relative a un seul client, mais elle peut avoir plusieurs lignes.

Ligne de commande ne peut contienne qu’un seul article.

Les articles sont regroupé par plusieurs catégorie ce qui explique l’association entre la classe article et famille article.

Par le besoin de suivre on doit centrer le nom de l’agent qui a créé la commande, c’est par on a ajout l’association commande – utilisateur.

La classe utilisateur associé à la classe rôle ce qui précise les fonctionnalités d’un utilisateur.

Il s'agit d'établir la description des classes et le dictionnaire de données nécessaires pour le traitement automatique de cette application. Il se présente comme le montre l’annexe.

* + 1. Diagramme de séquence

Le diagramme de séquence est un diagramme d’interaction qui expose en détail la façon dont les opérations sont effectuées : quel message sont envoyés et quand ils le sont. Les diagrammes de séquence sont organisés en fonction du temps. Le temps s’écoule au fur et à mesure que vous parcourez la page. Les objets impliqués dans l’opération sont répertoriés de gauche à droit en fonction du moment où ils prennent part dans la séquence de message.

* **Diagramme de séquence d’ajout une commande**

Ce diagramme de séquence explique l’interaction enter les classes durant le passage d’une commande de façon détailler.

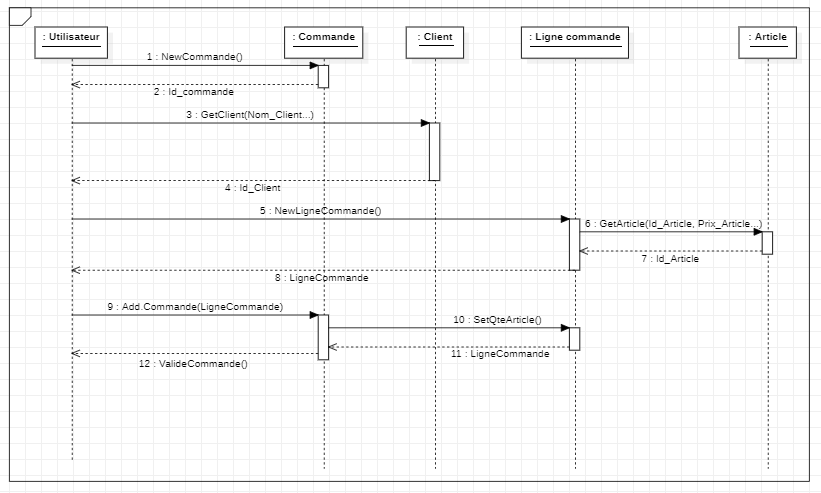


Figure : Diagramme de séquence d'ajout de commande

L'utilisateur interagira avec l'interface pour ajouter des commandes.

La première chose à faire est de sélectionner le client.

L'ajout d'une ligne de commande nécessite l'ajout d'un article, ce qui nous permet de choisir la quantité que nous voulons demander pour le client.

* **Diagramme de séquence d’ajout un article**

Ce diagramme de séquence explique l’interaction enter les classes durant l’ajout d’un article de façon détailler.

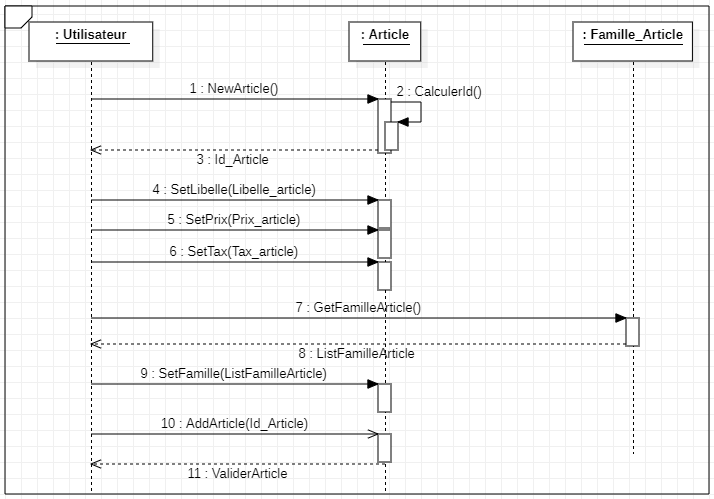


Figure : Diagramme de séquence d'ajout d'article

Ajoutez un article et l'utilisateur interagira avec un formulaire permettant la saisie des propriétés de l'article.

* + 1. Diagramme d’activité

Le diagramme d’activité fournit une vue du comportement d’un système en décrivant la séquence d’action d’u processus. Les diagrammes d’activité sont similaires aux organigrammes de traitement de l’information, car ils montrent les flux entre les actions dans une activité.

On a essai de construire un diagramme d’activité qui explique le déroulement d’un système aux cours d’ajout une commande et un article.

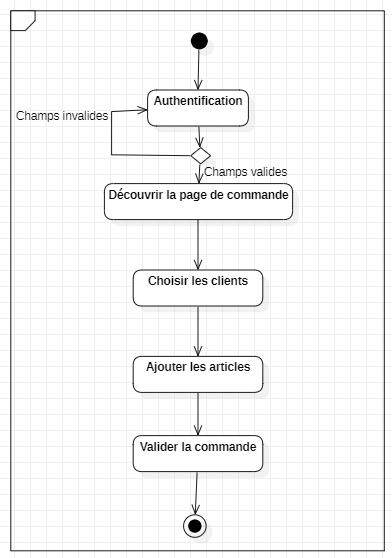
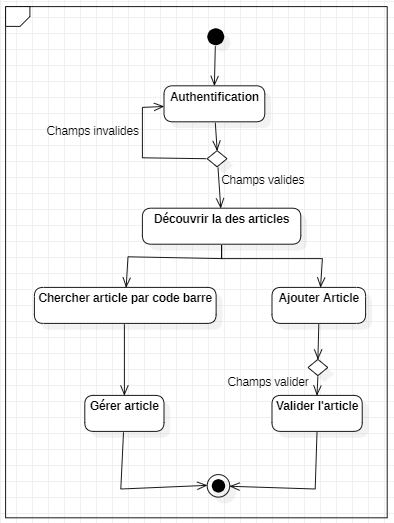
 

Figure : Diagramme d'activité « Passer une commande » Figure : Diagramme d'activité « ajouter article»

Conclusion

Dans ce chapitre, nous avouons détaillée les différant vue conceptuelle d’application à réaliser à travers des modèles UML nécessaire. Cette partie de travail facilite la phase suivante « La réalisation ». .

