|  |  |  |
| --- | --- | --- |
| **REPUBLIQUE DU CAMEROUN**  *Paix -Travail-Patrie*  **MINISTERE DE L’ENSEIGNEMENT SUPERIEUR** |  | **REPUBLIC OF CAMEROON**  *Peace- Work –Fatherland*  **MINISTRY OF HIGHER EDUCATION** |

Conception et réalisation d’une bibliothèque numérique au sein de l’IUC

Rapport de projet tutoré

Tuteur

**Hypolite TEKEU**

**Niveau :** BAC + 4

Rédigé et Présenté par :

**Patricia YOUFANG**

Matricule n° :

Et

**Jessica TSOPGNI**

Matricule n° : IUC16E0013896

**IUC - 3IAC - CS2I 4 2019-2020**

Sommaire

[Liste des figures et tableaux iii](#_Toc5349340)

[Introduction 1](#_Toc5349341)

[I. Etude de l’existant 2](#_Toc5349342)

[1. Description de l’existant 2](#_Toc5349343)

[2. Critique de l’existant 4](#_Toc5349344)

[3. La concurrence 4](#_Toc5349345)

[4. Solution proposée 5](#_Toc5349346)

[II. Cahier de charges de la solution à mettre en place. 6](#_Toc5349347)

[1. Objectifs 6](#_Toc5349348)

[2. Les cibles 6](#_Toc5349349)

[3. Les responsabilités 6](#_Toc5349350)

[a. Maîtrise d’ouvrage 6](#_Toc5349351)

[b. Maîtrise d’œuvre 6](#_Toc5349352)

[4. Description de la solution 7](#_Toc5349353)

[a. Les besoins fonctionnels 7](#_Toc5349354)

[b. Les contraintes générales 8](#_Toc5349355)

[5. Identité graphique 9](#_Toc5349356)

[a. Logo 9](#_Toc5349357)

[b. Slogan 9](#_Toc5349358)

[c. Polices & Couleurs 9](#_Toc5349359)

[6. Les livrables attendus 9](#_Toc5349360)

[a. Documents 9](#_Toc5349361)

[b. Les fichiers 9](#_Toc5349362)

[c. Autres 9](#_Toc5349363)

[III. Analyse et conception du système 9](#_Toc5349364)

[1. Analyse et Conception 10](#_Toc5349365)

[a. Identification des acteurs 10](#_Toc5349366)

[b. Vue statique du système 10](#_Toc5349367)

[c. Vue dynamique du système 13](#_Toc5349368)

[IV. Gestion du projet 17](#_Toc5349369)

[1. Organisation du travail 17](#_Toc5349370)

[2. Communication interne 17](#_Toc5349371)

[3. Business model 17](#_Toc5349372)

[V. Réalisation du projet 19](#_Toc5349373)

[1. Développement de la base de données 19](#_Toc5349374)

[a. Présentation et téléchargements des outils 19](#_Toc5349375)

[b. Création de la base de données, des tables 21](#_Toc5349376)

[2. Implémentation des services 23](#_Toc5349377)

[a. Architecture du logiciel 23](#_Toc5349378)

[b. Présentation et installation des outils 24](#_Toc5349379)

[3. Interfaces graphiques 31](#_Toc5349380)

[a. La page d’accueil 31](#_Toc5349381)

[b. La page des restaurants 31](#_Toc5349382)

[Conclusion 33](#_Toc5349383)

[VI. Références 34](#_Toc5349384)

[1. Bibliographie 34](#_Toc5349385)

[2. Webographie 34](#_Toc5349386)

[VII. Annexes 35](#_Toc5349387)

# Liste des figures et tableaux

[Figure 1 : Diagramme de flux d’informations 3](file:///D:\JESSICA\CS2I3\Semestre_2\Projet_tutoré\Documentations\pnt.docx#_Toc5349295)

[Figure 2 : Diagramme de package 10](#_Toc5349296)

[Figure 3 : Diagramme de cas d'utilisation gestion des commandes 11](#_Toc5349297)

[Figure 4 : Diagramme de cas d'utilisation gestion des livraisons 11](#_Toc5349298)

[Figure 5 : Diagramme de classe 12](#_Toc5349299)

[Figure 6 : Diagramme de déploiement 13](#_Toc5349300)

[Figure 7 : Diagramme d'activité passer une commande 14](#_Toc5349301)

[Figure 8 : Diagramme d'activité effectuer une livraison 15](#_Toc5349302)

[Figure 9 : Diagramme d'états-transitions de l'objet commande 16](#_Toc5349303)

[Figure 10 : Diagramme de séquence créer un compte 16](#_Toc5349304)

[Figure 11 : Diagramme de Gantt 17](#_Toc5349305)

[Figure 12 : Choix de SQL Server 19](#_Toc5349306)

[Figure 13 : Inscription pour téléchargement 20](#_Toc5349307)

[Figure 14 : Téléchargement 20](#_Toc5349308)

[Figure 15 : Création de la base de données graphiquement 21](#_Toc5349309)

[Figure 16 : Création de la base de données par la syntaxe SQL 21](#_Toc5349310)

[Figure 17 : Création d'une table graphiquement 22](#_Toc5349311)

[Figure 18 : Création d'une table par syntaxe SQL 22](#_Toc5349312)

[Figure 19 : Architecture Orientée Service par Darryl Nelson 23](#_Toc5349313)

[Figure 20 : Architecture MVC 24](#_Toc5349314)

[Figure 21 : Choix de Visual Studio 25](#_Toc5349315)

[Figure 22 : Interface de demarrage 25](#_Toc5349316)

[Figure 23 : Creation d'un projet ASP.NET 26](#_Toc5349317)

[Figure 24 : Creation d'un projet ASP.NET (Suite) 26](#_Toc5349318)

[Figure 25 : Mapping avec la base de données 27](#_Toc5349319)

[Figure 26 : Mapping avec la base de données (Suite) 27](#_Toc5349320)

[Figure 27 : Mapping avec la base de données (Suite) 28](#_Toc5349321)

[Figure 28 : Création d’un controller 28](#_Toc5349322)

[Figure 29 : Création d'un controller (Suite) 29](#_Toc5349323)

[Figure 30 : Création d'une instance de PeeknTchopEntities 29](#_Toc5349324)

[Figure 31 : GetRestaurant by id 29](#_Toc5349325)

[Figure 32 : DeleteRestaurant by id 30](#_Toc5349326)

[Figure 33 : Interface de démarrage de Postman 30](#_Toc5349327)

[Figure 34 : Page d'accueil de notre application 31](#_Toc5349328)

[Figure 35 : Page des restaurants de notre application 31](#_Toc5349329)

[Figure 36 : Ecran principal du logiciel HotelPro 35](#_Toc5349330)

[Tableau 1: Récapitulatif de la critique de l'existant 4](#_Toc5348417)

[Tableau 2 : Récapitulatif des critères de qualité 8](#_Toc5348418)

[Tableau 3 : Plan de communication 18](#_Toc5348419)

[Tableau 4 : Rentabilisation du projet 18](#_Toc5348420)

# Résumé

Inscrites à l’IUC en Management des systèmes d’information pour le compte de l’année académique 2019-2020, l’une des étapes à franchir pour l’obtention du diplôme de fin d’année est la présentation d’un projet qualifié de tuteuré. Le projet tuteuré a deux principaux buts : la mise en exergue et en pratique des connaissances acquises en classe (durant les cours) ainsi que la valorisation du travail d’équipe. C’est dans ce sillage que nous avons élaboré ce travail intitulé conception et réalisation d’une bibliothèque numérique au sein de l’Institut Universitaire de la côte dont l’objectif comme son nom le dit si bien est de donner à l’IUC sa propre bibliothèque numérique pour une formation toujours intégrale et effective de ses apprenants. Pour ce faire nous avons grâce à une étude menée sur le terrain, un cahier de charges et après analyse des faisabilités, opté non pas pour l’adoption d’une solution mais plutôt pour la conception d’une solution répondant de façon spécifique à nos besoins.

# Abstract

# Introduction Générale

Au Cameroun, le secteur des sciences technologiques est sans doute l’un des domaines qui n’évolue peut-être pas à la vitesse de la lumière mais à un rythme tellement rapide au point où l’illettré d’aujourd’hui est cette personne qui ne sait pas ce que c’est que les TICs (Technologie de l’Information et de la Communication), au point où quasiment chaque jour naît une invention/innovation technologique. Les sciences technologiques influencent tous les secteurs d’activités de nos jours et le domaine éducatif n’en pas de reste. C’est pourquoi parmi les multiples applications de gestion scolaire il ne saurait manquer des logiciels de dématérialisation.

La dématérialisation a commencé par des initiatives individuelles et localisées visant à réduire les contraintes matérielles des traitements traditionnels de l'information, notamment en matière de délais de [communication](https://fr.wikipedia.org/wiki/Communication) ou de disponibilité de l'[information](https://fr.wikipedia.org/wiki/Information). En illustration, dès [1967](https://fr.wikipedia.org/wiki/1967), l'envoi d'une [disquette](https://fr.wikipedia.org/wiki/Disquette)  de surcroît infiniment duplicable pouvait être plus rapide et moins coûteux que celui des cent pages d'équivalent [papier](https://fr.wikipedia.org/wiki/Papier). Le bénéfice obtenu a ensuite été recherché par un plus grand nombre, toujours mieux pourvu en matériel requis, puis de manière de plus en plus systématique et professionnelle, au point d'en faire une caractéristique de l'[informatisation](https://fr.wikipedia.org/wiki/Informatisation) des structures ou des cultures. Aujourd’hui le besoin se fait encore plus surtout pressant en ce temps de crise sanitaire dû au Covid-19 où nous prônons le confinement, les distanciations sociales, l’usage de plus en plus restreint des outils (objets, accessoires) commun au grand public. Il est ainsi question pour nous de traiter de la dématérialisation ou de la numérisation (pour certains). Nous pouvons donc nous demander à juste titre : **Comment pourrait-on rendre les documents numériques de notre bibliothèque accessible à tous ?**

Tout au long de notre présentation nous suivrons le cheminement suivant pour essayer de répondre à toutes ses difficultés :

* + Nous débuterons par une étude de l’existant,
  + Ensuite, nous présentation le cahier de charges et ainsi que l’analyse et la conception du système,
  + Et enfin, les spécifications techniques et la gestion du projet.

# Etude de l’existant

L’élaboration d’une bibliothèque numérique au sein de l’IUC naît d’abord d’un constat, une cause et d’un besoin spécifique auquel nous nous devons de répondre. Mais avant d’y arriver, il est primordial de comprendre de fond en comble ce sujet en lui-même. D’où naît le besoin d’implémenter une bibliothèque numérique ? Pourquoi une bibliothèque numérique à l’IUC ?

## Description de l’existant

L’IUC a à son sein une bibliothèque ouverte à des heures de cours qui contient des livres physiques et des livres numériques. Après une enquête sur le terrain nous constatons plusieurs faits :

**Cas 1 :**

Pour ce qui est des livres physiques, ils sont rangés sur des étagères. Ils sont prêtés aux étudiants en fonction de leur besoin. Sauf que cette fois ce processus est accompagné par un logiciel PMB qui nous renseigne avec exactitude sur la position du logiciel dans l’étagère. Donc les étudiants doivent se présenter à la bibliothèque aux heures recommandées c’est-à-dire entre 8h et 17h, demander à avoir un livre spécifique qui sera facilement repérable grâce au logiciel PMB et pourront rester sur place pour l’étudier.

**Cas 2 :**

Quant aux livres numériques, cette bibliothèque en possède plus de 500, tous contenus dans leur base de données.

## Critique de l’existant

**Cas 1 :**

Malheureusement la bibliothèque de l’IUC n’est pas assez grande pour contenir les milliers d’étudiants et les centaines de professeurs présents au campus.

**Cas 2 :**

Bien qu’IUC soit riche en livres numériques, ils restent tout de même inaccessibles car non publiés par manque de plateforme numérique dédiée.

## La concurrence

Une fois notre critique faite, nous avons enquêté pour trouver des solutions existantes. Et voilà ce que nous avons trouvé :

* **FBReader (Favorite Book Reader) :** C’est une application gratuite multilingue (34 langues) et multi plateforme (Android, Mac OS X (en cours), Linux, Windows, Blacberry 10 et bien d’autres).  Ici aussi, vous pouvez paramétrer vos préférences d’affichage, lire vos ebooks et documents aux formats les plus connus (ePub, Mobi, PDF, html, rtf, et bien d’autres).
  + **Avantages :**
    - Sa légèreté,
    - Sa rapidité à s’ouvrir
  + **Inconvénients** :
    - Possibilité de téléchargements.

## Solution proposée

Pour doter l’IUC d’une bibliothèque numérique, plusieurs possibilités de solutions s’offrent à nous :

* **Logiciel de bibliothèque numérique gratuit :** un logiciel gratuit ou freeware est un logiciel propriétaire distribué [gratuitement](https://fr.wikipedia.org/wiki/Gratuit%C3%A9_(%C3%A9conomie)) sans toutefois conférer à l'utilisateur certaines libertés d'usage.

Dans le cadre de notre projet nous n’avons pas trouvé de gratuiciel respectant les exigences de notre cahier des charges.

* **Logiciel de bibliothèque numérique libre et Open Source :** ce sont des applications accessibles gratuitement et offrant la possibilité de modification du code source afin de l’adapter à des besoins plus spécifiques. Sûrement à première vue nous opterons pour implémenter notre bibliothèque à partir de ce noyau sauf que cette option est des moins avantageuses :
* L’**ergonomie** est trop souvent oubliée au profit de la performance d’où une prise en main difficile pour les moins qualifiés ;
* La **configuration** n’est pas **donnée**, en effet, adapter un logiciel libre à ses propres besoins est bien souvent trop compliqué, trop d’options de configuration et souvent mal compris du grand public ;
* Les **drivers** généralement disponibles plus tardivement ;
* Le **déficit de compétences** auprès des informaticiens de gestion charges de la maintenance ;
* La **disponibilité du code source** rend celui-ci également accessible aux concepteurs de virus ;
* Les **garanties réduites** ou **inexistantes**.

Le logiciel libre et Open source bien qu’ayant des avantages indéniables ne peuvent se soustraire à quelques inconvénients dût à son caractères « **libre** » même. C’est pour cette raison non négligeable que ne n’avons pas voulu concevoir notre bibliothèque à partir d’un modèle libre et Open Source

* **Logiciel de bibliothèque numérique payant** : il présente peu d’inconvénients mais qui sont tout de même considérables :
* Le coût d’achat des licences est important ;
* Les termes de licences sont contraignants ;
* Le risque d’abandon d’un produit ou la fin du support de certaines versions ;
* Les logiciels propriétaires sont la cible privilégiée des virus ;
* La dépenses de l’entreprise vis-à-vis du propriétaire or nous sommes sans ignorer que les relations de dépenses ne sont pas la plupart du temps productives et encore faudrait qu’il réponde à nos besoins spécifiques.

