

Livrable 2 : Conception

**Projet Conception et**

**Programmation Objet**

**Groupe 4 :**

**FIHAKHIR Houda**

**BELBATI Aness**

**BENNIOU Aymen**

**Sommaire :**

1. Introduction.
2. Présentation du projet.
3. Les diagrammes.

3.1. Diagramme de cas d’utilisation

3.2. Diagramme séquence

3.3. Diagramme d’activité

3.4. Diagramme de classes.

1. Environnement de développement opérationnel.
2. Conclusion.
3. **Introduction :**

Dans ce projet, nous allons réaliser un logiciel : une architecture de type client-serveur composée d’une application et d’une base de données.

Ce projet est composé de deux parties, une première consacrée à l’appropriation du présent cahier des charges, la composition du groupe de travail et son organisation. Quant à la deuxième, nous réaliserons application et la base de données.

1. **Présentation du projet :**

Une start-up de vente en ligne de composants électroniques se lance dans un nouveau projet : développer son système d’information.

Nous sommes donc l’équipe chargée de la réalisation de la solution.

Cette dernière est un logiciel de gestion digitalisant certains de ses processus métiers. Les processus visés par notre client sont :

* La gestion du personnel ;
* La gestion des clients ;
* La gestion des commandes ;
* La gestion du stock ;
* La gestion des statistiques.

1. **Les diagrammes :**

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

Il existe trois types de digrammes, et chaque type possède plusieurs diagrammes.

*Diagrammes de comportement :*

**Diagramme des cas d'utilisation :** représentation des possibilités d'interaction entre le système et les acteurs, c'est-à-dire de toutes les fonctionnalités que doit fournir le système.

**Diagramme d’activité** : représentation sous forme de flux ou d'enchaînement d'activités du comportement du système ou de ses composants.

*Diagrammes d'interaction :*

**Diagramme de séquence :** représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

*Diagrammes de structure :*

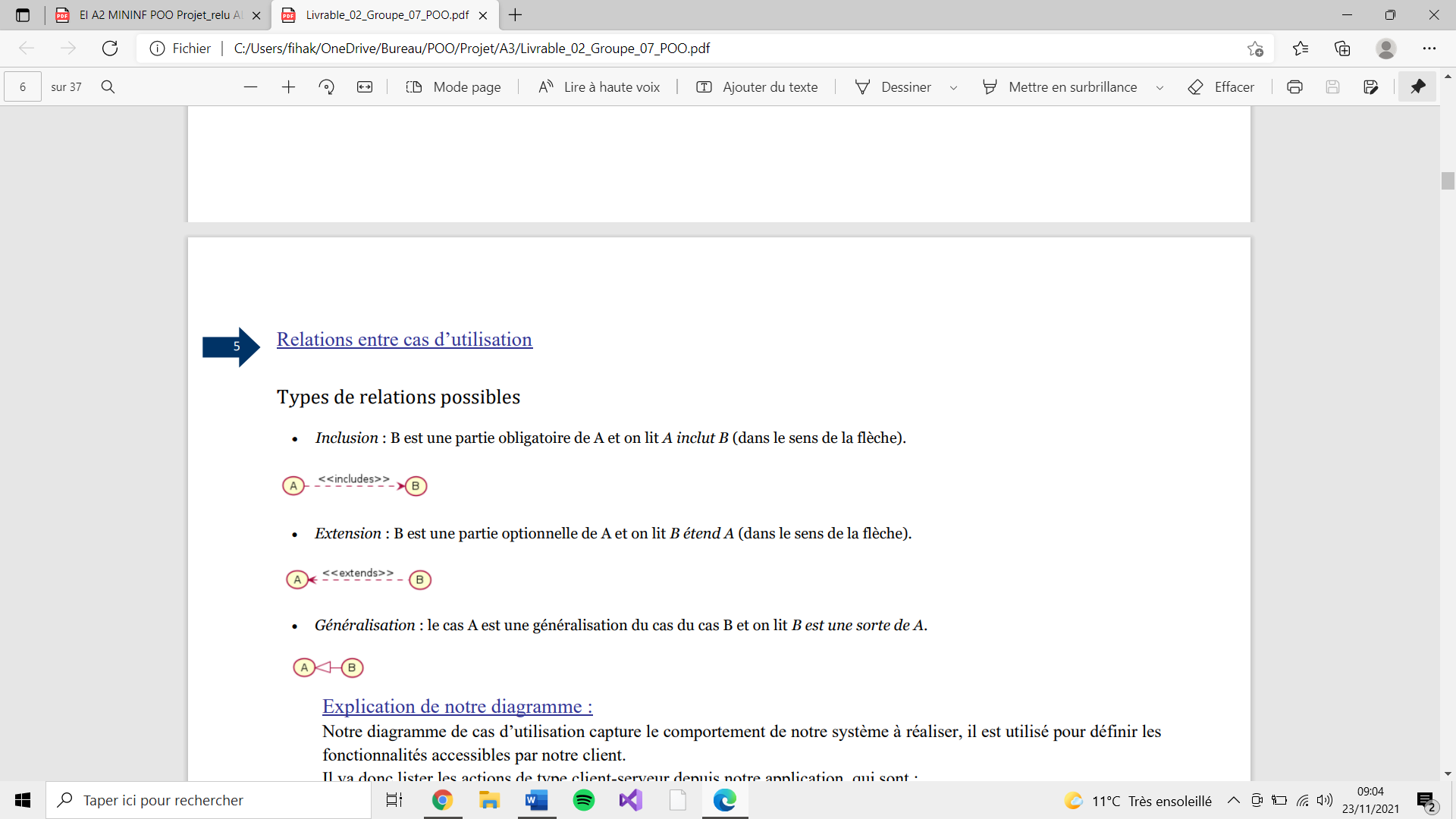
**Diagramme de classes** : représentation des classes intervenant dans le système.