Chapitre 3: Réalisation

Introduction

Après avoir effectué l’étape de conception, nous avons vous présenter la dernière partie dans ce rapport.

On va vous présenter les environnements de travail logiciel et matériel du projet. Ensuite, nous intéressons à la description d'interface de l’application mobile.

1. Environnements de travaille
2. Environnement matériel

|  |  |  |
| --- | --- | --- |
| Marque | MSI | HP |
| Système d’exploitation | Windows 11 Professionnel 64 bits | Windows 10 famille 64 bits |
| Processeur | Intel® Core™ i5-10300H CPU | AMD E2-7110 APU |
| Ram | 16 Go | 4 Go |
| Disque Dur | 500 ssd | 500 ssd |

Tableau : Environnements matériels

1. Environnement logiciel

Pour réaliser ce projet il existe d’installer plusieurs logiciels en locale qu’ils sont :

* **Visuel studio 2022 :** est une suite de logiciels de développement pour  [Windows](https://www.techno-science.net/definition/7613.html)  conçu par Microsoft. La dernière version s'appelle Visual Studio 2010. C’est un [ensemble](https://www.techno-science.net/glossaire-definition/Ensemble.html) complet d'outils de développement permettant de générer des applications Web ASP.NET, et des applications mobiles.



Figure : Microsoft Visual Studio

* **Visuel studio code :** est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C++, C#, Java, Python, PHP) et des runtimes (tels que .NET et Unity).

****

Figure : Visual studio code

* **SQL Server 2012 :** est un système de gestion de base de données relationnelle  conçu pour l'environnement d'entreprise. Comme ses prédécesseurs, SQL Server 2012 comprend un ensemble d'extensions de programmation pour améliorer le langage de requête structuré ( [SQL](https://searchsqlserver.techtarget.com/definition/SQL) ), un langage interactif et de programmation standard permettant d'obtenir des informations et de mettre à jour une base de [données](https://searchsqlserver.techtarget.com/definition/database).



Figure : SQL Server 2012

* **Start UML :** est un [logiciel](https://fr.wikipedia.org/wiki/Logiciel) de modélisation [UML](https://fr.wikipedia.org/wiki/UML_(informatique)), qui a été « cédé comme [open source](https://fr.wikipedia.org/wiki/Open_source) » par son éditeur, à la fin de son exploitation commerciale (qui visiblement continue ...), il gère la plupart des diagrammes spécifiés dans la norme UML.



Figure : Start UML

* **Postman :** est une plate-forme API pour la création et l'utilisation d'API. Postman simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin que vous puissiez créer de meilleures API plus rapidement.

****

Figure : Postman

* **IIS :** est un serveur web FTP complet doté de puissants outils d’administration, de fonctionnalités de sécurité solides et peut être utilisé pour héberger des applications ASP.NET et PHP sur le même serveur, il trois méthodes d’installer et de configurer un site web dans IIS sur un windows : interface utilisateur graphique, Power Shell ou windows CMD.

****

Figure : Microsoft IIS

1. Technologies utilisé

* **C# :** est un langage phare du framework .net qui se popularise pour la conception de sites Web (ASP).

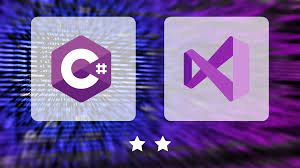


Figure : C#

* **ASP.NET CORE :** est un framework web multiplateforme, hauts performance et open source pour la création d’application moderne, avec ASP.NET Core on peut crée des applications et des services web, des applications internet des objets et des bakcends mobiles.

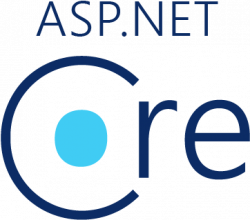


Figure : ASP .NET

* **Ionic :** est une boîte à outils d'interface utilisateur open source permettant de créer des applications mobiles et de bureau performantes et de haute qualité à l'aide de technologies Web - HTML, CSS et JavaScript - avec des intégrations pour des frameworks populaires tels que [Angular](https://ionicframework.com/docs/angular/overview) , [React](https://ionicframework.com/docs/react) et [Vue](https://ionicframework.com/docs/vue/overview) .



Figure : ionic

* **Angular :** est une plateforme de développement, construite sur [Manuscrit](https://www.typescriptlang.org/). En tant que plate-forme, Angular comprend :
* Un cadre basé sur des composants pour créer des applications Web évolutives
* Une collection de bibliothèques bien intégrées qui couvrent une grande variété de fonctionnalités, y compris le routage, la gestion des formulaires, la communication client-serveur, etc.
* Une suite d'outils de développement pour vous aider à développer, construire, tester et mettre à jour votre code.

****

Figure : Angular

1. Tache réalise
2. Création du logo

A l’aide de canvas pros, on a essai de réaliser un logo de notre application mobile.



Figure : logo de notre application

1. Création de base de données

L’implémentation de cette tache la création de base de données on utilise un système de gestion de données puissant et fiable SQL Server 2012, dans le quelle nous commençons de gérée les tables et les lignes comme se suit :