En guise d’illustration nous avons **Winbiblio**.

* **Logiciel de bibliothèque numérique conçu sur mesure** : de plus en plus d’entreprises optent aujourd’hui pour le développement de leur propre application. En faisant créer des logiciels sur mesure, elles sont sûres d’avoir des outils qui s’adapteront parfaitement aux spécificités de leur métier et à leur problématique. Le logiciel métier spécifique est donc développé selon un cahier de charges précis afin de répondre aux besoins précis de l’entreprise. Le développement de ce type de logiciel a de réels avantages sur le long terme pour les entreprises.

C’est dans ce sillage que nous avons décidé d’implémenter nous même une bibliothèque numérique pour le compte de IUC. Les multiples avantages suivent :

* Cette bibliothèque numérique sera un outil de travail personnalisé car conçu selon la demande de IUC donc ne proposant que les fonctionnalités attendues ;
* Elle permettra une meilleure flexibilité puisque nous pourrons l’ajuster à la guise de l’établissement ;
* La maintenance sera plus aisée et objective car elle sera réalisée par les concepteurs de l’application et non par des concepteurs essayant de s’approprier une application ;
* Cette bibliothèque reviendra moins chère qu’un logiciel payant vu que nous n’aurons pas payé de cout de licence ou d’hébergeur vu que IUC en possède déjà.

# Cahier de charges de la solution à mettre en place.

## Objectifs

* Satisfaire nos étudiants ;
* Permettre aux étudiants d’acquérir une plus grande connaissance ;

## Les cibles

* Cible principale : les **étudiants**
* Cible secondaire : les **professeurs**

## Les responsabilités

### Maîtrise d’ouvrage

L’entité porteuse du besoin est IUC, département CS2I spécialité Système d’Information basé à Douala.

### Maîtrise d’œuvre

Ce rôle a été attribué à **Jessica TSOPGNI** et **Patricia YOUFANG,** toutesétudiantes en 4ième année CS2I à l’Institut Universitaire de la Côte (IUC).

## Description de la solution

### Les besoins fonctionnels

Pour le bibliothécaire :

* Créer un compte ;
* S’authentifier ;
* Créer des catégories et sous catégories de documents ;
* Ajouter/Supprimer des documents numériques en ligne ;
* Effectuer une recherche ;
* Consulter toutes les catégories et sous-catégories de documents ;
* Ouvrir et lire un document sans possibilité de le télécharger ;

Pour les utilisateurs :

* Créer un compte ;
* S’authentifier ;
* Effectuer une recherche ;
* Consulter toutes les catégories et sous-catégories de documents ;
* Ouvrir et lire un document sans possibilité de le télécharger ;
* Faire une requête dans le but d’une suggestion ou une recommandation ;

### Les contraintes générales

* Qualité de la solution

**BIBLIO** sera évalué principalement sur les critères et les taux d’acceptabilité suivants :

Tableau 1 : Récapitulatif des critères de qualité

|  |  |
| --- | --- |
| CRITERES DE QUALITE | TAUX D’ACCEPTABILITE |
| Capacité fonctionnelle : précis, conforme aux normes et sécurisé | 90% |
| Facilité d’utilisation : compréhensible, intuitif et robuste | 90% |
| Fiabilité : résultats exacts, tolérance aux pannes matériels et logiciels | 99% |
| Performance : rapidité, l’extensibilité (capacité à maintenir la performance même en cas d’utilisation intensive) | 80% |
| Maintenabilité : extensible (peu d’effort pour ajouter des nouvelles fonctionnalités) | 80% |
| Interopérabilité : la communication entre les appareils exécutant BIBLIO doit être fluide et sans perte de données | 100% |
| TOTAL | **88,17%** |

* Effectifs en ressources humaines

02 personnes (la maîtrise d’œuvre) seront affectées à cette réalisation, en qualité d’analystes et de développeurs d’application web.

## Les livrables attendus

### Documents

* Le dossier d’analyses et techniques ;
* Le guide utilisateur ;
* Le manuel de déploiement.

### Les fichiers

* Le fichier source du programme Web ;
* La sauvegarde de la base de données.

### Autres

* Le rapport de présentation du projet ;
* Le Power Point de présentation.

# Analyse et conception du système

Il existe plusieurs méthodologies d’analyse telles que XP et 2TUP. Le principe de la méthode XP est de faire collaborer étroitement tous les acteurs du projet et d’opter pour des itérations de développement très courtes.

Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu’échangent les acteurs et le système, à modéliser le contexte. C’est ainsi que nous avons décidé d’opter pour cette méthodologie.

## Analyse et Conception

### Identification des acteurs

Dans notre système, nous dénombrons 2 acteurs à savoir :

* Le bibliothécaire : C’est celui qui gère l’application ;
* L’étudiant : C’est celui qui utilise l’application

Pour plus de spécifications, nous avons présenté ci-dessous quelques diagrammes :

### Vue statique du système

#### Diagramme de packages

C’est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de votre application, et vous aide à identifier les liens de généralisation et de dépendance entre les packages.

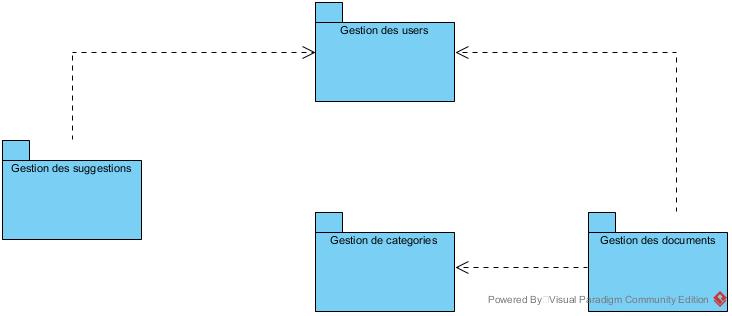


Figure 1 : Diagramme de package

#### Diagramme de cas d’utilisation

C’est un diagramme qui permet de décrire l'interaction entre l'acteur et le système.

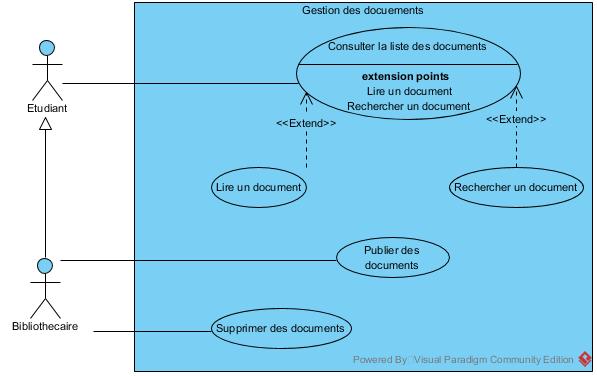


Figure 2 : Diagramme de cas d'utilisation gestion des documents

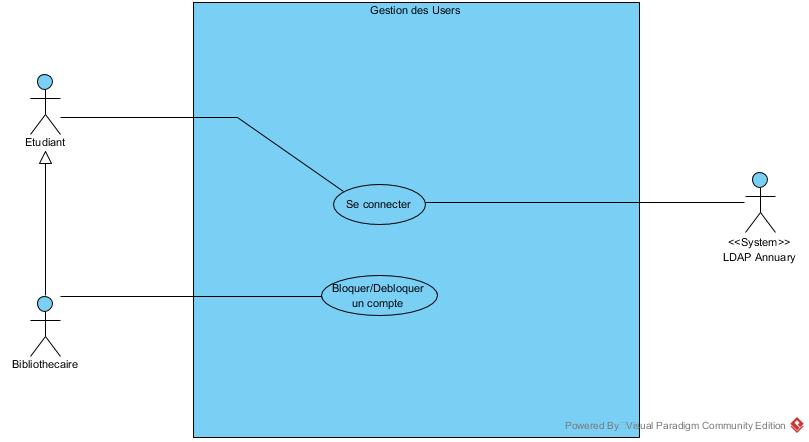


Figure 3 : Diagramme de cas d'utilisation gestion des users

#### Diagramme de classe

C’est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

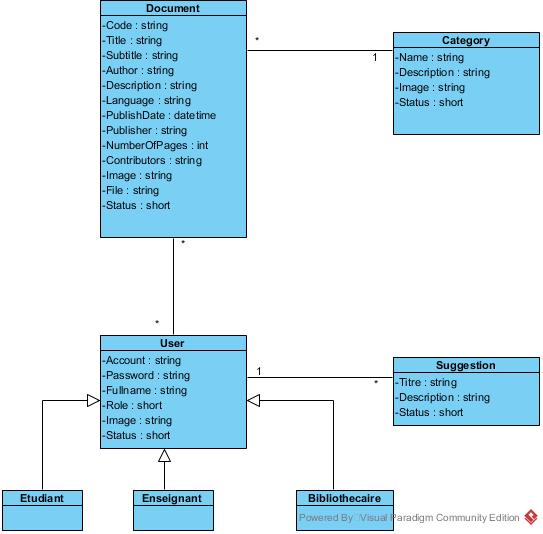


Figure 5 : Diagramme de classe

#### Diagramme de déploiement

C’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.

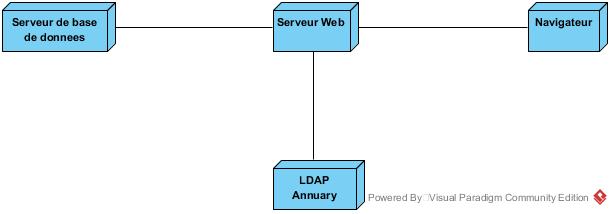


Figure 6 : Diagramme de déploiement

### Vue dynamique du système

#### Diagramme d’activité

C’est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multi-threads ou multiprocessus).

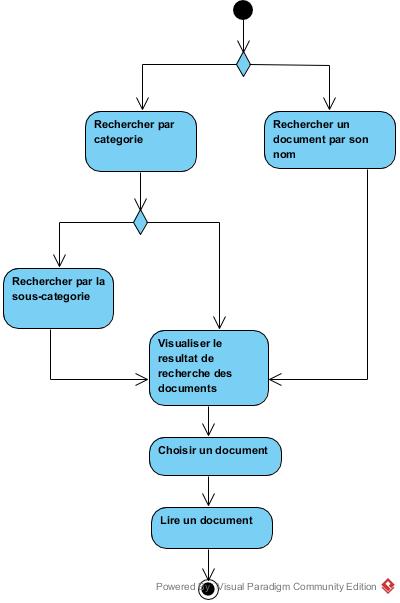


Figure 7 : Diagramme d'activité passer une commande

#### Diagramme de séquence

C’est un diagramme qui permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.

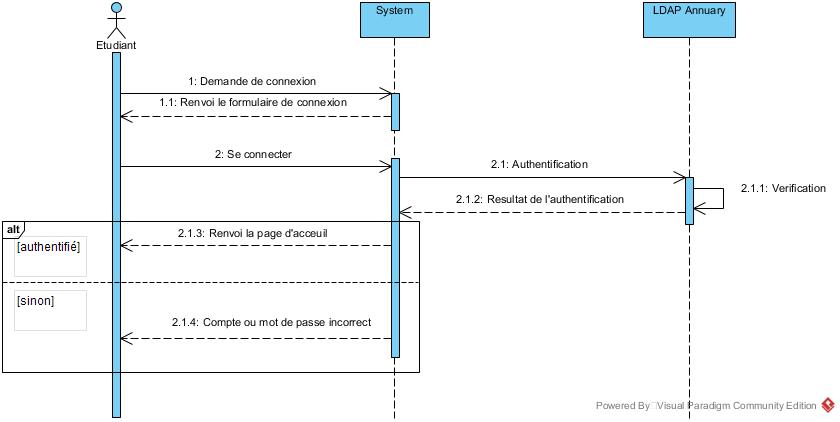


Figure 10 : Diagramme de séquence s’authentifier

# 

# Gestion du projet

## Organisation du travail

Le temps alloué pour réaliser le projet est de 02 mois, réalisé par un diagramme de Gantt basé sur le cycle en V.

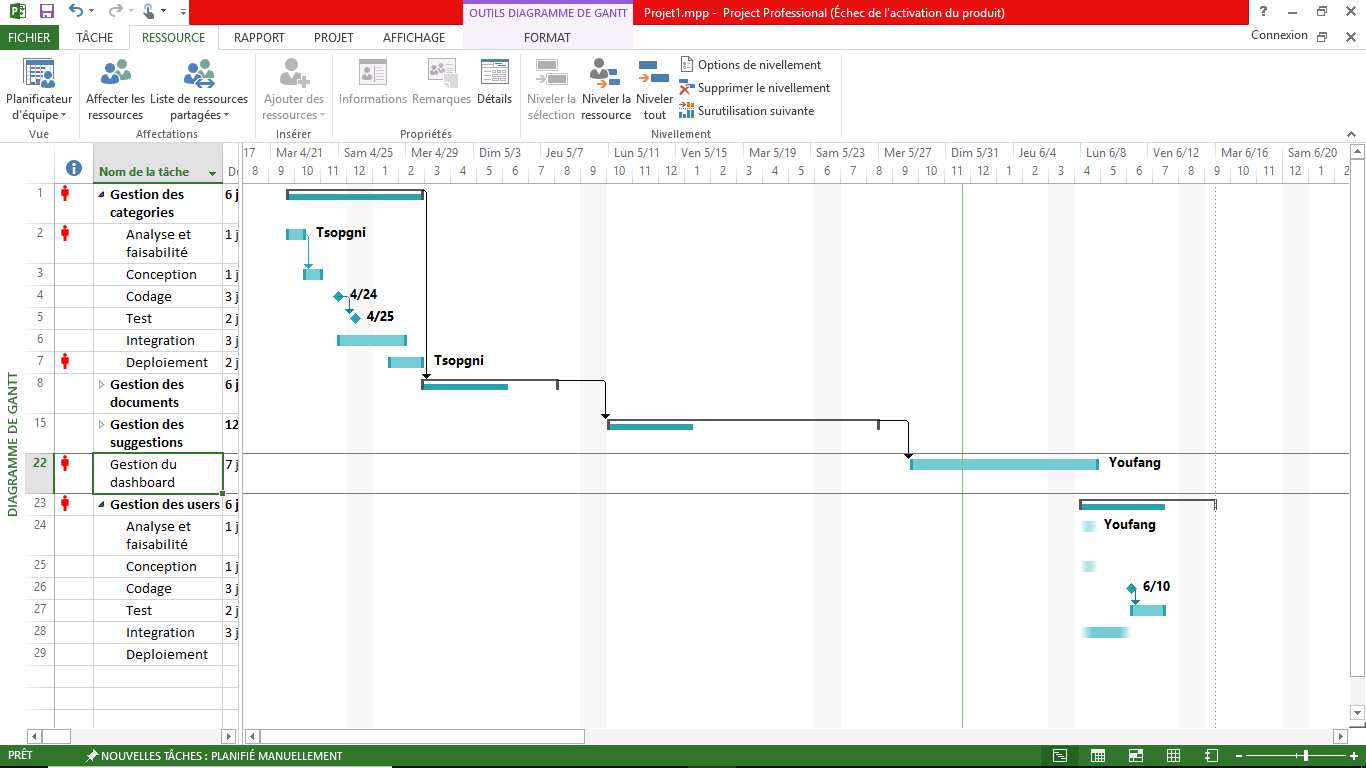


Figure 11 : Diagramme de Gantt

## Communication interne

Pour mener à bien notre projet, nous avons utilisé **Telegram** pour faciliter les échanges et **Github** où nous échangions tous nos travaux. Le travail a été subdivisé en **Milestones**, qui quant à eux, renfermaient des taches à réaliser.