Diagramme de cas d’utilisation :

Les diagrammes de cas d'utilisation (DCU) 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. En effet, un cas d'utilisation (use cases) représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Ainsi, dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), et ils apparaissent dans les cas d'utilisation.

*Types de relations possibles :*

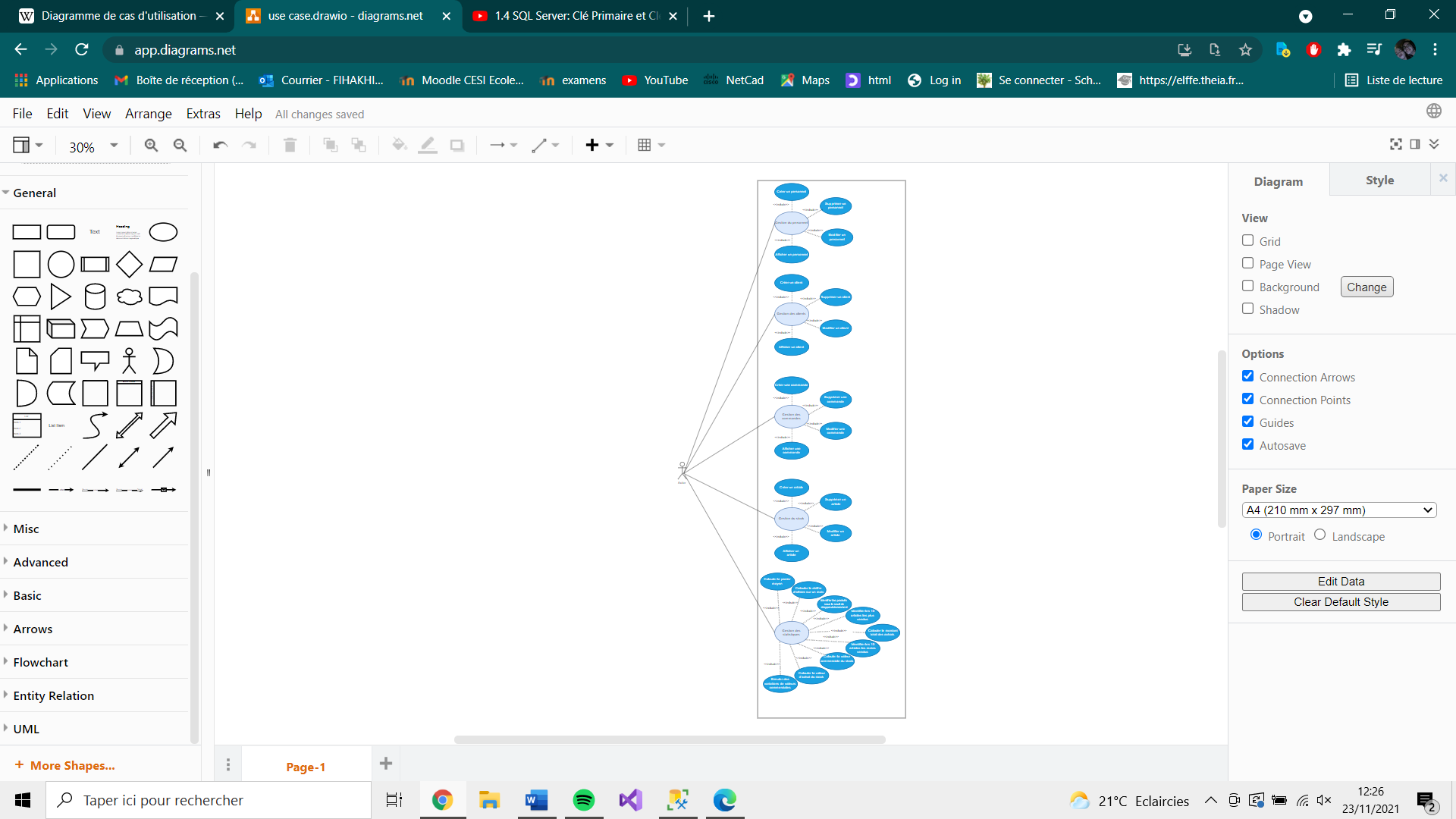
Inclusion : B est une partie obligatoire de A et on lit A inclut B (dans le sens de la flèche).