On commence par la création de base de données



Figure : Création de base de données

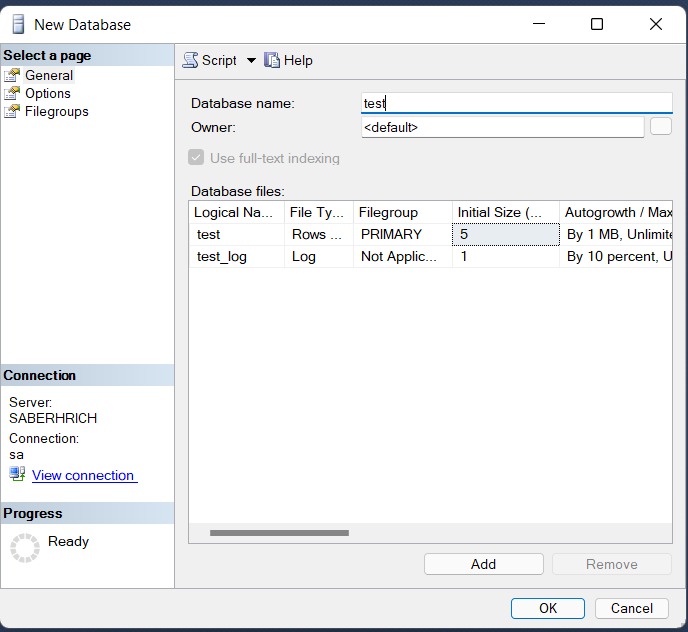


Figure : Configuration de bas de donnée

Après on va créer les table comme se suit

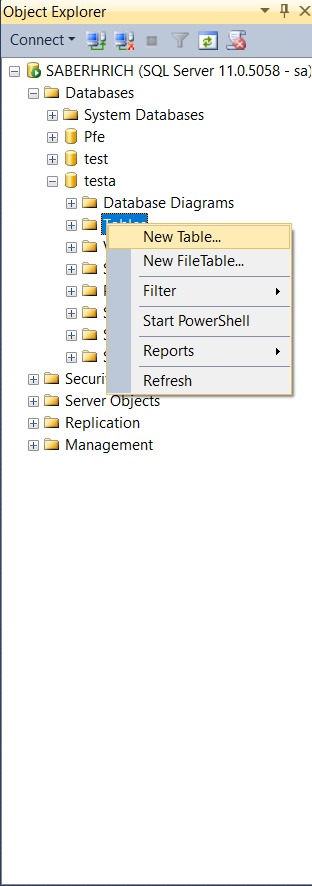


Figure : Création de tableau

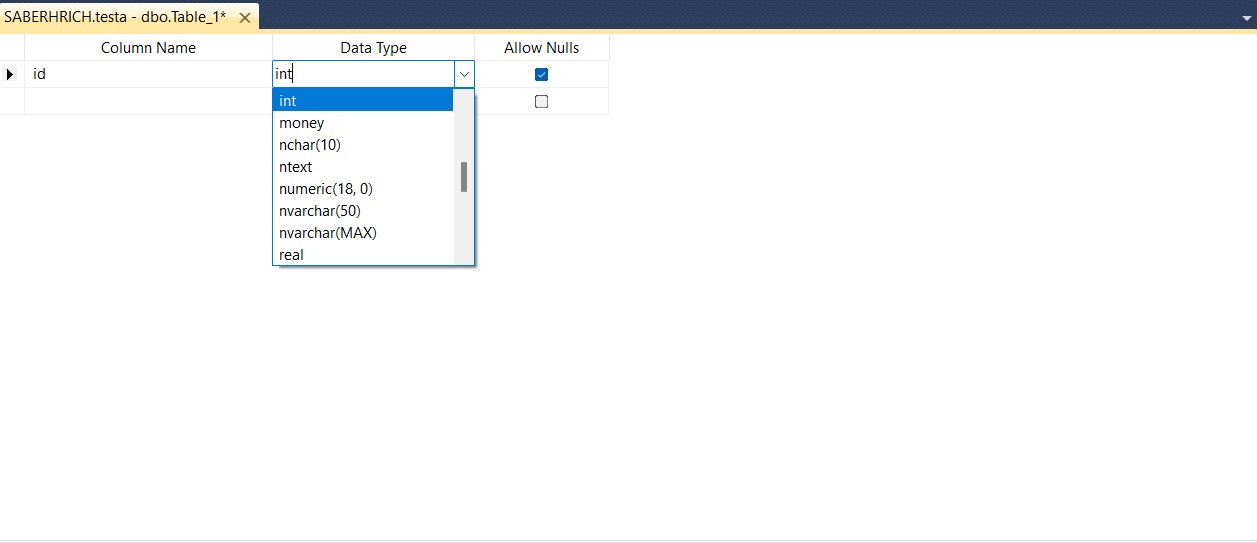


Figure : insérer les champs de base de données

SQL Server Mangement Studio permet de créer une représentation graphique de la base de données. Cette représentation est appelée schéma de base de donnée.

Voisez le schéma de notre base de données :

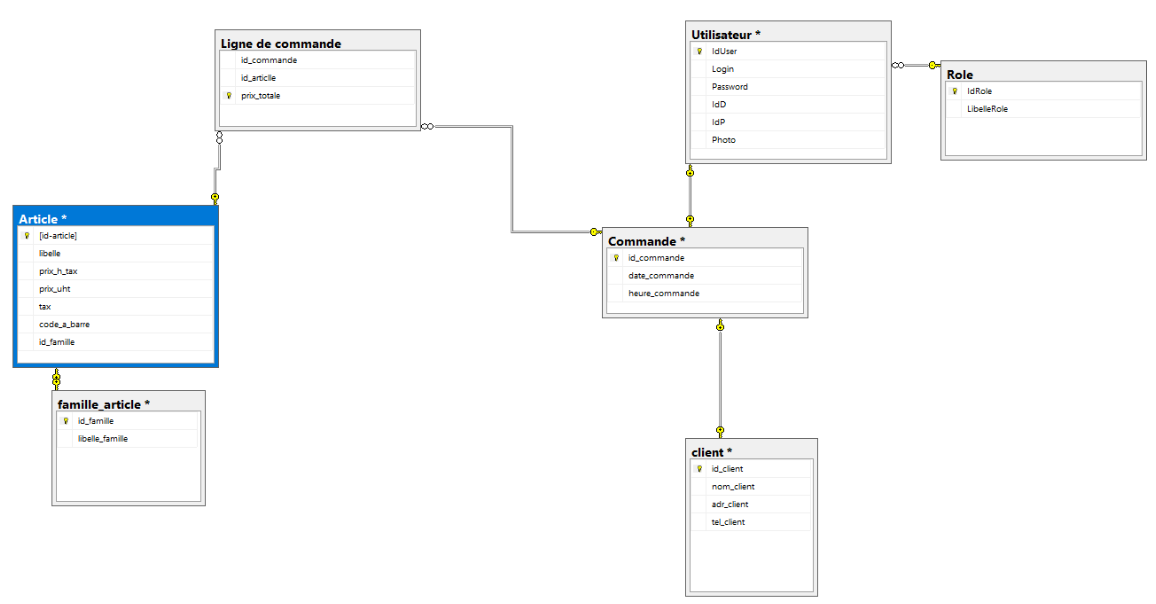


Figure : Schéma de base de donnés

1. Réalisation API

* **Création d’API**

On a commencé par réaliser un projet API par l’utilisation du Visual Studio 2022 :

1. Installer les packages
2. Implémenter les contrôler qui définisse les attributs de chaque tableau
3. Utiliser les requêtes Linq : La requête de langage intégré (LINQ) est un modèle de programmation de framework Microsoft .NET, qui ajoute des capacités de requête aux langages de programmation .NET. Ces extensions fournissent une syntaxe plus courte et expressive pour manipuler les données. Un certain nombre de fonctionnalités ont été ajoutées à C # et Visual Basic pour prendre en charge LINQ. Il enveloppe les requêtes puissantes sur les objets, les fichiers XML et les bases de données.