## 

## Business model

Concevoir cette application n’a pas été chose facile. En effet, il a fallu dépenser dans le management de l’équipe, dans le fournissement en internet. Malgré cela, cette application est entièrement gratuite pour tous. Néanmoins, cette application pour se faire connaitre a besoin d’un plan de marketing, qui ne coute pas rien.

Initialement nous avons un budget de 0 FCFA. Mais nous comptons faire financer notre projet en se rapprochant d’éleveurs de fonds : banques, organismes de financement et particuliers car nous avons besoin d’un budget initial de 10 Millions FCFA de marketing et de communication sur une durée d’un an réparti comme suit :

Tableau 3 : Plan de communication

|  |  |
| --- | --- |
| DEPENSES | MONTANT |
| Spot Radio | 1 Million |
| Spot Télé | 2 Millions |
| Panneaux publicitaires | 3 Millions |
| Flyers | 500 Mille |
| Réseaux Sociaux | 1 Millions |
| Conférences & Séminaires | 1 Millions |
| Autres | 1,5 Millions |
| TOTAL | **10 MILLIONS** |

Ce budget de communication sera bien évidemment doublé chaque année soit 310 Millions sur 5 ans avant de rester stable pour les 5 prochaines autres années.

Nous espérons rentabiliser le projet en prenant 50 FCFA pour chaque transaction dans notre système. Le tableau suivant montre donc le nombre de transaction estimé sur une période de 5 ans :

Tableau 4 : Rentabilisation du projet

|  |  |  |
| --- | --- | --- |
| ANNEE | NOMBRE DE TRANSACTION | MONTANT |
| 1ère année | 1000/jour soit 365 000/an | 50\*365 000 = 18 250 000 |
| 2e année | 2000/jour soit 730 000/an | 50\* 730 000 = 36 500 000 |
| 3e année | 5000/jour soit 1 825 000/an | 50\*1 825 000 = 91 250 000 |
| 4e année | 10 000/jour soit 3 650 000/an | 50\*3 650 000 = 182 500 000 |
| 5e année | 20 000/jour soit 7 300 000/an | 50\*7 300 000 = 375 000 000 |
| TOTAL |  | **703 000 000** |

Ça fait donc un taux d’accroissement en moyenne de **50** **%** par an. Et des bénéfices à partir de la 1ere année.

# Réalisation du projet

## Développement de la base de données

### Présentation et téléchargements des outils

Microsoft SQL Server est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) (SGBD) en langage [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language) - qui est le langage standardisé de manipulation des bases de données - incorporant entre autres un SGBDR (SGBD [relationnel](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle) ) développé et commercialisé par la société [Microsoft](https://fr.wikipedia.org/wiki/Microsoft). Nous avons l’avons choisi comme SGBD car il a comme avantage majeur, un niveau de sécurité élevé. Il est téléchargeable à l’adresse suivante : <https://www.microsoft.com/fr-fr/sql-server/sql-server-downloads>. L’exécutable téléchargé contient en son sein le moteur de base de données SQL Server et SQL Server Management Studio (SSMS), outil graphique de manipulation des bases de données SQL Server. Son installation se fait comme suit :

* Choix de SQL Server

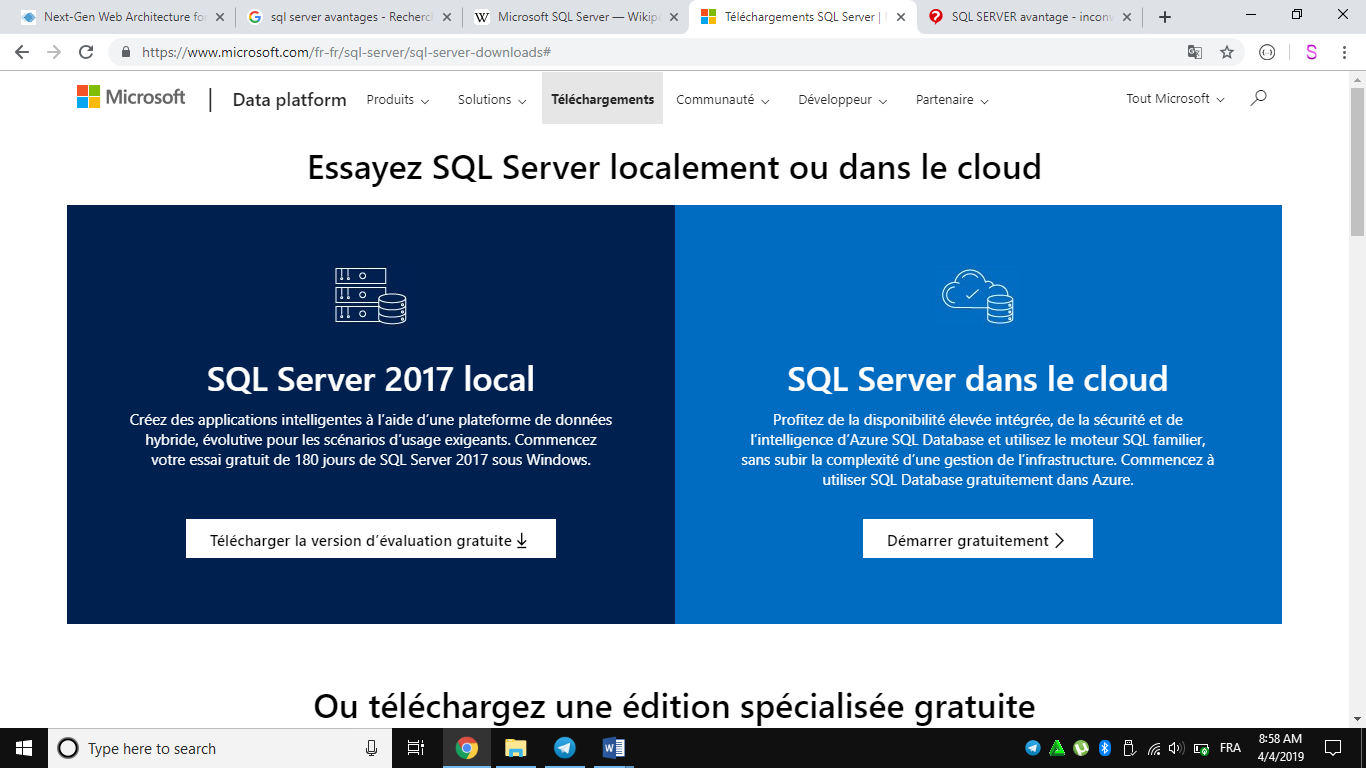


Figure 12 : Choix de SQL Server

* Inscription

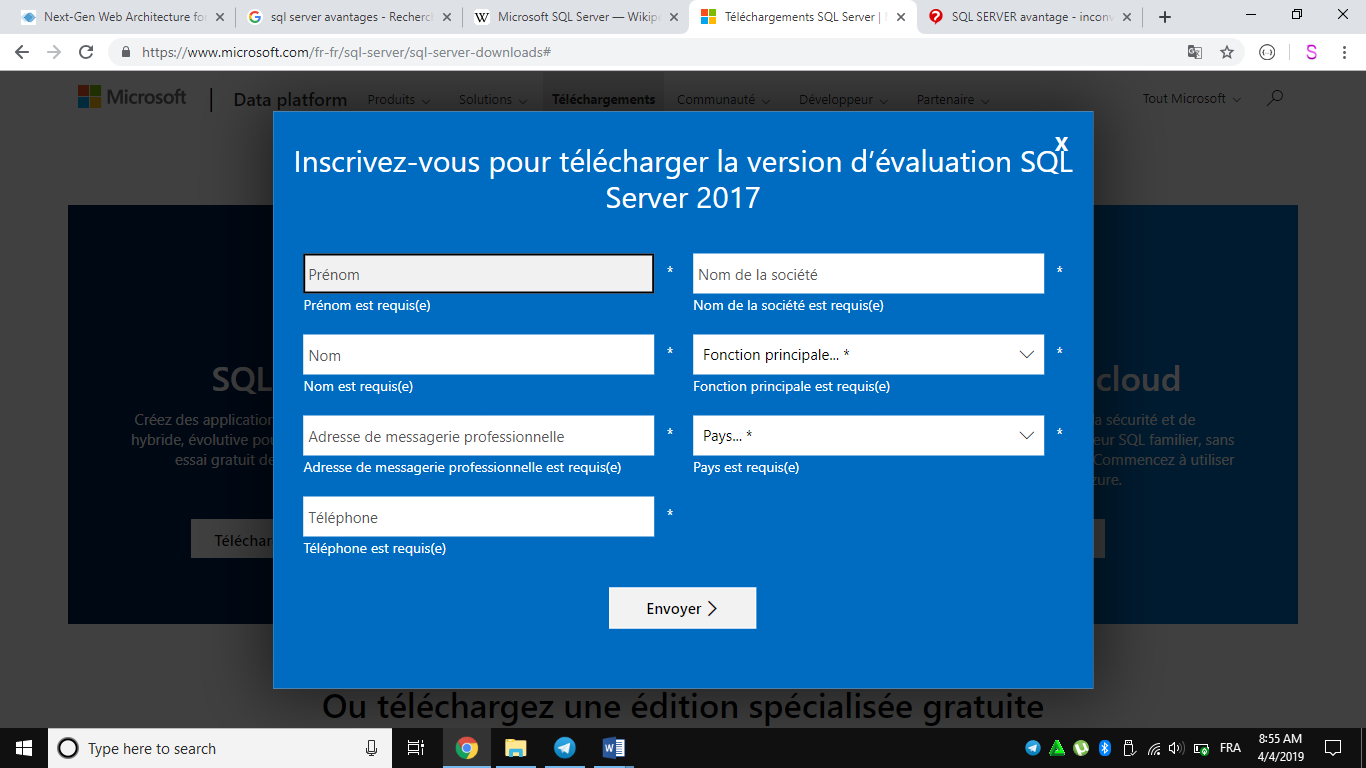


Figure 13 : Inscription pour téléchargement

* Téléchargement

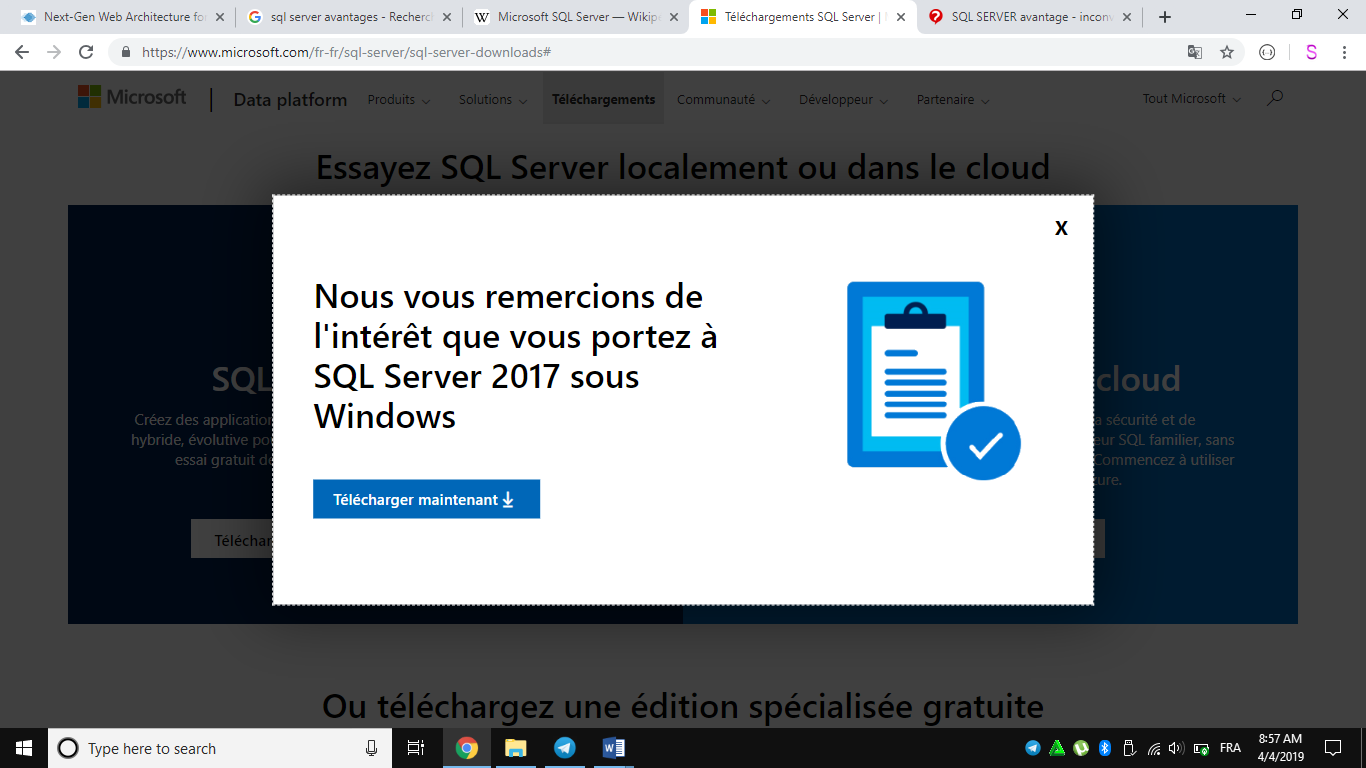


Figure 14 : Téléchargement

### Création de la base de données, des tables

#### La base de données

Pour créer une base de données sous SQL Server, nous pouvons procéder par l’interface graphique ou alors par la syntaxe SQL.

* Graphiquement : une fois créée, notre base de données apparaitra dans la liste des bases de données.