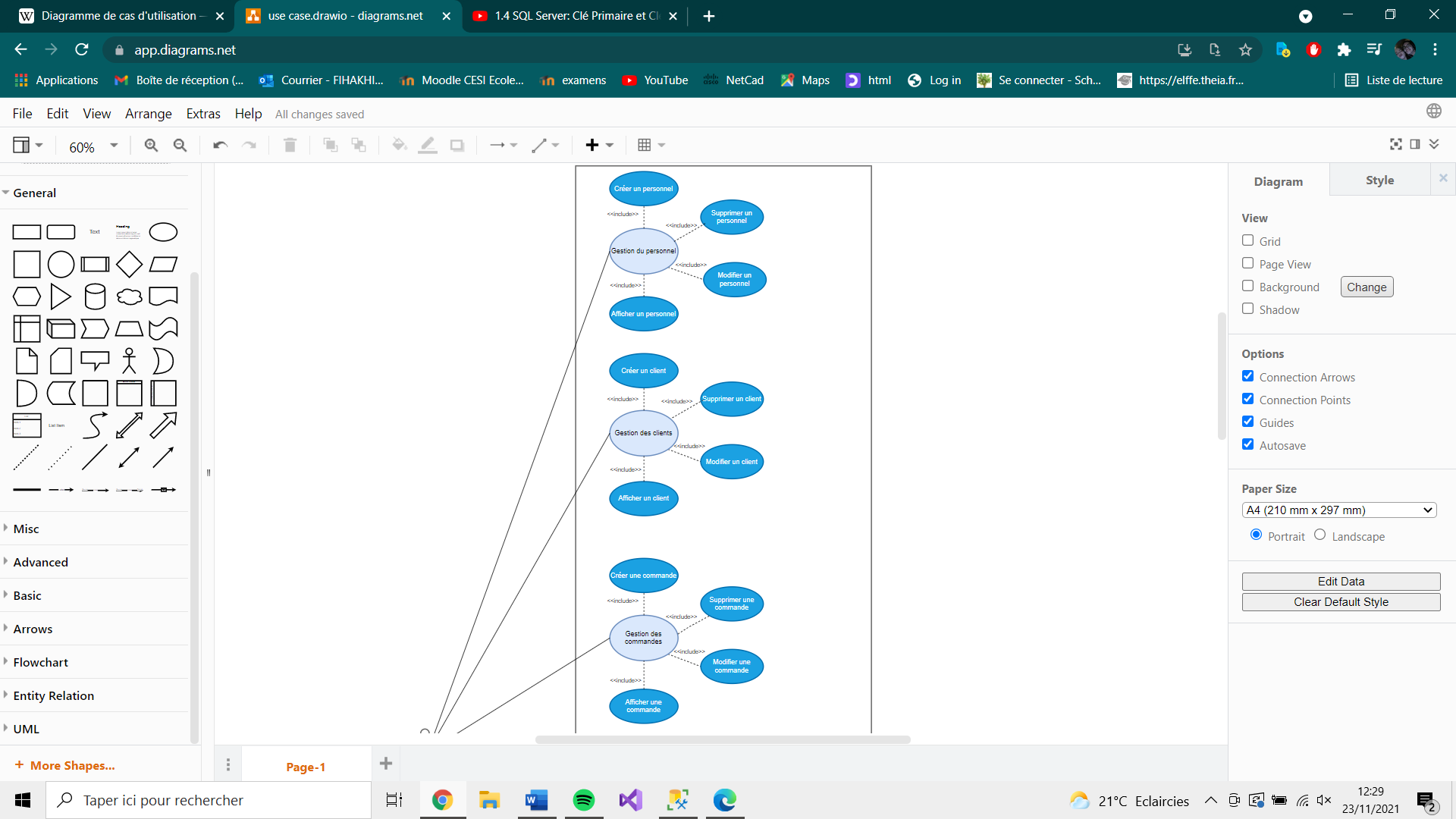
Extension : B est une partie optionnelle de A et on lit B étend A (dans le sens de la flèche).Une image contenant texte