Il prend en charge un certain nombre de fournisseur LINQ comme ce suit :

* LINQ to XML: convertit les documents XML en un groupe d'objets XElement. Ces objets sont ensuite interrogés par le moteur d'exécution local, qui fait partie de l'opérateur de requête standard.
* LINQ to SQL: permet à LINQ d'être utilisé pour interroger les bases de données du serveur SQL. Étant donné que les données du serveur SQL résident dans une base de données distante et utilisent son propre moteur de requête, ce fournisseur convertit d'abord une requête LINQ en requête SQL et l'envoie au serveur SQL pour traitement.
* LINQ to DataSets: permet la prise en charge de toute base de données générique. Ce fournisseur utilise ADO.net, qui gère la communication avec la base de données.
* LINQ to Objects: utilise le moteur d'exécution de requête local pour les collections en mémoire. Le code généré référence l'implémentation des opérateurs de requête standard.

Voisez un exemple de requête LINQ :

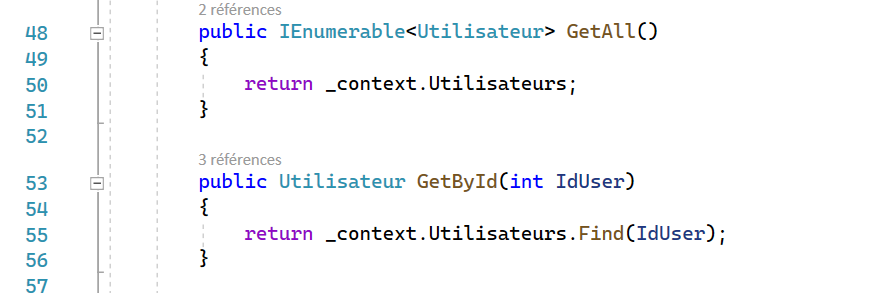


Figure :Exemple de requête Linq

* **Connexion a la base de données**

A l’instant on peut relier notre base de données avec projet .NET comme on le montre

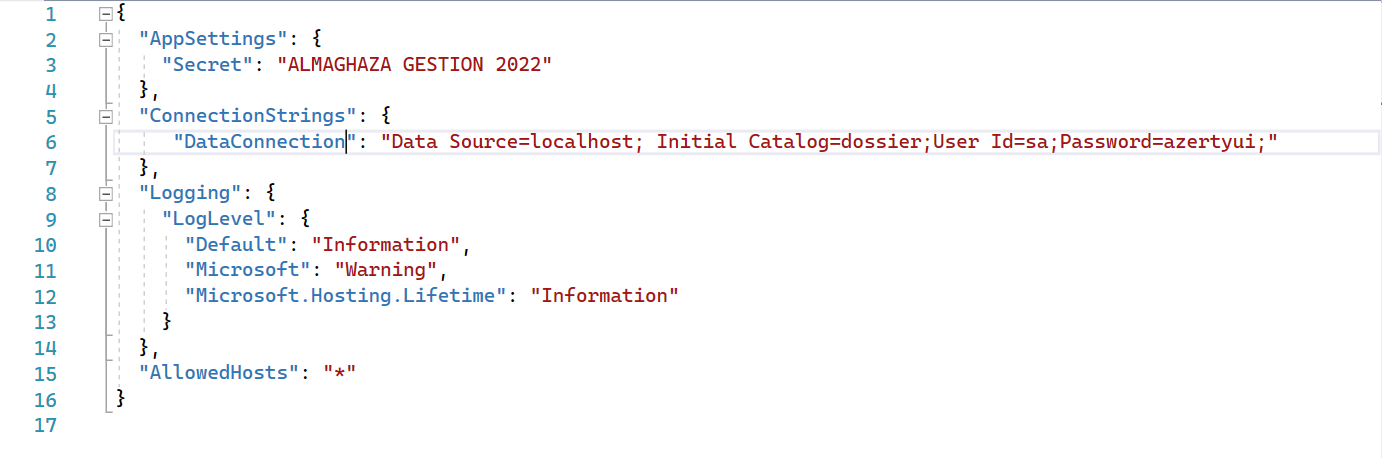


Figure : Fichier configuration de base de données Config.xml

* **Test et validation d’API**

On a testé notre API par le logiciel Postman on utilise l’url d’hébergement par IIS, on a obtenu un résultat de type json comme se suit :

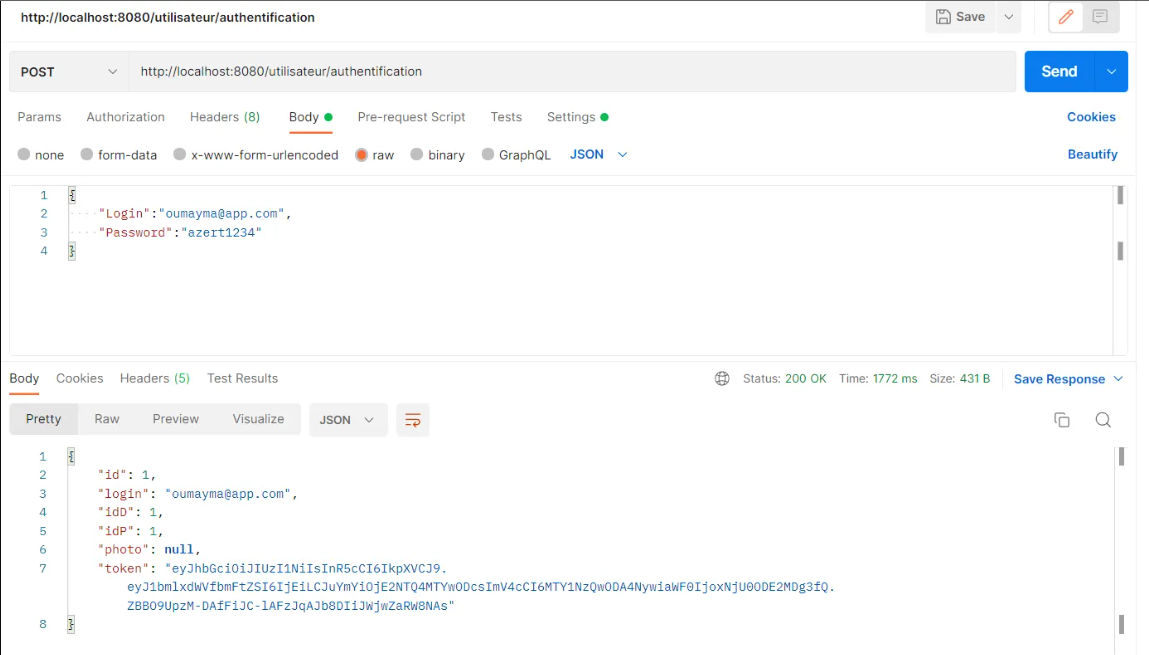


Figure : test de fonction

1. Configuration d’IIS pour héberger un api .net core

Pour installer IIS on a utilisé la méthode d’interface utilisateur graphique voisée les étapes :