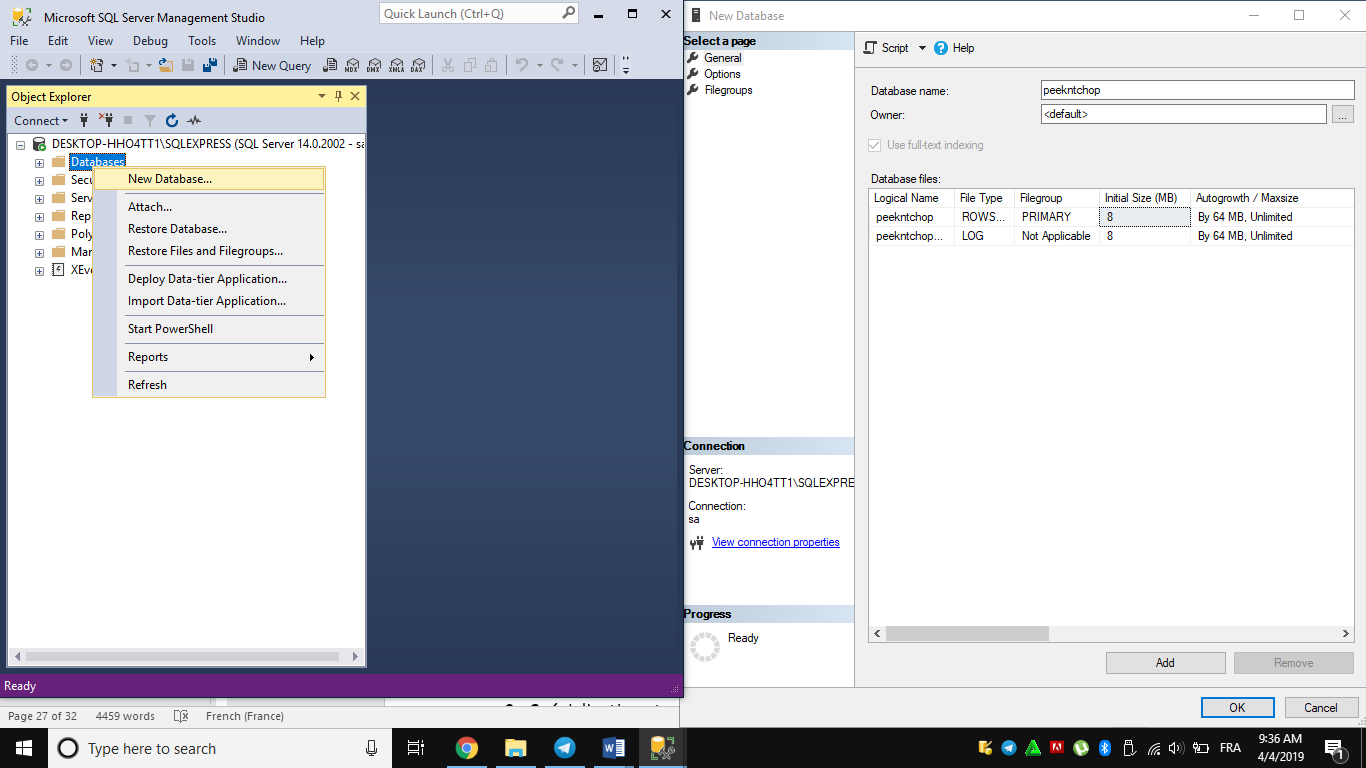


Figure 15 : Création de la base de données graphiquement

* Syntaxe SQL : nous cliquons sur l’onglet « New Query », ensuite nous écrivons notre syntaxe et enfin, nous l’exécutons.

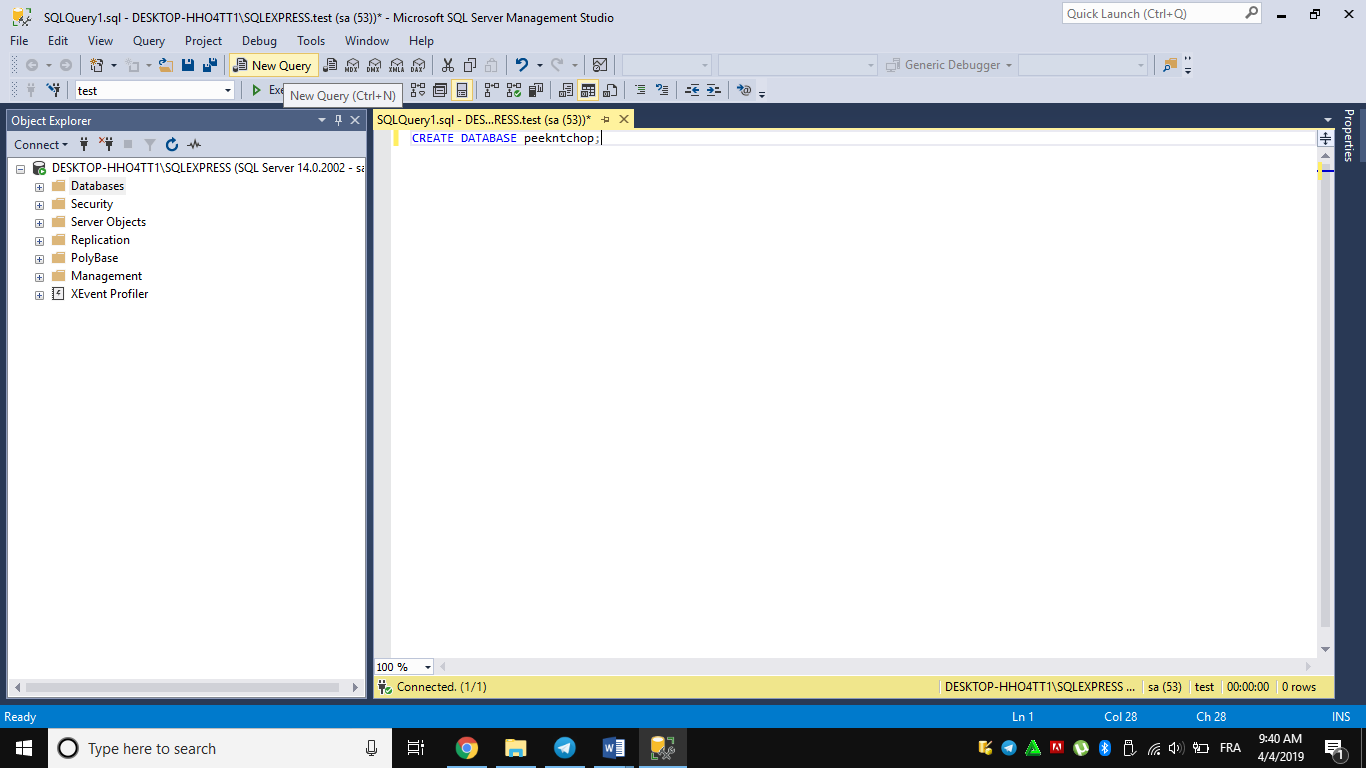


Figure 16 : Création de la base de données par la syntaxe SQL

#### La table

Pour créer une table sous SQL Server, nous pouvons procéder par l’interface graphique ou alors par la syntaxe SQL.

* Graphiquement : nous cliquons sur notre base de données, nous avons fait un clic droit sur tables et cliqué sur « **Table** » ; Ainsi nous renseignons nos attributs et leurs types.

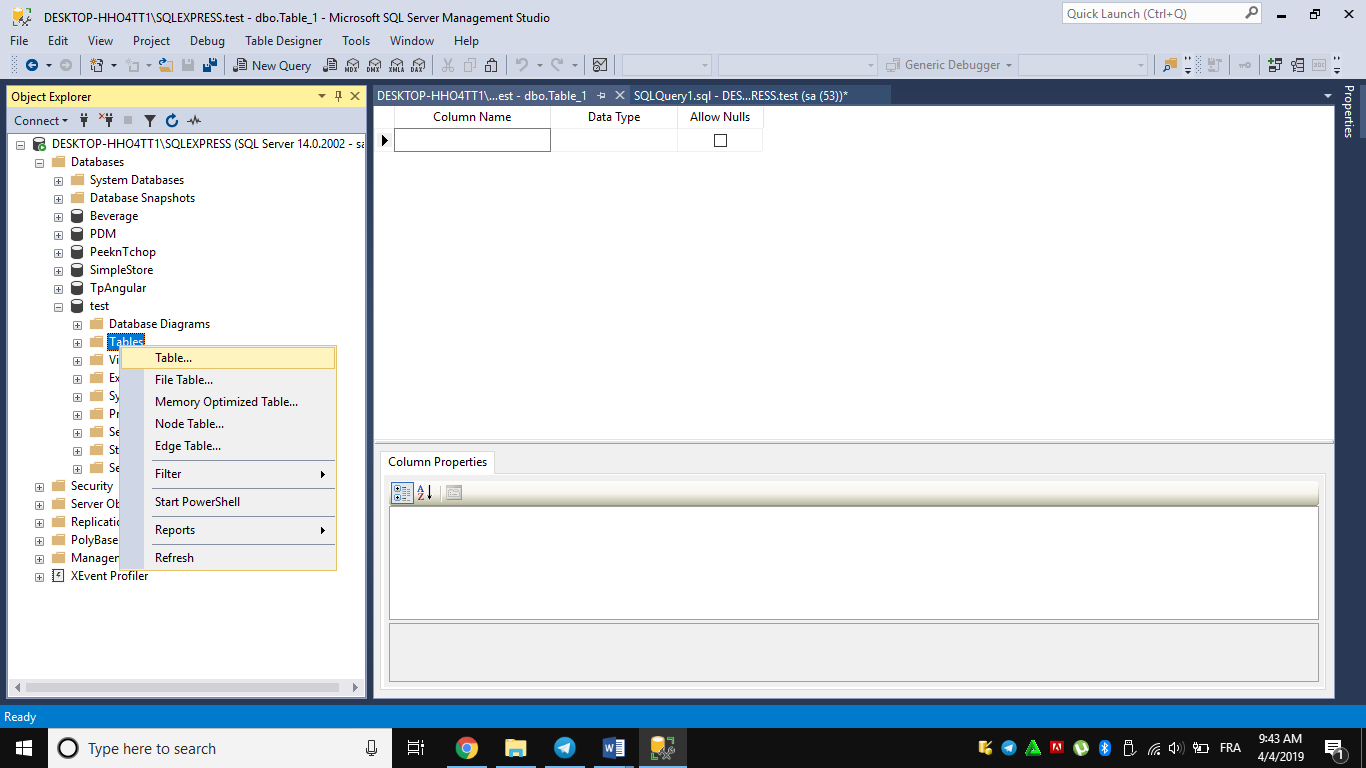


Figure 17 : Création d'une table graphiquement

* Syntaxe SQL : nous cliquons sur l’onglet « **New Query** », après avoir spécifié quelle base de données nous allons utiliser, nous avons écrit notre syntaxe et enfin, nous l’avons exécuté.

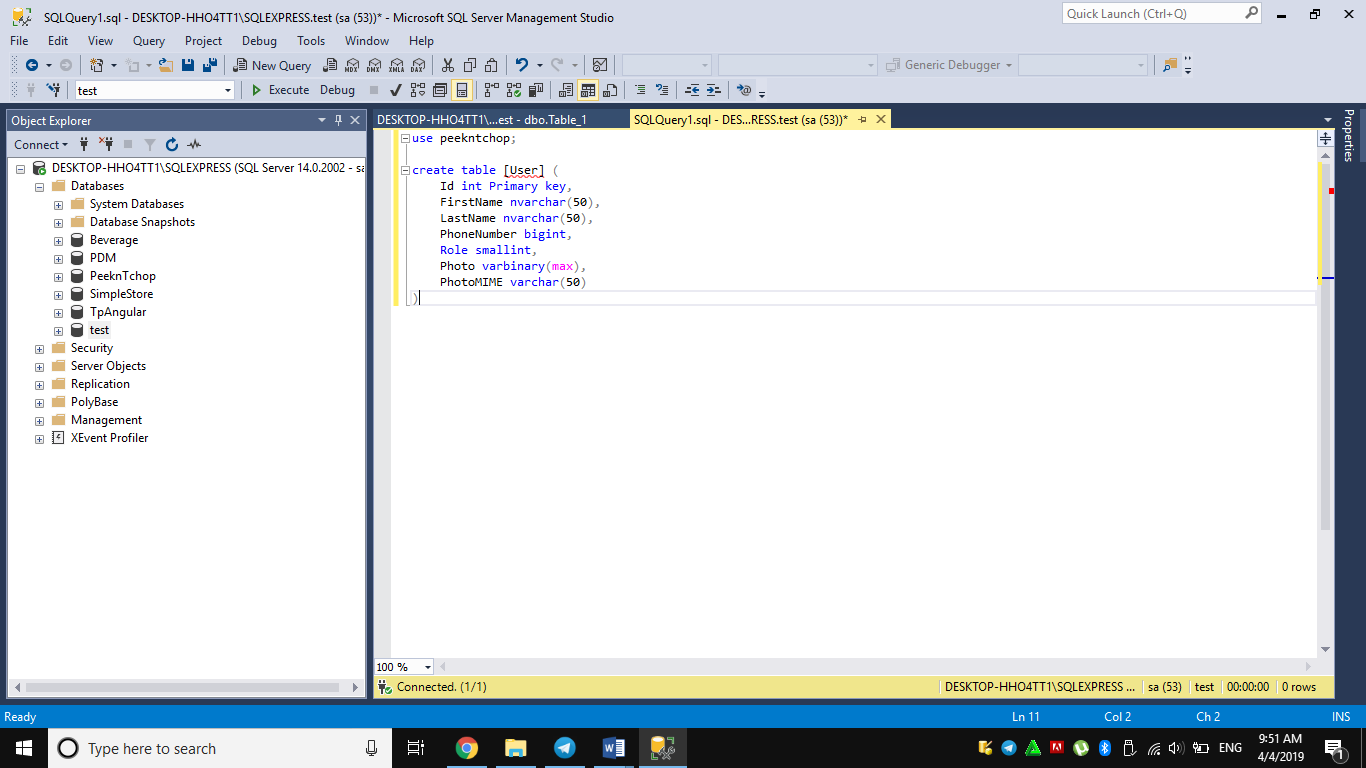


Figure 18 : Création d'une table par syntaxe SQL

## Implémentation des services

### Architecture du logiciel

Pour ce logiciel nous avons opté pour l’architecture orientée service car il permet une réutilisabilité possible des composants, une meilleures possibilités d’évolution et une plus grande tolérance aux pannes. Comme le montre le schéma ci-dessous elle offre une grande souplesse pour le développement des applications reparties ou distribuées.