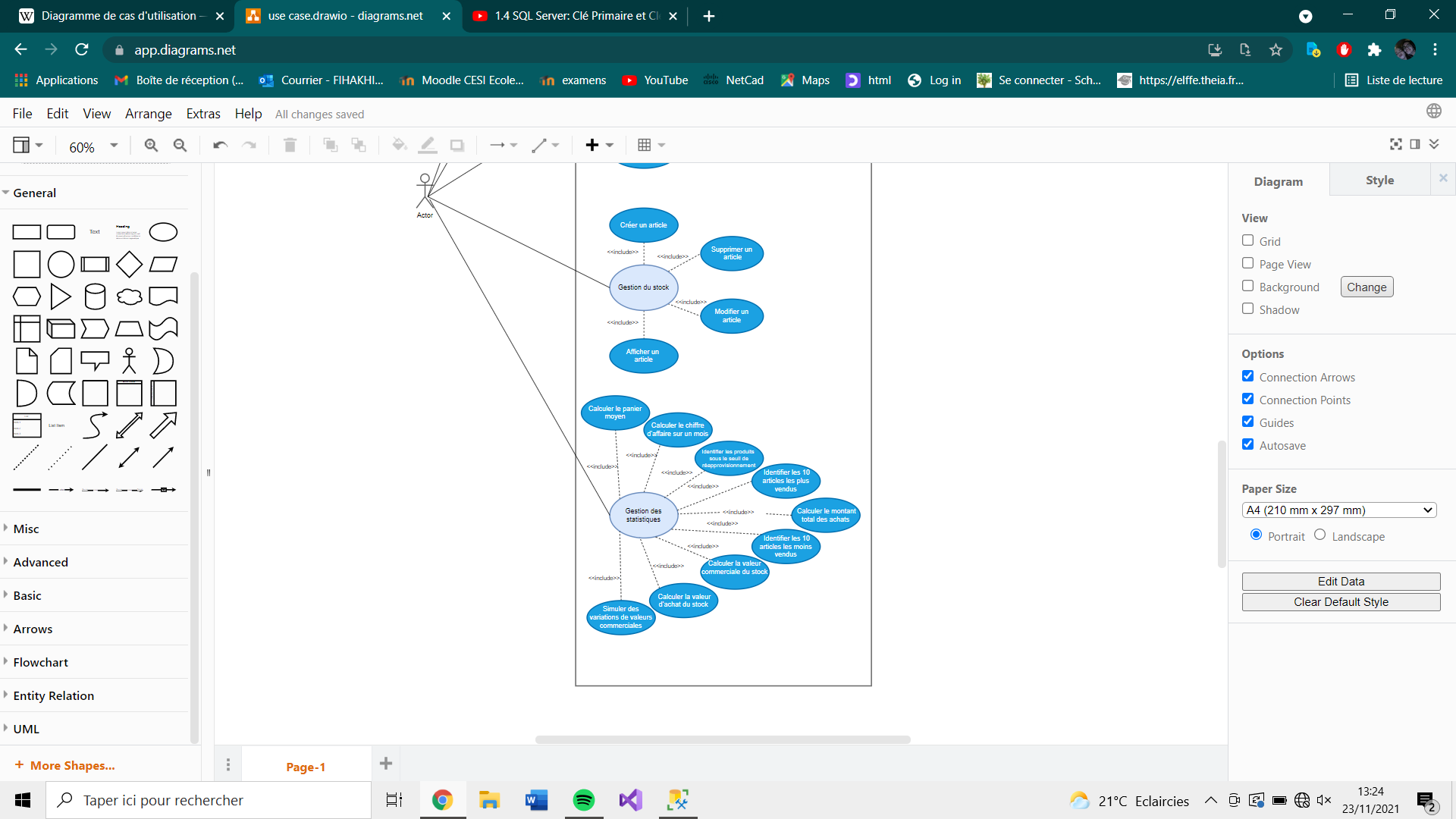
Description générée automatiquement

*Réalisation :*



*En zoomant :*





*Explication :*

Notre diagramme de cas d’utilisation capture le comportement de notre système à réaliser, il est utilisé pour définir les fonctionnalités accessibles par notre client.

Il va donc lister les actions de type client-serveur depuis notre application, qui sont :

• La gestion du personnel ;

• La gestion des clients ;

• La gestion des commandes ;

• La gestion du stock ;

• La gestion des statistiques.

L’acteur, le client (entreprise) pourra donc créer, modifier, supprimer et afficher chacun des clients, du personnel, des commandes ainsi que le stock. Ce dernier est capable de gérer les statistiques, facturer les commandes avec les informations client-serveur nécessaires, le tout grâce à l’incorporation d’une base de données.

Dans cette base de données seront donc stockés les informations client, article et commande.

Diagramme de séquence :

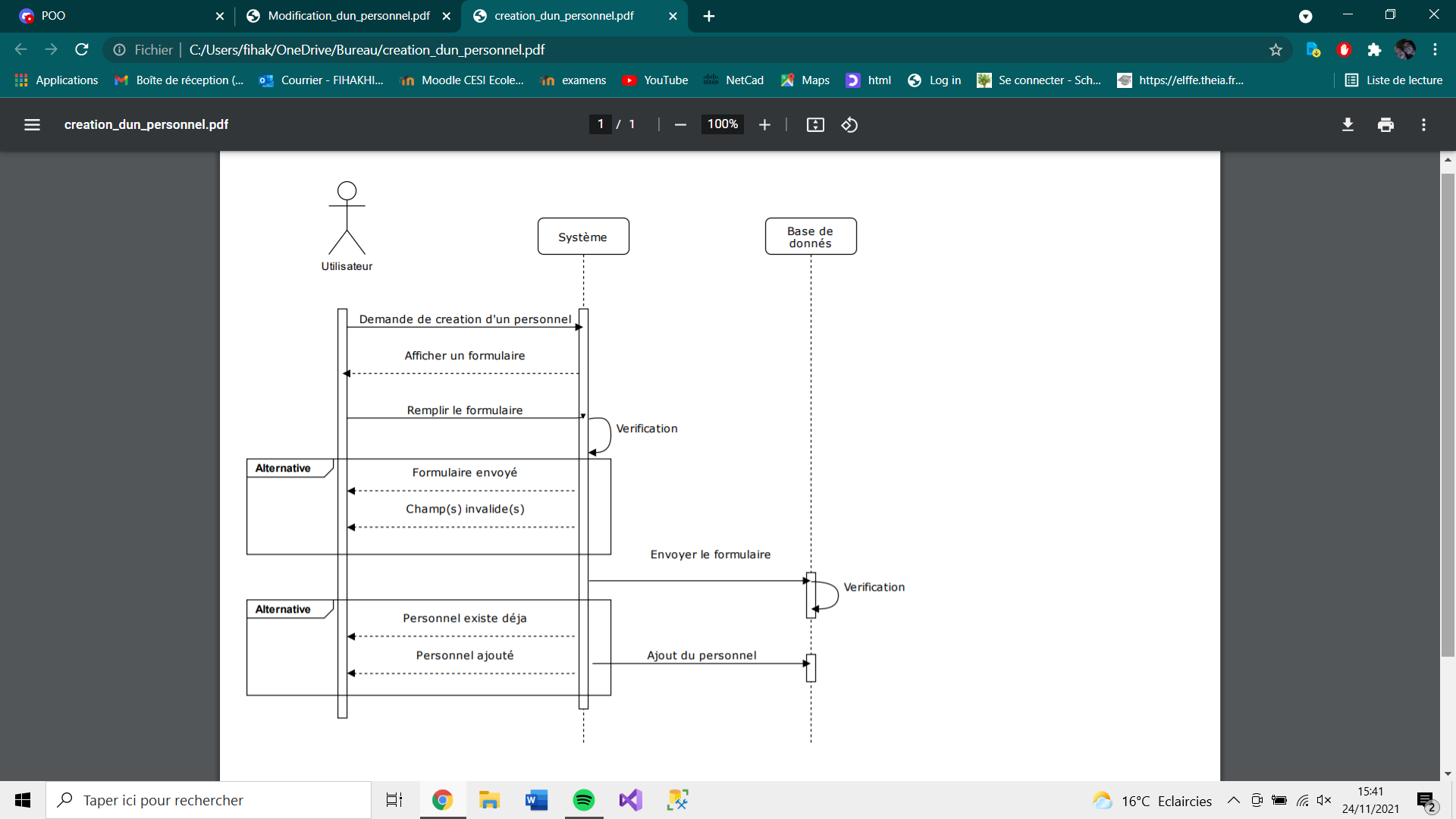
Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but est de décrire comment se déroulent les interactions entre les acteurs ou objets.