1. Commencer à taper « activer ou désactiver les fonctionnalités windows » dans la barre de recherche.

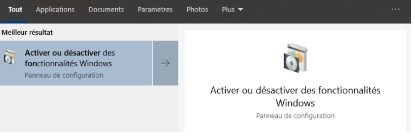


Figure : L'emplacement d’IIS

1. On a obtenu cette boite de dialogue et coucher « Internet Information Services ».

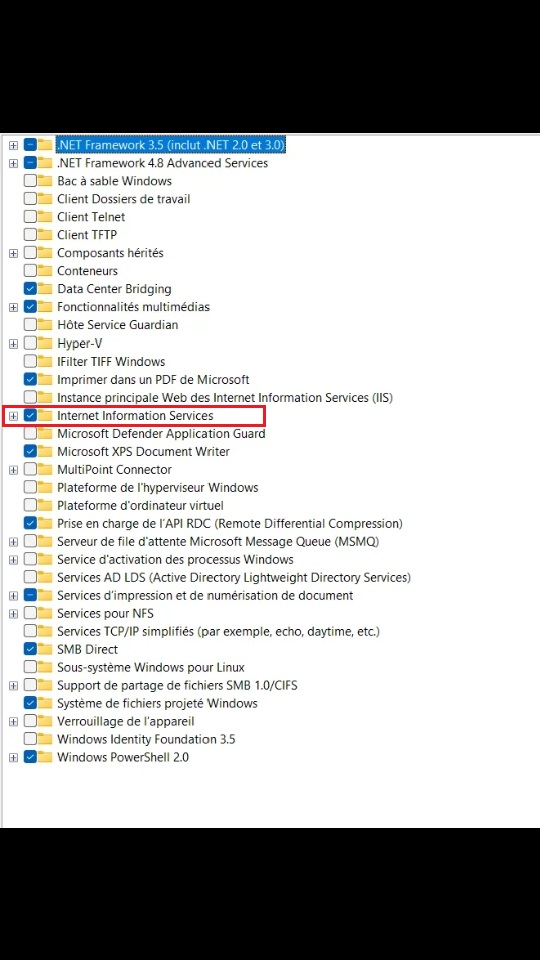


Figure : Activation de service d’IIS

1. L’installation va commencer et peut prendre quelque minute.
2. Il s’agit d’un site d’espace réservé installé avec IIS, on va consulter l’URL qu’on le trouve dans « default web site »

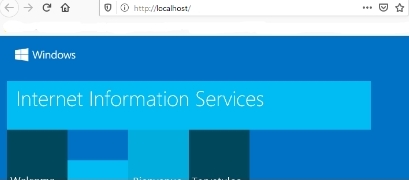


Figure : default web site d’IIS

Maintenant on va héberger notre application ASP.NET Core voisée les étapes :

1. Pour publier l’application ASP .NET Core cliquer sur le bouton droit sur le projet dans l’explorateur de solution et sélectionnez publier

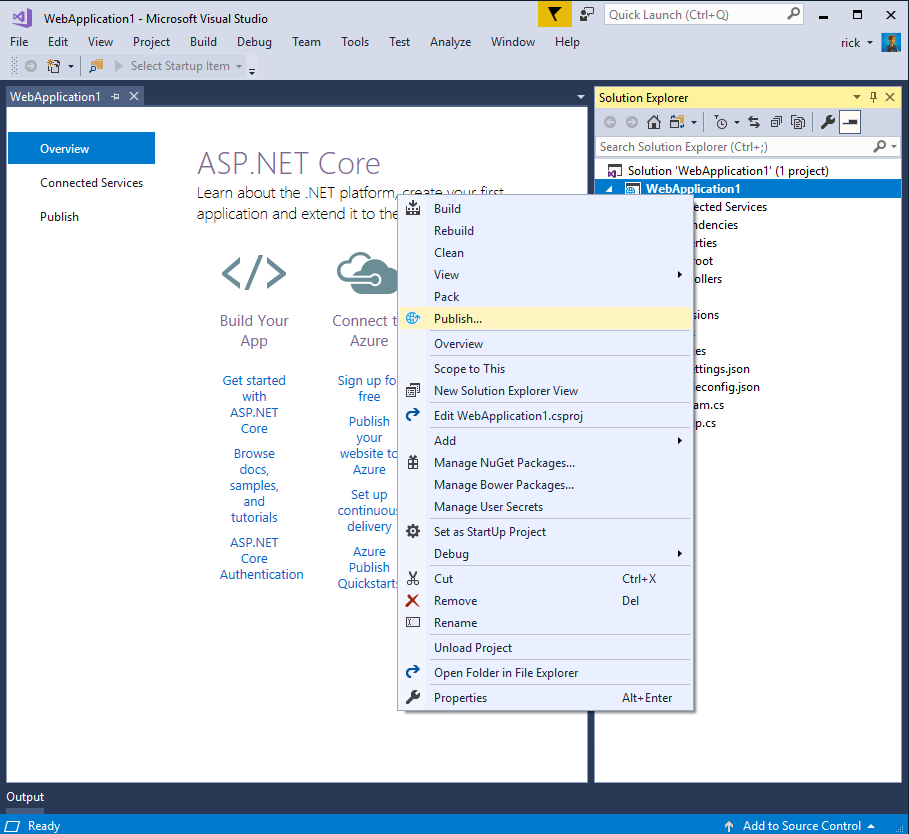


Figure : publier API

1. Dans la boîte de dialogue Choisir une cible de publication, sélectionnez l'option de publication Dossier**.**
2. Créé un dossier pour le site IIS qui est disponible sur la machine de développement en tant que partage réseau, indiquez le chemin d'accès au partage. L'utilisateur actuel doit disposer d'un accès en écriture pour publier sur le partage.