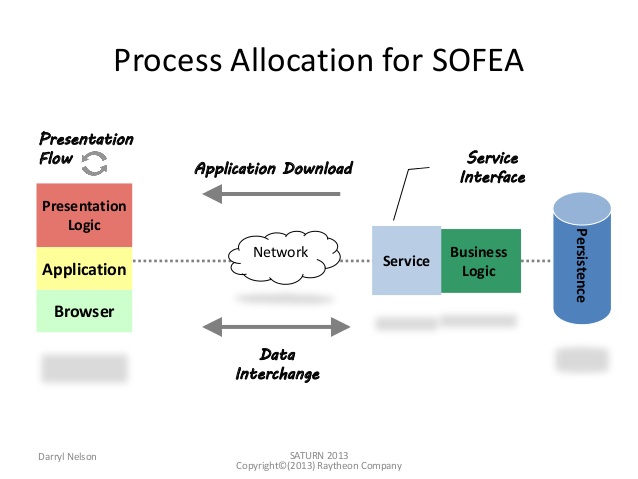


Figure 19 : Architecture Orientée Service par Darryl Nelson

Le pattern MVC, qui est l’un des plus célèbres design pattern, permet de bien organiser son code source. Il vous aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

* **Modèle** : cette partie gère les données de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.
* **Vue** : cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.
* **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

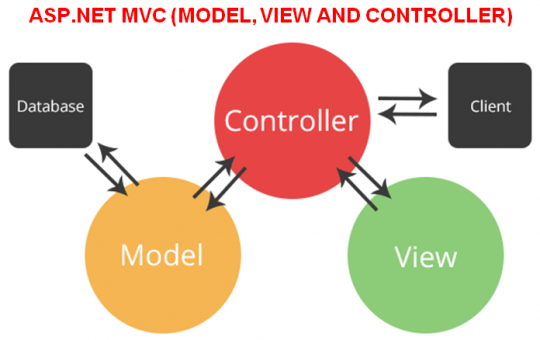


Figure 20 : Architecture MVC

### Présentation et installation des outils

#### ASP.NET et Visual Studio

ASP.NET est une technologie créée par Microsoft permettant la réalisation d'un site Web. En technologies, nous avons le **Framework .NET** et **ASP.NET**. Le Framework .NET, nous permet d’accéder à une large gamme de fonctionnalités telles que la création de contrôles, l'accès aux données, des classes destinées au développement Windows. ASP.NET se patche au framework .NET, il va procurer des outils, des contrôles et des classes toutes prêtent spécialement destinées au développement de sites Web. Le **C#** est le langage de programmation crée par Microsoft et qui permet de construire toute sorte d'applications (Windows, jeux, mobiles). Il se couple parfaitement avec le Framework .NET. Pour coder avec ASP.NET, il nous faut des outils adaptés. Nous allons utiliser du C# mêlé avec de l'ASP.NET pour créer facilement nos site Web mais aussi **Visual Studio 2017**, qui est un environnement de développement intégré qui va fournir tous les outils nécessaires au développement d'une application Windows ou d'un site Web. Il est téléchargeable à l’adresse suivante : <https://visualstudio.microsoft.com/downloads/>

Son installation se fait comme suit :

* Choix de Visual Studio : dans notre cas nous avons Visual Studio 2017 Professionnel

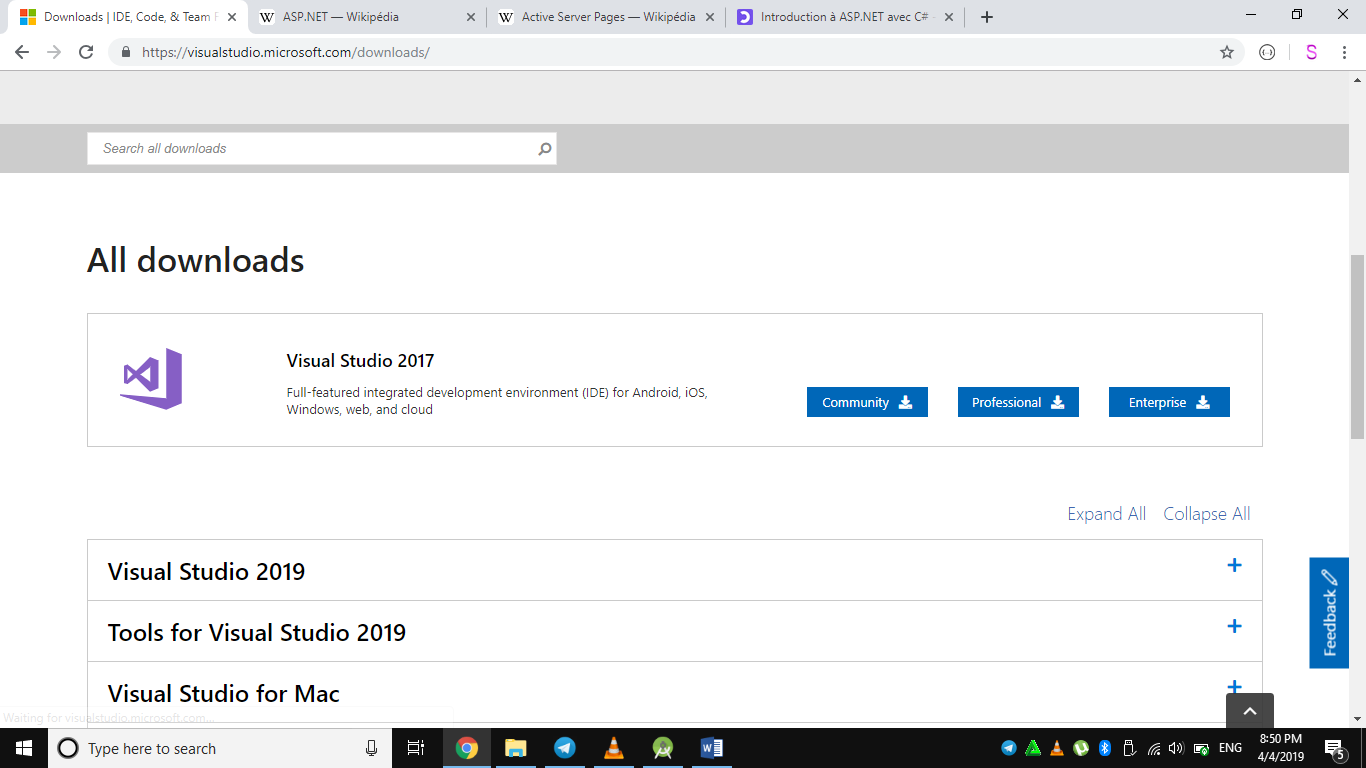


Figure 21 : Choix de Visual Studio

Lors de l’installation, il faut cocher le module ASP et ainsi il sera incorporé dans Visual Studio. Une fois installée, vous aurez une interface de démarrage présentant les projets récents déjà crées, ou encore vous demandant de créer un nouveau projet :

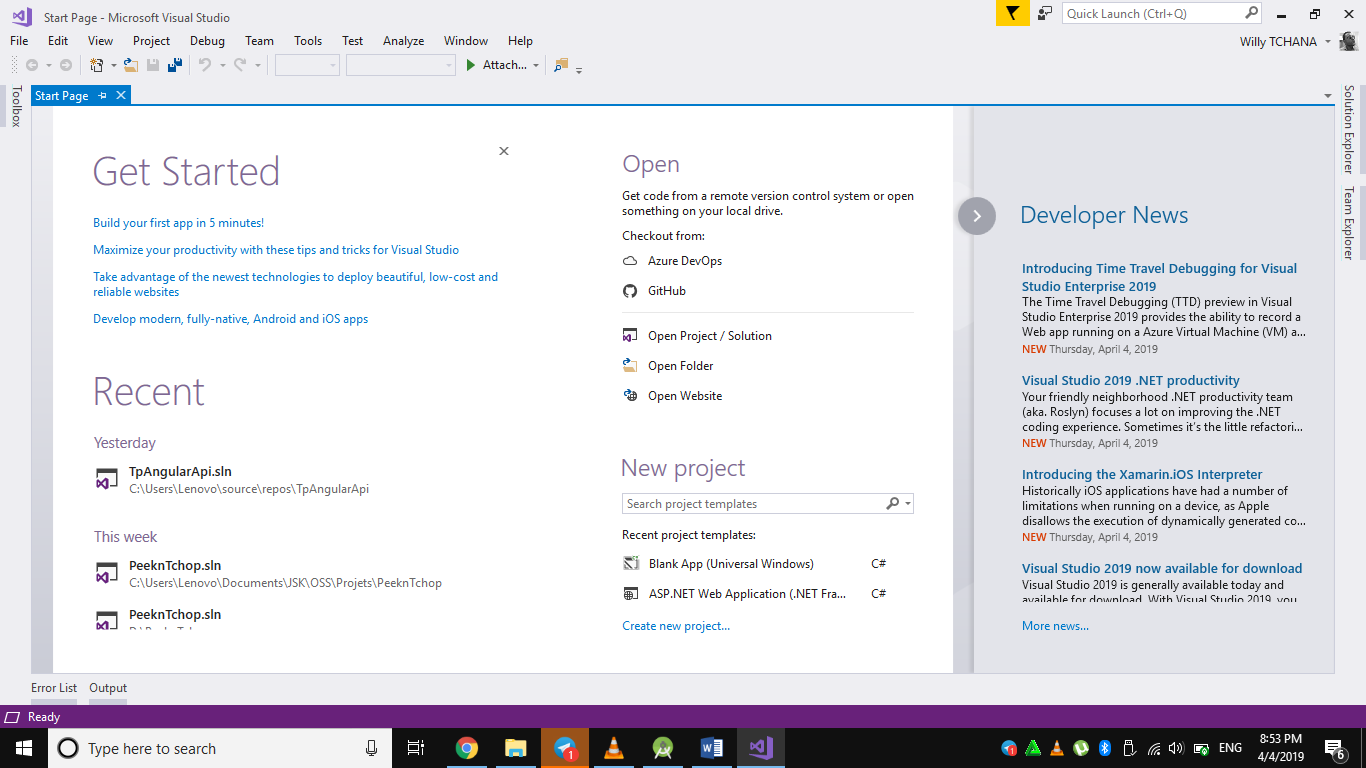


Figure 22 : Interface de demarrage

Pour créer un projet ASP.NET, nous avons cliqué sur « **New project** » et choisi « **ASP.NET Web Application** ». Il est important de mentionner qu’il faut différencier le nom de la solution et nom du projet que vous créez ; car dans la solution, nous pouvons avoir plusieurs petits projets.

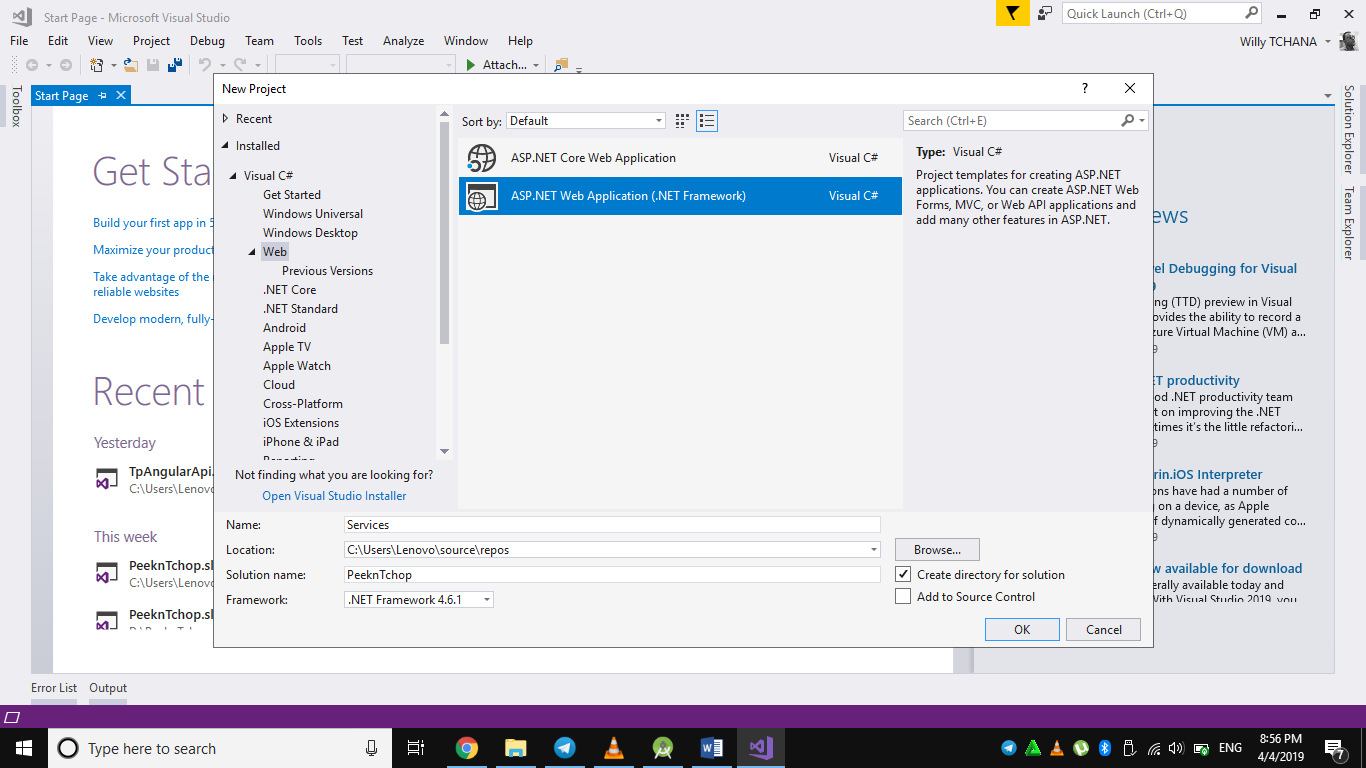


Figure 23 : Creation d'un projet ASP.NET

Après avoir cliqué sur « **Ok** », une autre fenêtre apparaitra où il faudra cocher « **Empty** » et « **MVC** » et enfin « **Ok** ». Le projet est ensuite créé.