Nous allons réaliser quatre diagrammes afin de modéliser commet :

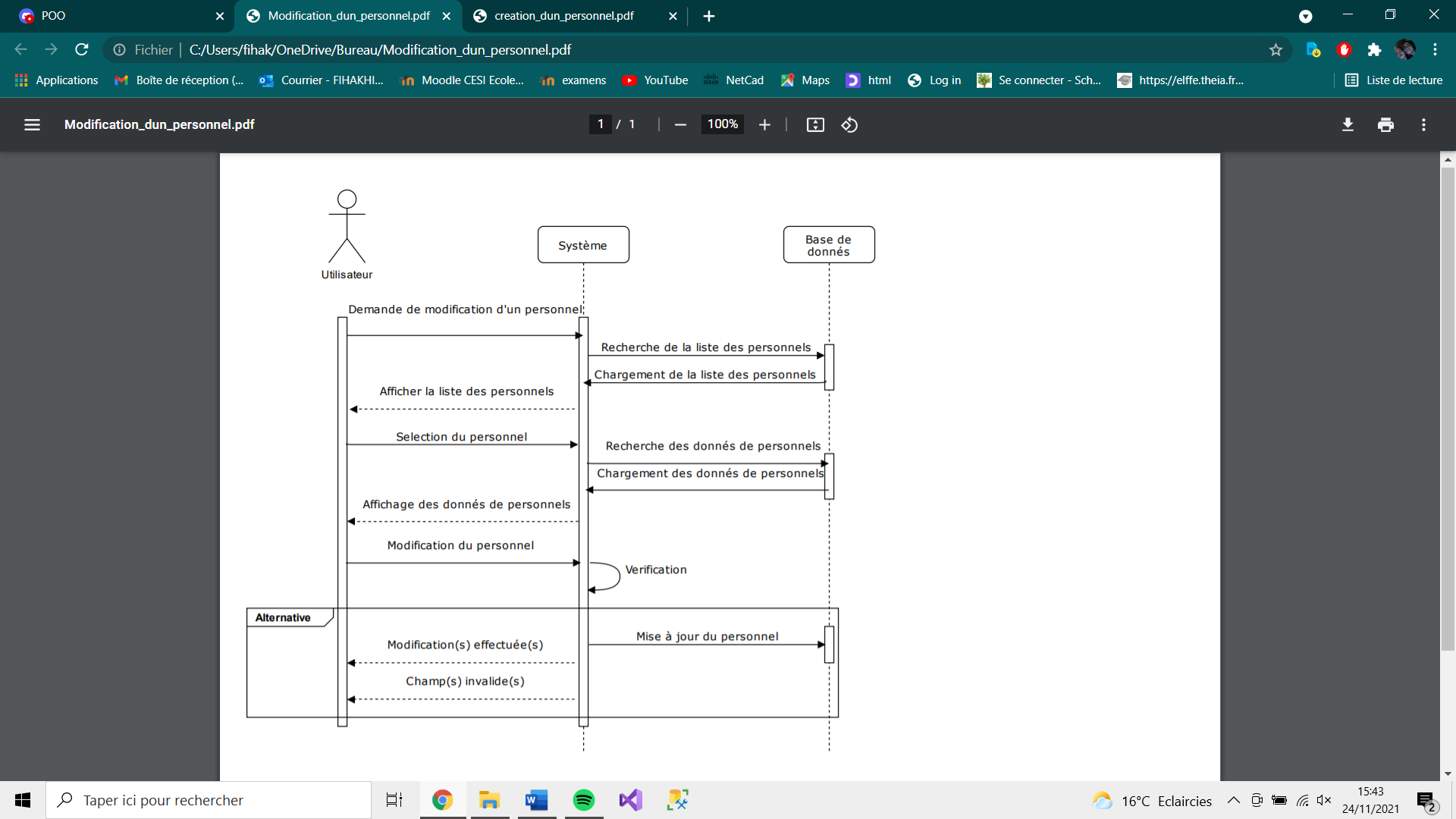
* Créer un personnel ;
* Modifier un personnel ;
* Supprimer un personnel ;
* Afficher un personnel.