On a installé ces extensions IIS :

* **Cors module (**Cross Origin Resource Sharing)**:** est un mécanisme permettant à un agent utilisateur d'accéder aux ressources d'un domaine en dehors du domaine à partir duquel la première ressource a été servie. CORS définit un moyen en utilisant des en-têtes HTTP supplémentaires pour autoriser les autorisations de demande à accéder à une ressource sélectionnée.
* **.NET Core Hosting bundle :** est un programme d'installation pour le runtime .NET Core et le module ASP.NET Core. Le bundle permet aux applications ASP.NET Core de s'exécuter avec IIS.

Maintenant on peut appeler les fonctions ASP .NET Core d’après l’URL qui l’on crée par l’IIS.

1. Réalisation du l’application ionic

L’outil que nous avons utilisé pour développer l’application front end mobile est Ionic version 6 en utilisant le framework Angular.

Il est important de connaitre que Angular est créé avec le langage Typescript : un langage de programmation qui s’appuie sur la java scripte, l’installation de type scripte nécessite le Node Js



Figure : Node Js

* **Pour quoi ionic**

Le développement d'applications Android seul nécessite d'apprendre à utiliser l'IDE (Integrated Development Environment) d'Android Studio et de connaître le langage de programmation Java ou Kotlin. Vous devrez également comprendre les concepts spécifiques à Android comme les activités et les intentions, et vous créerez les interfaces utilisateur de votre application avec une syntaxe XML personnalisée.

Pour iOS, vous devrez utiliser l'EDI Xcode (qui n'est disponible que pour Mac), ainsi que les langages de programmation Objective-C ou Swift d'Apple. Sans parler des nombreux SDK et concepts spécifiques à Mac.

Pour envoyer notre application sur plusieurs plateformes, nous devons réécrire la même fonctionnalité dans trois langues différentes, Développer des applications de cette manière ne nécessite pas seulement d'apprendre un grand nombre de langages et différents outils, cela signifie également que nous finissons par recoder la même logique pour deux ou trois langages de programmation et plates-formes mobiles différents. Et la même duplication des efforts est nécessaire lorsque nous devons faire des changements. Développer des applications en natif demande beaucoup de temps, d'argent et de compétences.

* **Sécurité**

On a sécurisé l’authentification par JWT (Json Web Token) JWT est un jeton permettant d’échanger des informations de manière sécurisée. Ce jeton est composé de trois parties, dont la dernière, la signature, permet d’en vérifier la légitimité. JWT est souvent utilisé pour offrir une authentification stateless au sein d’applications.

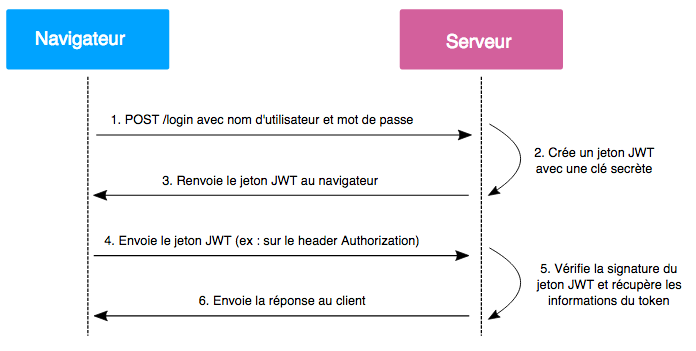


Figure : la fonctionnalité de JWT

Un JWT est composé de trois parties, chacune contenant des informations différentes :

* un header : identifie quel algorithme a été utilisé pour générer la signature, ainsi que le type de token dont il s’agit (souvent JWT, mais le champ a été prévu dans le cas où l’application traite d’autres types d’objet qu’un JWT).
* un payload : est la partie du token qui contient les informations que l’on souhaite transmettre. Ces informations sont appelées “claims”. Il est possible d’ajouter au token les claims que l’on souhaite, mais un certain nombre de claims sont déjà prévus dans les spécifications de JWT.
* la signature : est la dernière partie du token. Elle est créée à partir du header et du payload générés et d’un secret. Une signature invalide implique systématiquement le rejet du token.