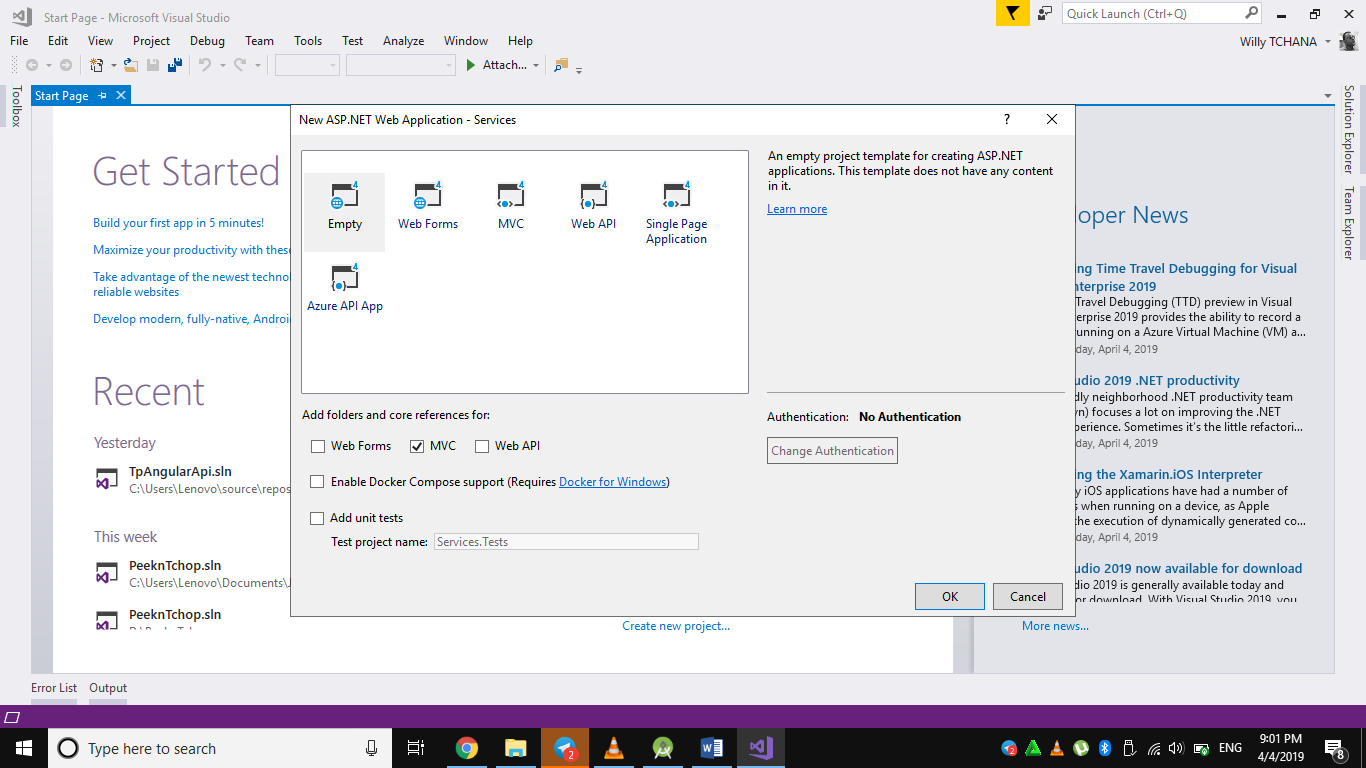


Figure 24 : Creation d'un projet ASP.NET (Suite)

#### Mapping avec la base de données

Pour connecter la base de données à notre projet, nous avons utilisé **Entity Framework**. C’est un outil permettant de créer une couche d'accès aux données (DAL pour Data Access Layer) liée à une base de données relationnelle. Pour cela, on ajoute un nouveau fichier de type « **ADO.NET Entity Data Model** » dans le dossier **Models** du projet et après nous avons sélectionné « **EF Designer from database** ».

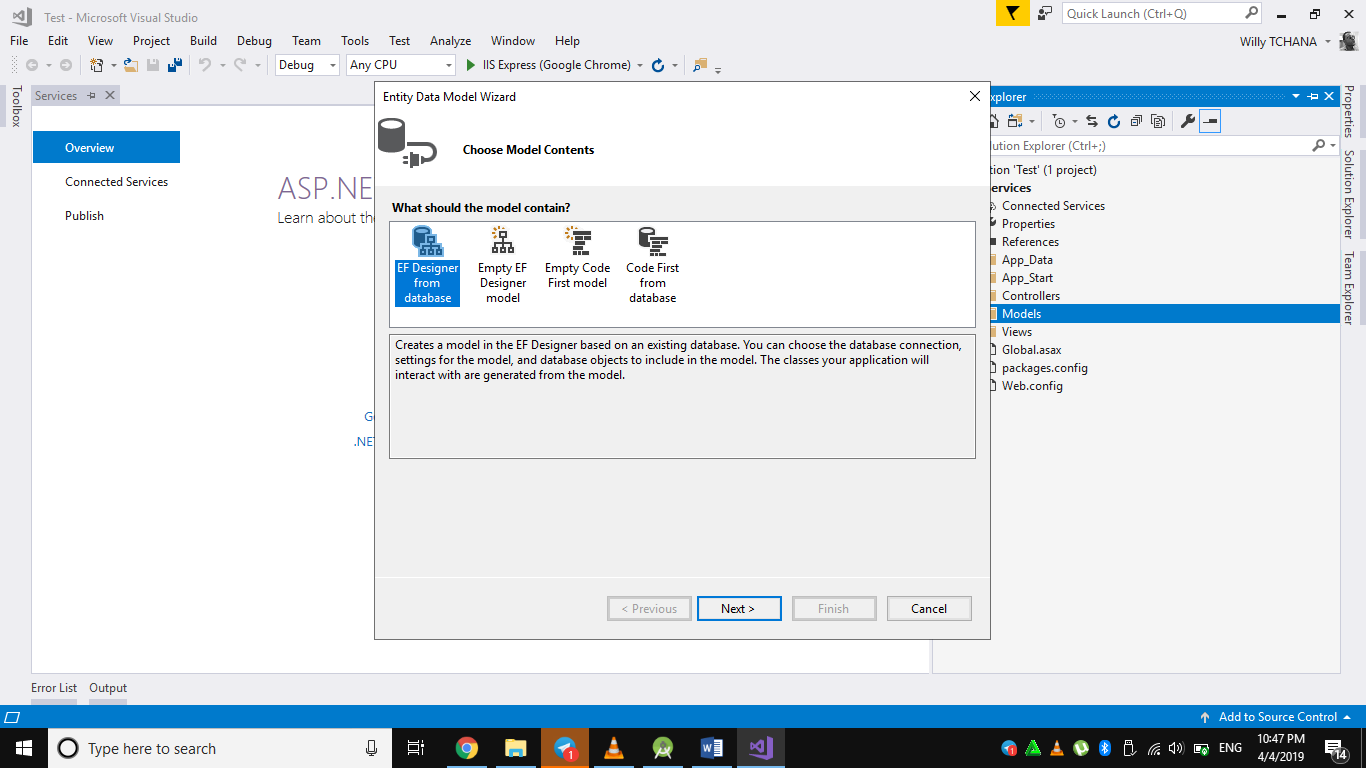


Figure 25 : Mapping avec la base de données

Après avoir cliqué sur « **New Connection** », nous avons renseigné le nom de notre serveur et ensuite choisi la base de données correspondante.

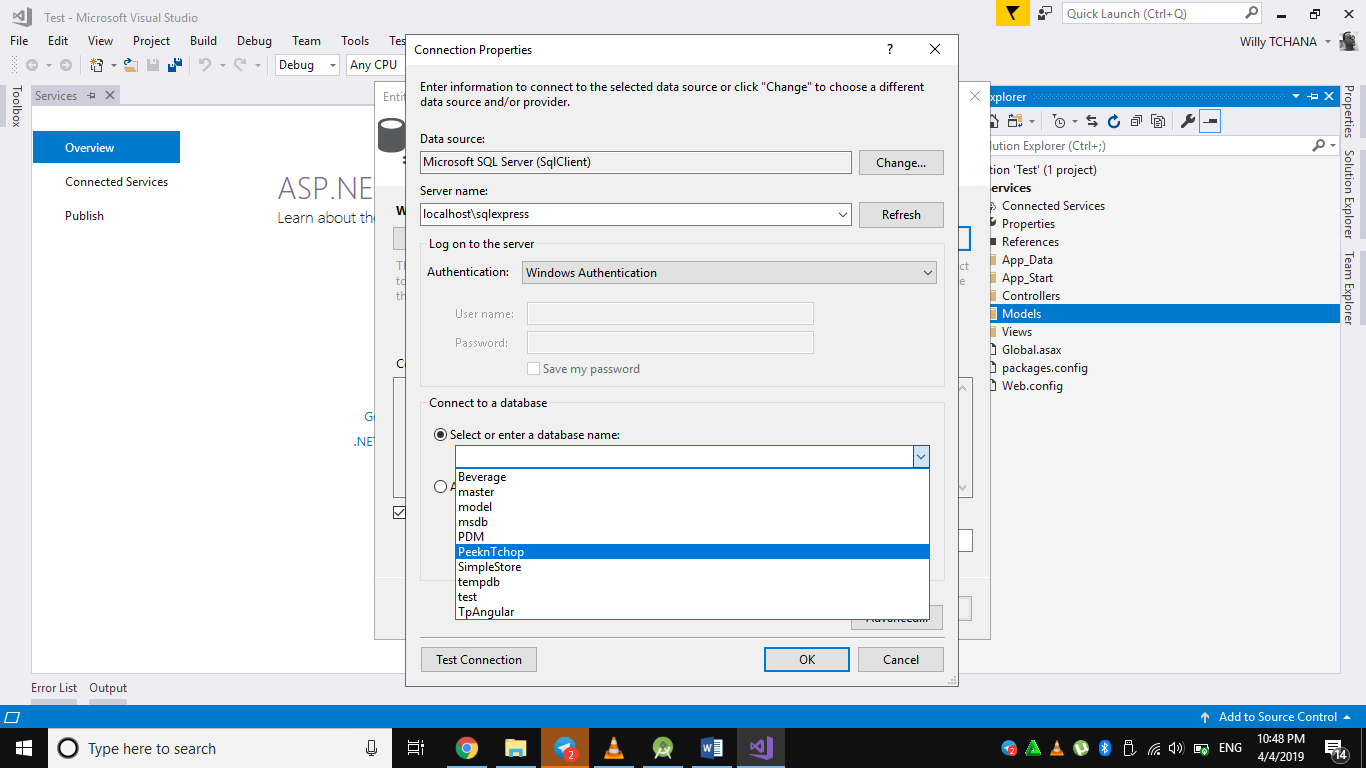


Figure 26 : Mapping avec la base de données (Suite)

Nous avons ensuite choisi les objets que nous allons inclure dans le modèle.

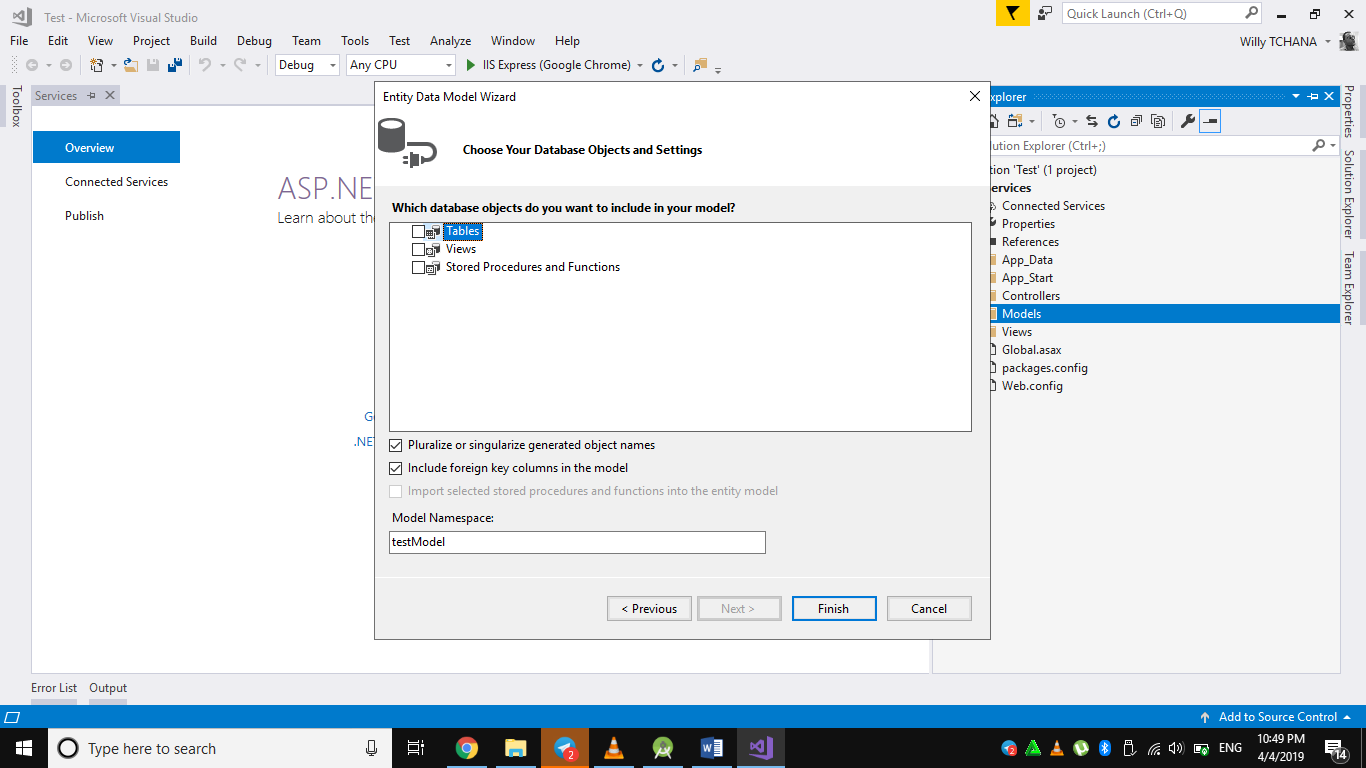


Figure 27 : Mapping avec la base de données (Suite)

#### Création des controllers

Pour créer un Controller, il suffit de faire un clic droit sur le dossier « **Controllers** » du projet, cliquer sur **Add** et sélectionner **Controller**.