*Réalisations :*Création du personnel :



*Explication :*

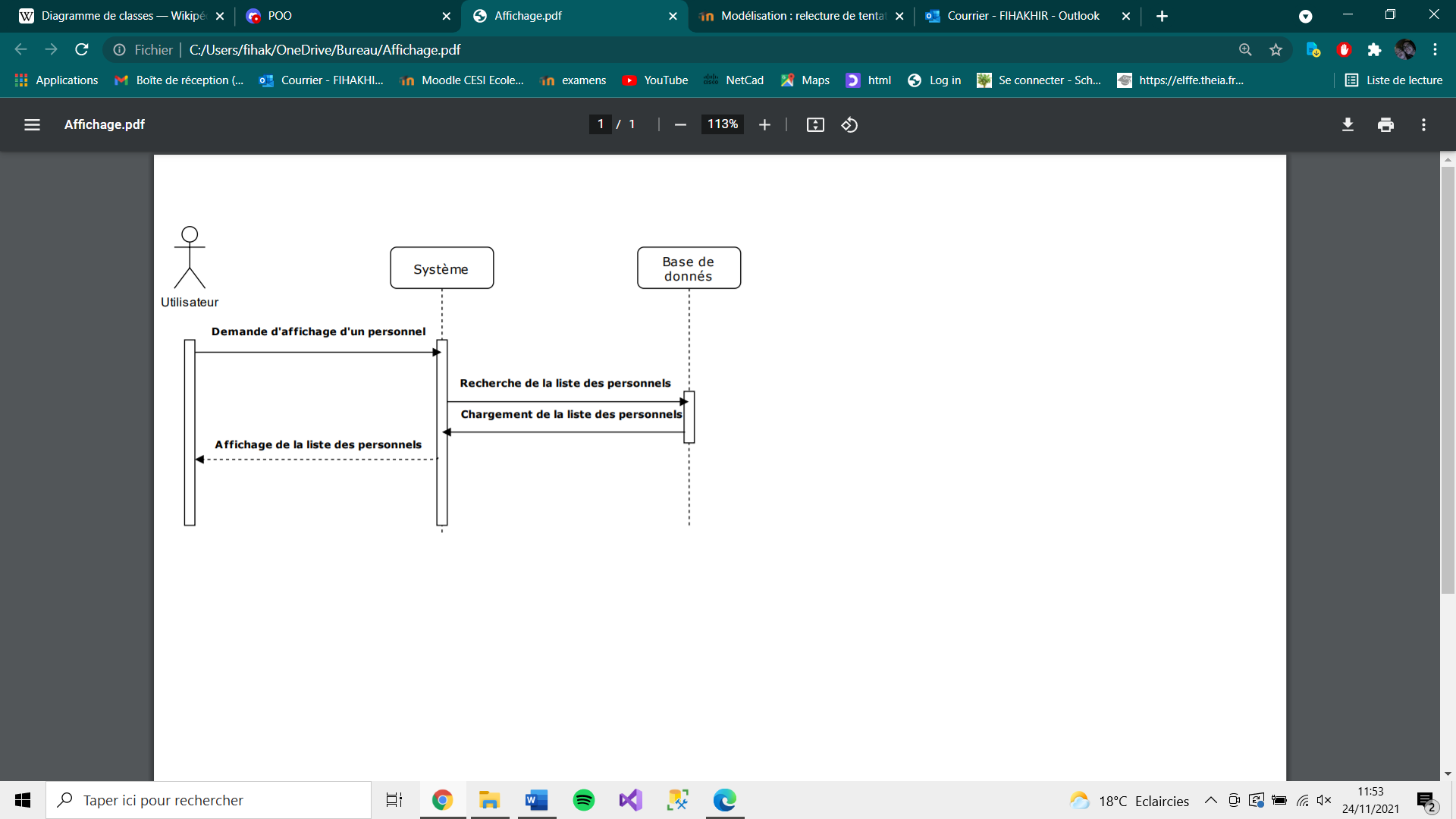
Afin d’ajouter un personnel, l’utilisateur rempli le formulaire en saisissant les caractéristiques (nom, prénom…) depuis l’interface graphique. Puis, le système va chercher dans la base de données, si le personnel existe, il affichera « le personnel existe », sinon il sera ajouté.

Modifications des données du personnel :

*Explication :*

L’utilisateur peut modifier un personnel en sélectionnant le personnel à modifier puis modifie la/les cases à modifier.