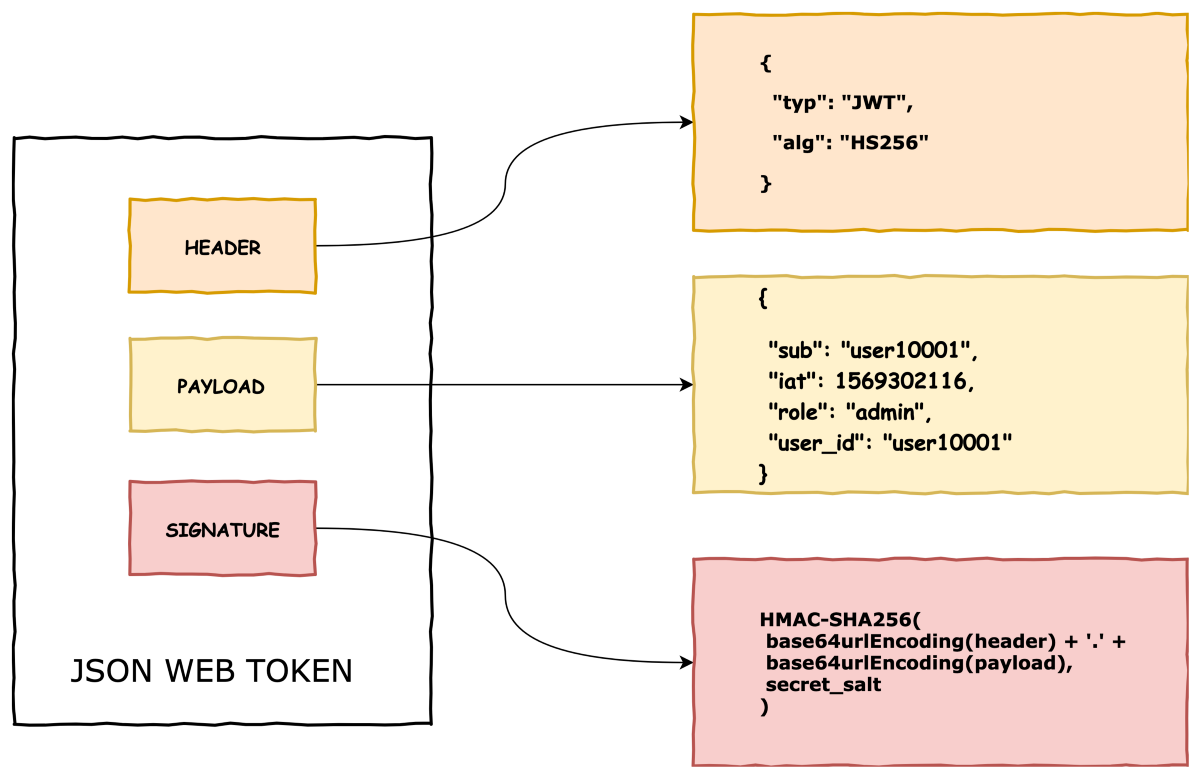


Figure : les compositions de JWT

1. Présentation des interfaces d’application
2. **Interface d’authentification**

La page d'authentification est la première page d'accueil affichée pour l'utilisateur, cette page redirige vers la page d'accueil en fonction du login et du mot de passe, car on a dit à l'authentification d'être protégée par JWT et un lien qui redirige a la page modifier mot de passe.

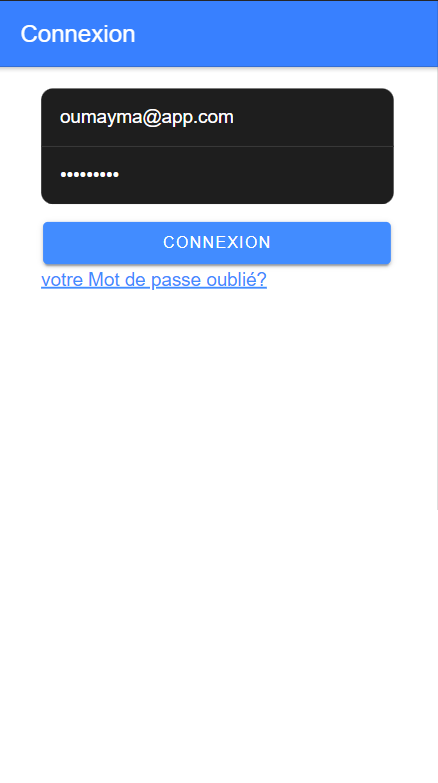
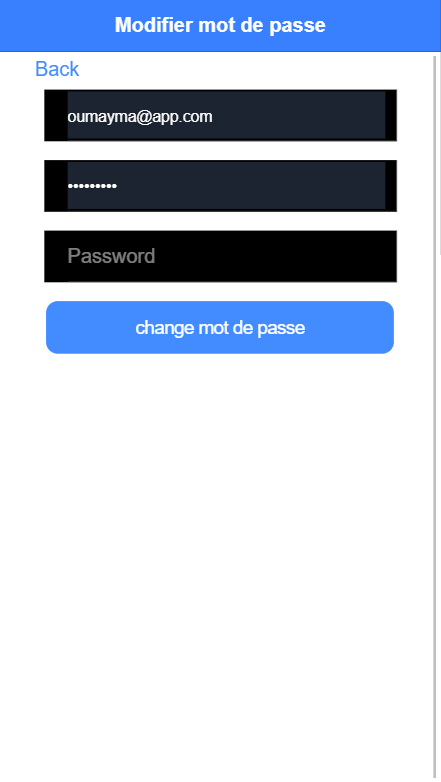
 

Figure : page d'authentification Figure : page modifier mot de passe

1. **Interface tableau de bord**

Cette page contient le menu et le bouton déconnexion

Figure : page tableau de bord Figure : menu d'application

1. **Interface de commande**

Cette page permet aux utilisateurs de passer une commande et de la partager.

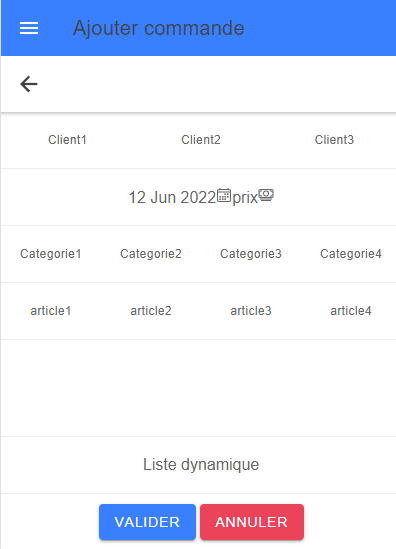


Figure : page de passage une commande

1. **Interface des articles**

Cette page récupère tous les articles de TOUT EST LA et permet également d'ajouter d'autres articles grâce à un bouton qui redirige vers la page d'ajout d'article et un autre bouton qui permet de scanner le code barre de l'article, ce qui nous permet d'accéder à leur propriété.