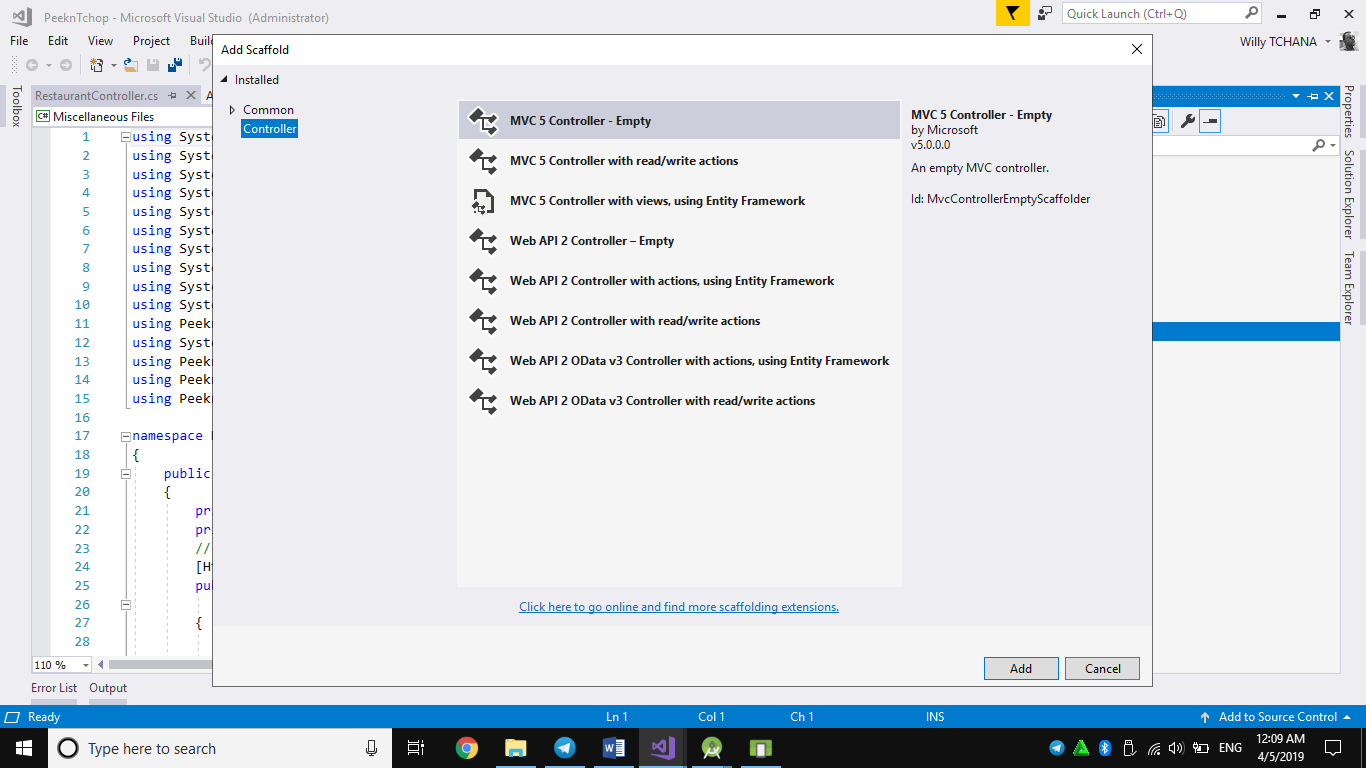


Figure 28 : Création d’un controller

Après avoir choisi la structure de notre Controller, il faut lui donner un nom. Il est important de savoir que le nom est toujours suffixé du mot **Controller**.

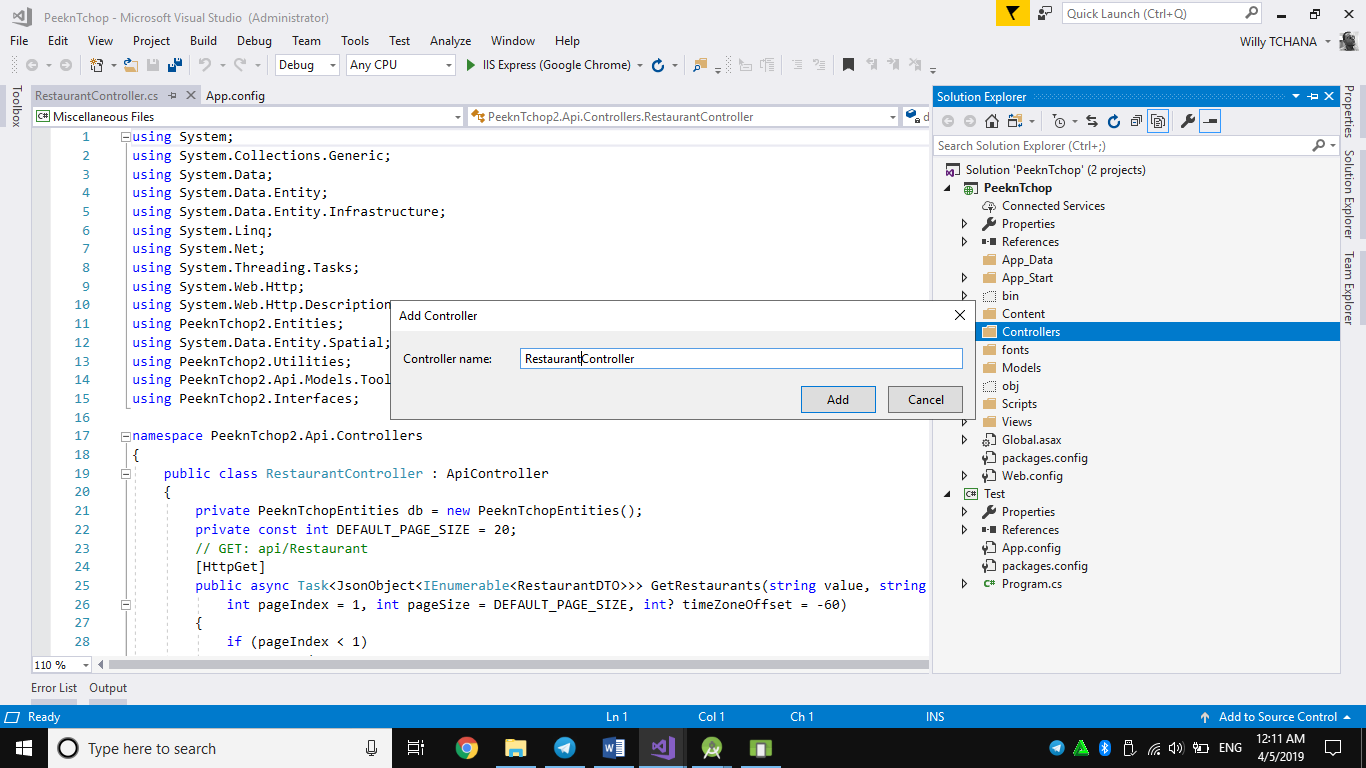


Figure 29 : Création d'un controller (Suite)

RestaurantController.cs est le fichier créé dans lequel nous allons écrire le code des différentes actions possibles sur le modèle **Restaurant**. Avant, nous devons créer une instance de **PeeknTchopEntities** pour pouvoir accéder aux objets de la base de données.

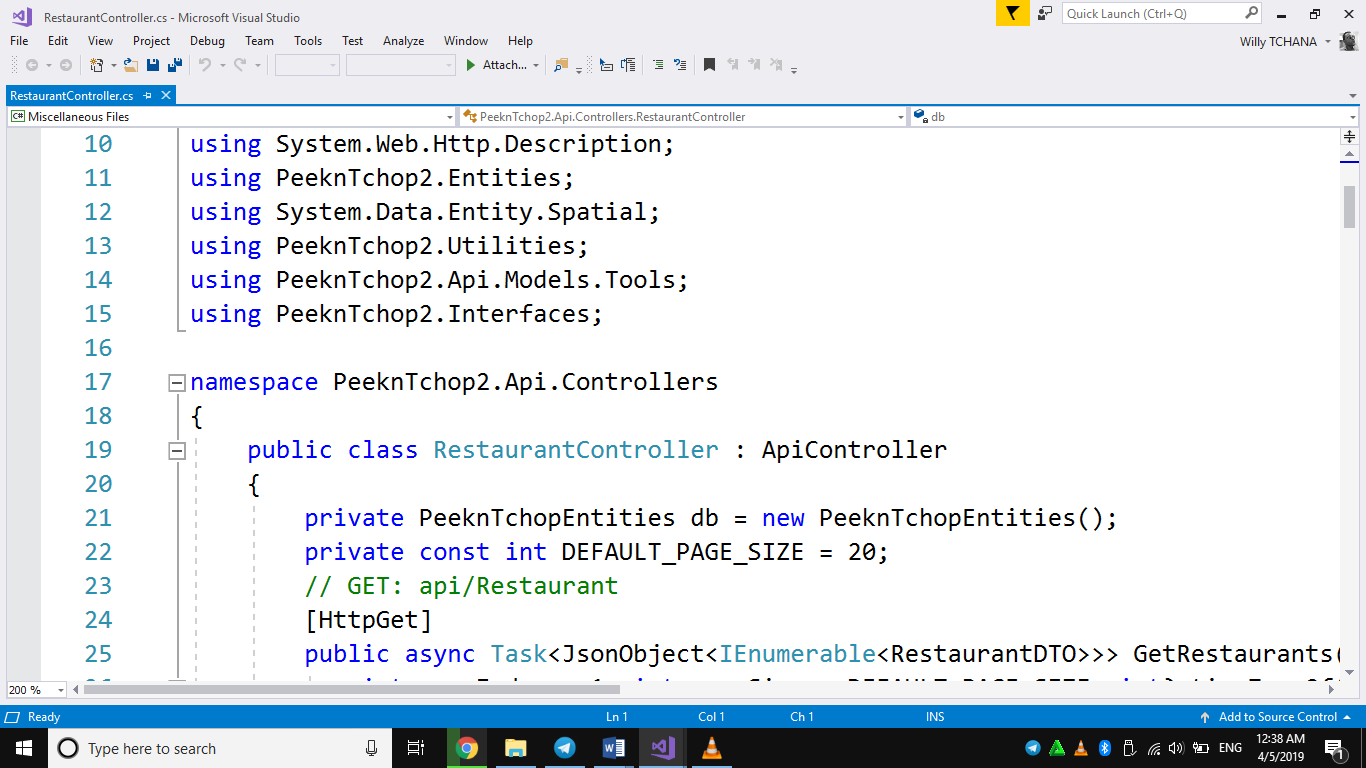


Figure 30 : Création d'une instance de PeeknTchopEntities

Cette action qui prend en paramètre un **id** de type **int**, permet d’afficher un restaurant spécifique.

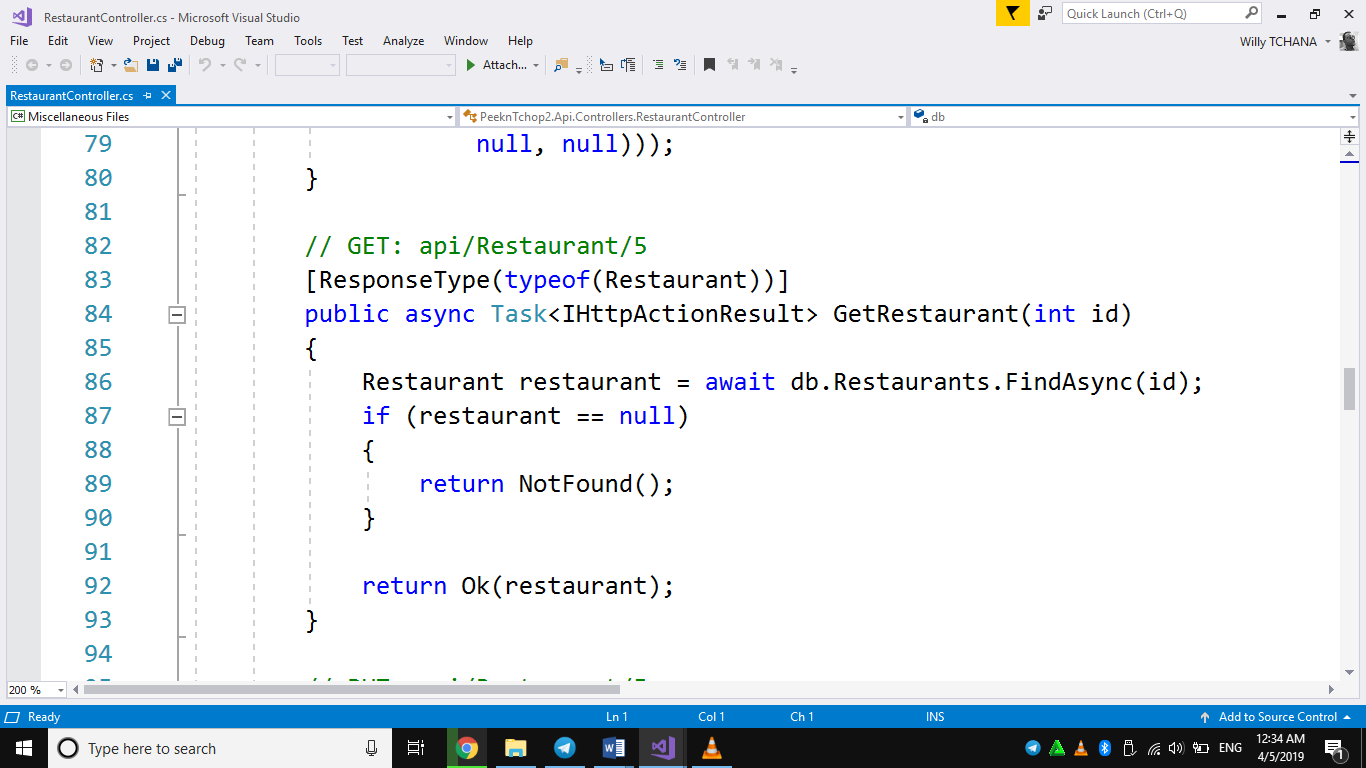


Figure 31 : GetRestaurant by id

Cette action qui prend en paramètre un **id** de type **int**, permet de supprimer un restaurant spécifique.

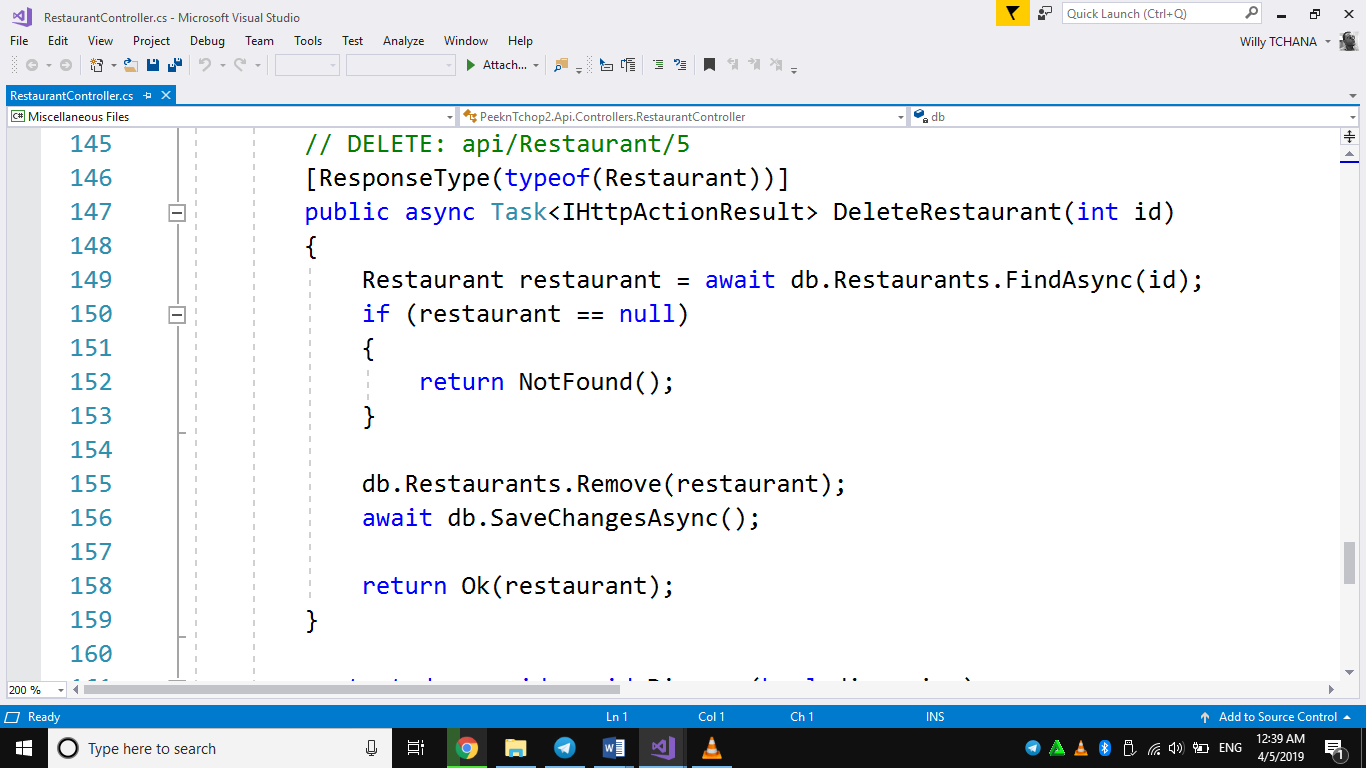


Figure 32 : DeleteRestaurant by id

On répètera cette manœuvre pour chacun des modèles.