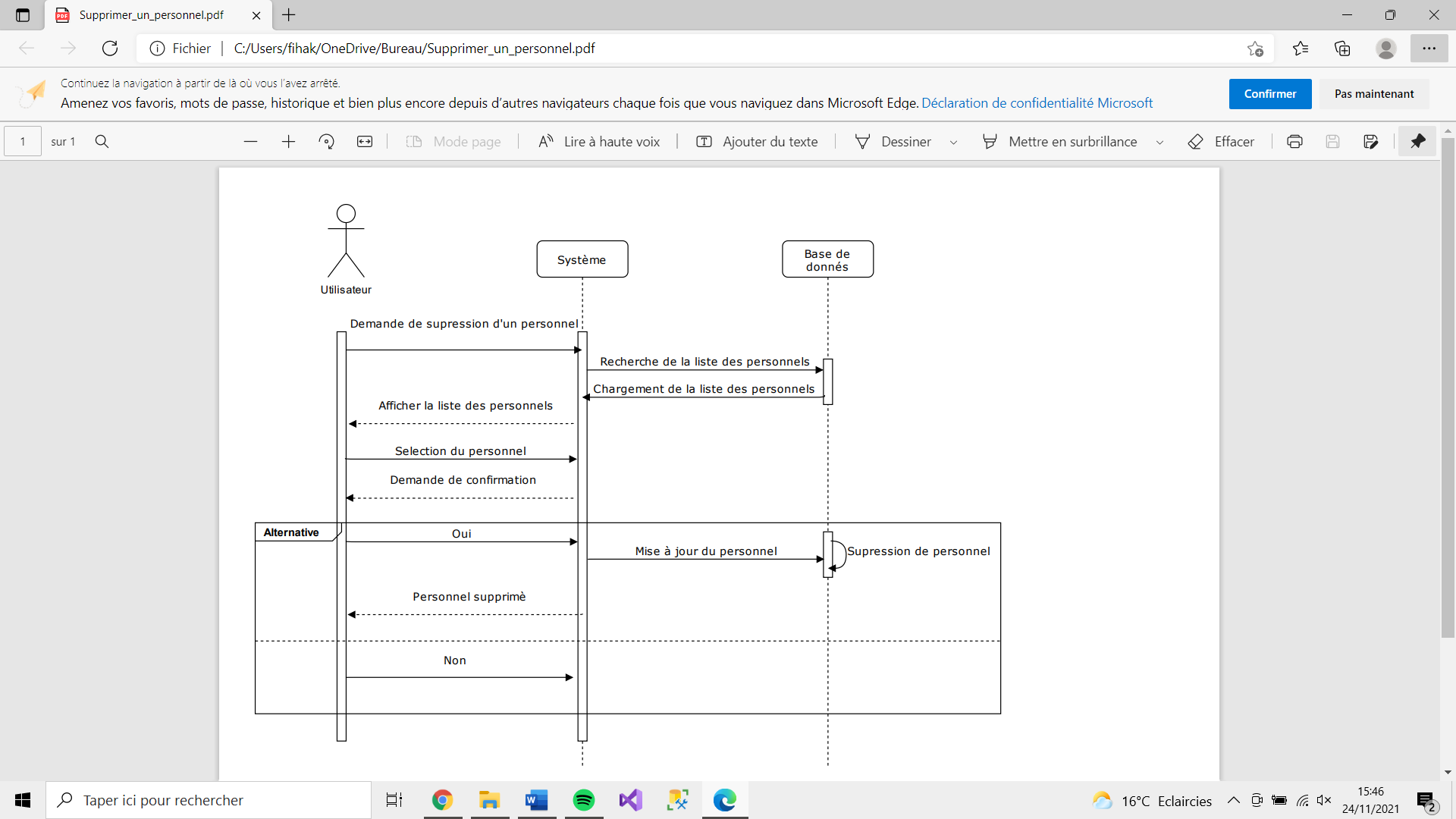
Affichage du personnel :



*Explication :*

Afin d’afficher un personnel, l’utilisateur demande d’afficher en saisissant les caractéristiques (nom, prénom…) depuis l’interface graphique. Puis, le système va chercher dans la base de données et l’afficher sur l’afficheur.

Suppression du personnel :



*Explication :*

Afin d’afficher un personnel, l’utilisateur demande d’afficher en saisissant les caractéristiques (nom, prénom…) depuis l’interface graphique. Puis, le système va chercher dans la base de données et l’afficher sur l’afficheur.

*Explication :*

Afin de supprimer un personnel, l’interface affiche la liste des personnels. Puis, l’utilisateur sélectionne le personnel à supprimer et le système le supprime.

*Explication :*

Il représente l’interaction acteur/système selon un ordre chronologique :

Dans chaque diagramme nous avons expliqué l’ordre chronologique dans lequel se passe modification/création/suppression/affichage, si nous prenons la création comme exemple :

Il est d’abord demandé de créer un nouveau personnel, le système consulte notre classe P : Personnel puis nous affiche

d’entrer les informations du personnel en question (nom, prénom, adresse et date d’embauche), après les avoir entrés un nouveau personnel se créera et s’enregistrera dans notre base de données.

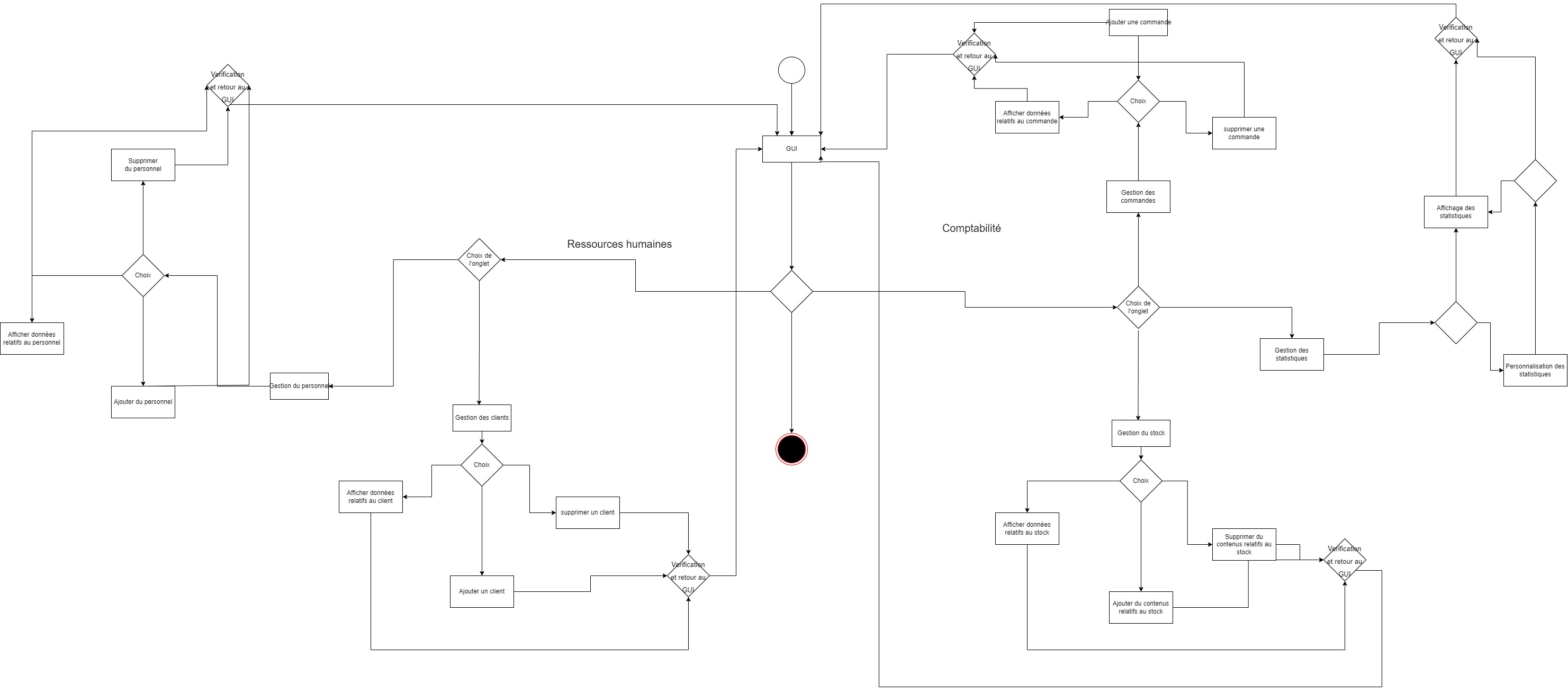
Diagramme d’activité :