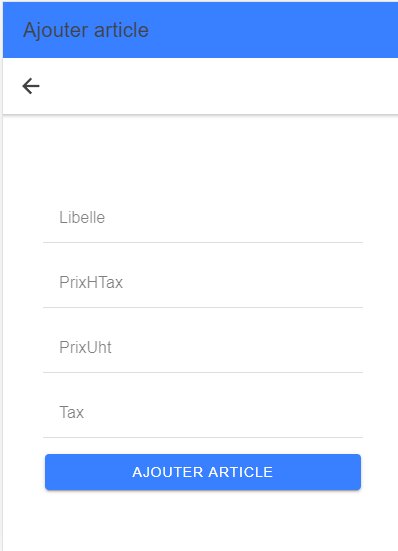


Figure : formulaire d'ajout un article

1. **Formulaire ajout famille article**

Ce formulaire permet à l’utilisateur d’ajouter une famille article



Figure : formulaire d'ajout famille article

1. **Formulaire ajout client**

Ce formulaire permet à l’utilisateur d’ajouter un client

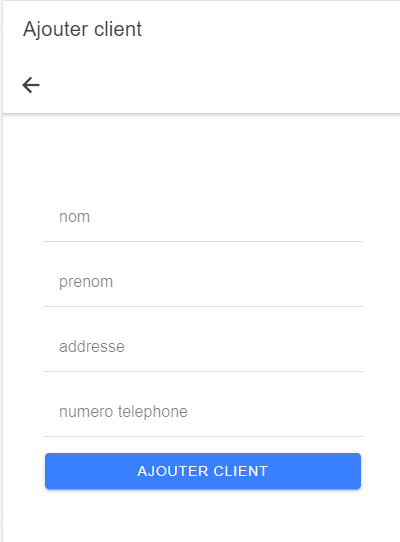


Figure : formulaire d'ajout client

1. **Formulaire ajout rôle**

Ce formulaire permet à l’administrateur d’ajouter un rôle

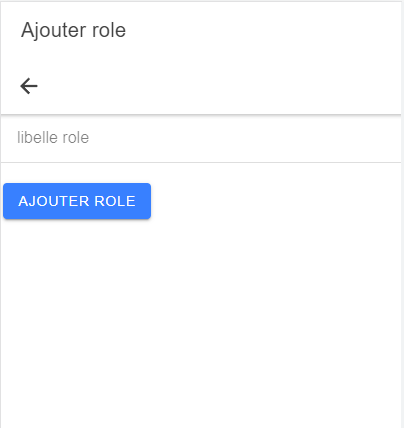


Figure : formulaire d'ajout un rôle

Conclusion

Ce nous a permis de mettre en œuvre notre travaille nous avons commencé par présenter l’environnement matériel et logiciel du projet. Et pour finir nous avons présenté quelque interface de l’application. .

Conclusion générale

L’objectif de ce projet de fin d’étude de concevoir et de développer une application mobile pour la gestion de commande.

Au long du stage nous à été bénéfique sur plusieurs plans, il nous permes de perfectionner nous connaissance acquises en développement et en conception UML. .

Du point de vue technique, ce projet nous à permis de nous adapter avec l’enivrement du développement informatique, de même il nous a permis de maîtriser la maîtriser la méthode de développement, le processus unifiée et de nouvelle technologie de programmation. .

Tout long de l’élaboration du projet, nous avons rencontré plusieurs difficultés tant aux niveaux la réalisation qu’au niveau de conceptuelle. Tout de même, nous avons réussi à les surpasser pour présenter en fin de compte une application opérationnelle. .

Comme tous les projets de fin d’études, notre travail peut supporter plusieurs perspectives. En effet l’application que nous avons développée pourrait être comparateur de cours de change toutes les méthodes de transaction enter TOUT EST LA et leurs clients.

Nous espérons en fin que ce travail que nous avons effectué a été a la hauteur de la confiance que nous a été donnée. .

Bibliographie & Nétographique

1. <https://www.techno-science.net/glossaire-definition/Visual-studio.html>
2. <https://code.visualstudio.com/docs>
3. <https://whatis.techtarget.com/definition/SQL-Server-2012>
4. <https://fr.wikipedia.org/wiki/StarUML>
5. <https://perso.esiee.fr/~perretb/I3FM/POO1/basecsharp/index.html>
6. <https://techlib.fr/definition/aspnet.html>
7. <https://ionicframework.com/docs>
8. <https://angular.io/>
9. <https://www.postman.com/product/what-is-postman/>
10. <https://devdactic.com/ionic-firebase-auth-upload/>
11. <https://fr.theastrologypage.com/language-integrated-query#:~:text=La%20requ%C3%AAte%20de%20langage%20int%C3%A9gr%C3%A9,expressive%20pour%20manipuler%20les%20donn%C3%A9es>.
12. <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0>

Annexe

* **Description des classes**

Le tableau suivant illustre la liste des classes et leur description :

|  |  |
| --- | --- |
| **Nom de classe** | **Description** |
| Utilisateur | Liste des utilisateurs inscrits |
| Rôle | Préciser le rôle d’utilisateur |
| Article | Liste d’article existe |
| Famille article | Préciser les catégories d’article |
| Ligne de commande | Regroupe-les cordonnés des commandes  et les cordonnés des produits |
| Commande | Liste des commande effectue |
| Client | Liste des clients |

Tableau : Description des classes

* **Dictionnaire de données**

Les tableaux suivant illustrent la liste des tables et des attributs de chaque classe, leurs libellés et leurs types:

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Type** | **Description** |
| Id\_utilisateur | Integer auto-incrémenté | L’identifient d’utilisateur |
| Nom\_utilisateur | character varying | Le nom d’utilisateur |
| Prenom\_utilisateur | character varying | Le prénom d’utilisateur |
| Tel\_utilisateur | character varying | Le numéro du téléphone  d’utilisateur |
| Adresse\_utilisateur | character varying | L’adresse d’inscription  d’utilisateur |
| Mdp\_utilisateur | Password | Le mot de passe d’utilisateur |
| Potos\_utilisateur | character varying | La photo d’un utilisateur |

Tableau : Les attributs du l’utilisateur

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Type** | **Description** |
| Id\_rol | Integer auto-incrémenté | L’identifient du role |
| Libelle\_role | character varying | Le role d’utilisateur |

Tableau : Les attributs du rôle

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Type** | **Description** |
| Id\_article | Integer auto-incrémenté | L’identifient d’article |
| Libelle\_article | character varying | Le nom d’article |
| Prix\_unitair | Number | Le prix unitaire d’article |
| Tax | Number | Les impôts obligatoires prélevés |
| Code\_barre | character varying | Le code a barre d’article |

Tableau : Les attributs du l’article

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Type** | **Description** |
| Id\_famille | Integer auto-incrémenté | L’identifient de la famille d’article |
| Libelle\_famille | character varying | Catégorie d’article |

Tableau : Les attributs du la famille d'article

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Type** | **Description** |
| Id\_lignec | Integer auto-incrémenté | L’identifient de ligne de command |
| Quantiter | Number | La quantité d’article commandé |
| Prix\_unitaire | Number | Le prix unitaire de chaque article |
| remise | Number | La remise effectuée de chaque article |
| Tax | Number | Les impôts obligatoires prélevés |
| description | character varying | La description de chaque article |

Tableau : Les attributs du la ligne de commande

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Type** | **Description** |
| Id\_command | Integer auto-incrémenté | L’identifient de command |
| Date\_command | Date | La date de command |
| Remise\_command | Number | La remise effectuée de command |

Tableau : Les attributs du commande

|  |  |  |
| --- | --- | --- |
| **Nom du champ** | **Type** | **Description** |
| Id\_client | Integer auto-incrémenté | L’identifient de client |
| Nom\_client | character varying | Le nom de client |
| Prenom\_client | character varying | Le prénom de client |
| Tel\_client | Number | Le numéro du téléphone de client |
| Adresse\_client | character varying | L’adresse personnelle de client |

Tableau : Les attributs du client