#### Test des services

La notion de test est fondamentale en programmation. En effet, elle permet de se rassurer de la véracité et de la justesse de la fonctionnalité implémentée. Pour tester chacune de nos fonctions, nous utiliserons **Postman.** Il permet d'**envoyer toutes sortes de requêtes et les personnaliser très finement.** Il est téléchargeable à l’adresse suivante : <https://www.getpostman.com/downloads/>.

Une fois téléchargé, il présente l’interface de démarrage suivant :

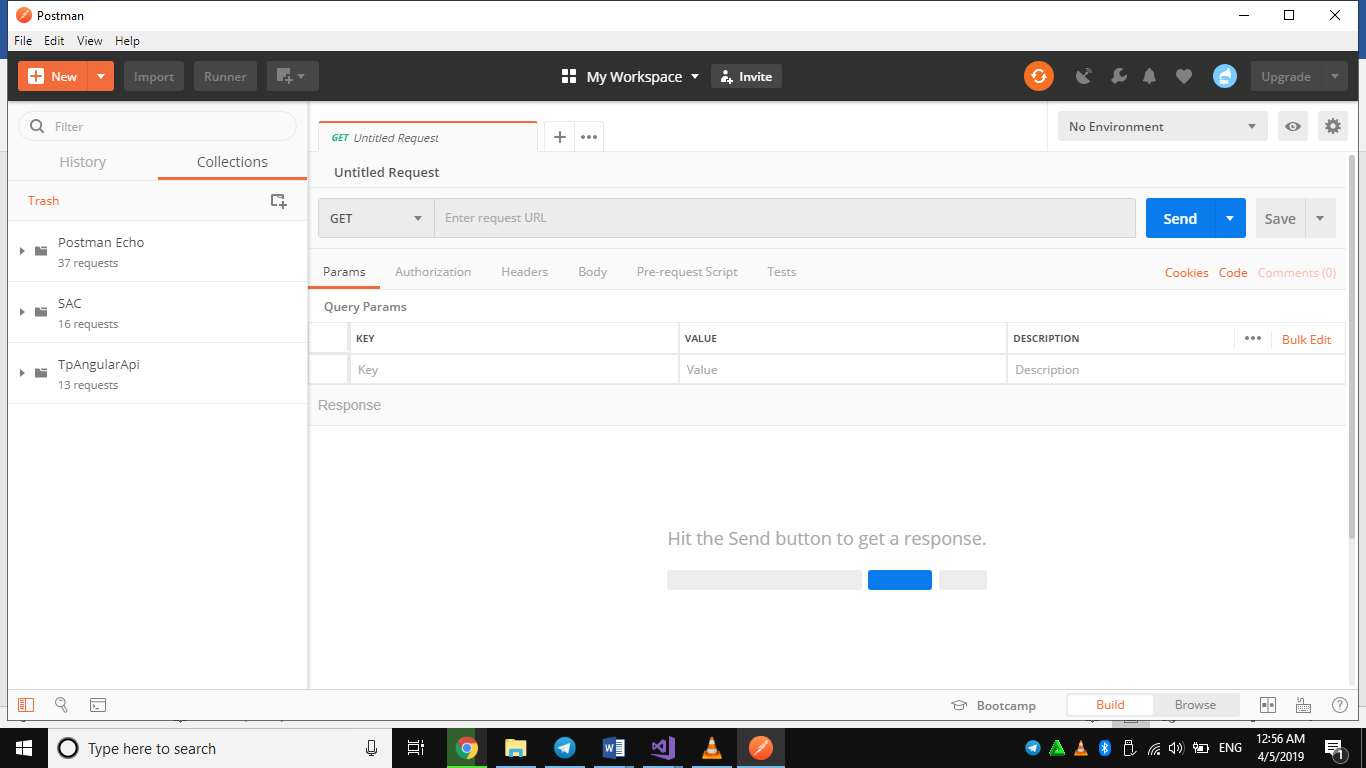


Figure 33 : Interface de démarrage de Postman

Après avoir démarré le projet sous Visual Studio, nous avons copie le lien et l’avons copié dans la barre de requête URL dans Postman.

## Interfaces graphiques

### La page d’accueil

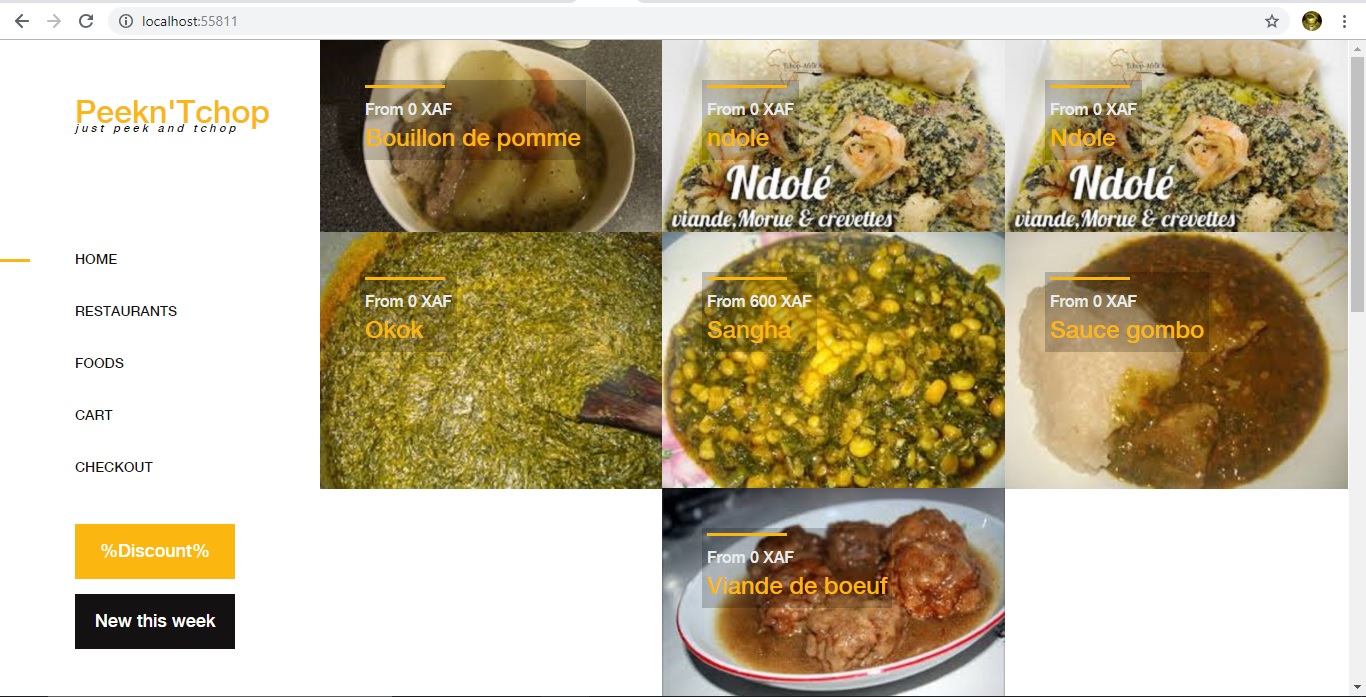


Figure 34 : Page d'accueil de notre application

### La page des restaurants

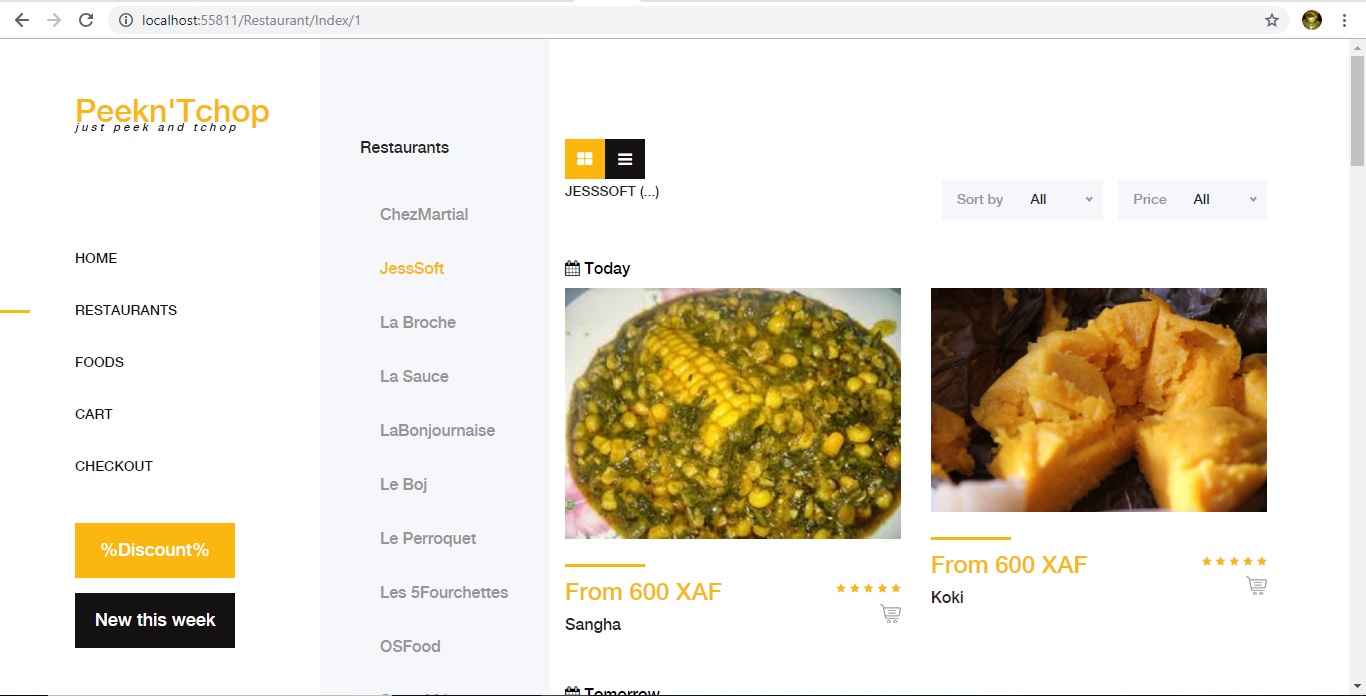


Figure 35 : Page des restaurants de notre application

# Conclusion

En définitive, il est question dans ce document de présenter le cahier de charge d’une solution résolvant un problème donné, de son analyse et ainsi que sa conception du système. Nous avons tout au long de notre devoir montrer, tout en justifiant, les infrastructures que nous avons utilisées pour mettre sur pied cette application. Nous avons également montré comment nous avons organisé notre travail grâce aux outils appris en classe.

Nous pouvons constater que mener à bien un projet n’est pas une tâche facile et qu’il est nécessaire d’y accorder attention et suivi.

# Références

## Bibliographie

|  |  |
| --- | --- |
| [1] | W-J. Tchana, Cours de UML et Développement Web*,* non publié,Douala, 2018. |
| [2] | F. Yonou, Cours d’Architecture REST, non publié, Douala, 2018. |
| [3] | S. Fushe, Cours de Base de données : Requêtes, non publié, Douala, 2018. |

## Webographie

Logiciel **HotelPro**

* <http://www.camer.be/46553/11:1/cameroun-un-logiciel-africain-desormais-a-portee-de-main-pour-la-gestion-hoteliere-cameroon.html>

Date de visite : 16/01/2019

Méthodologie XP

* <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-extreme-programming>

Date de visite : 15/01/2019

Méthodologie 2TUP

* https://fr.wikipedia.org/wiki/Two\_Tracks\_Unified\_Process

Date de visite : 15/01/2019

Architecture orienté service

* <https://www.slideshare.net/kunalashar/sofea-service-oriented-front-end-architecture-next-gen-web-architecture-for-the-cloud-era>

Date de visite : 08/05/2019

# Annexes

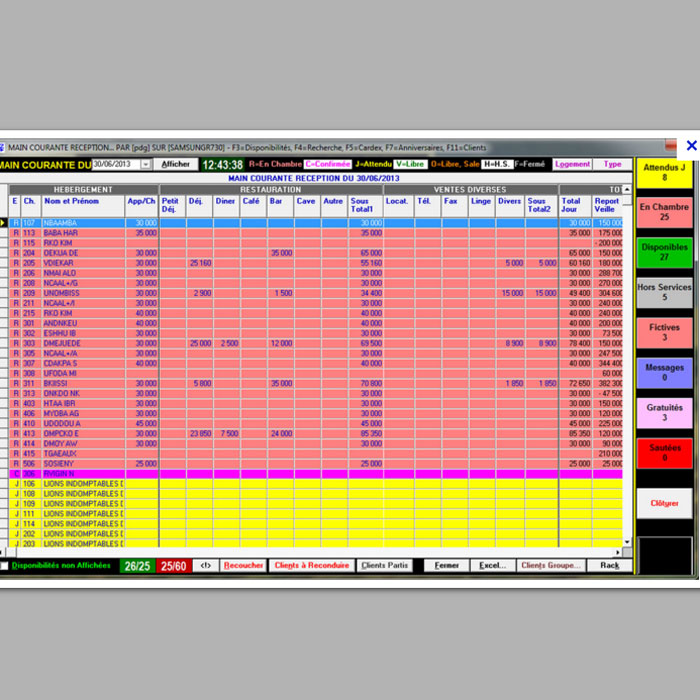


Figure 36 : Ecran principal du logiciel HotelPro