Le diagramme d'activité est sémantiquement proche des diagrammes de communication, ou d'état-transitions, ces derniers offrant une vision microscopique des objets du système.

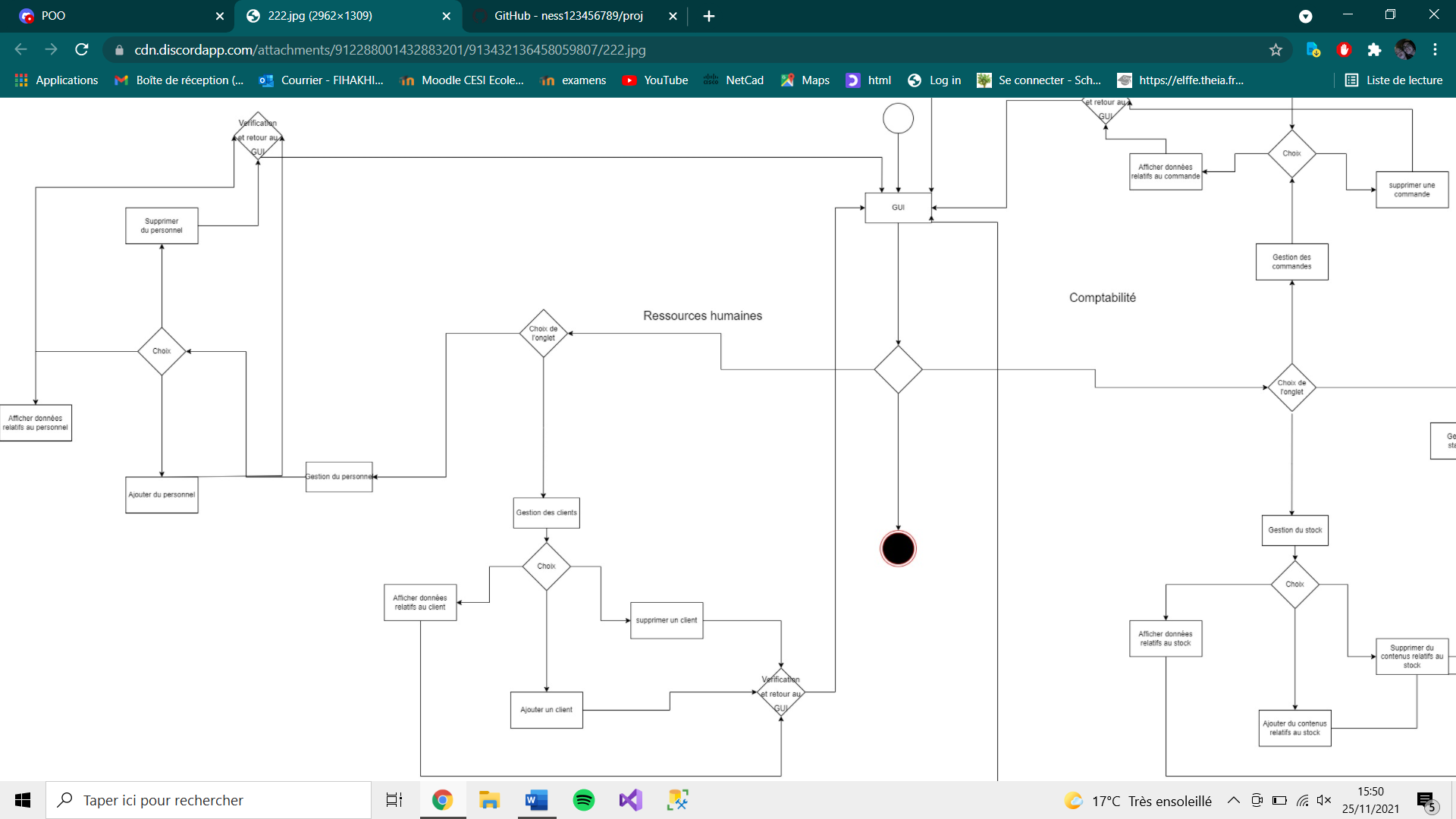
Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information).

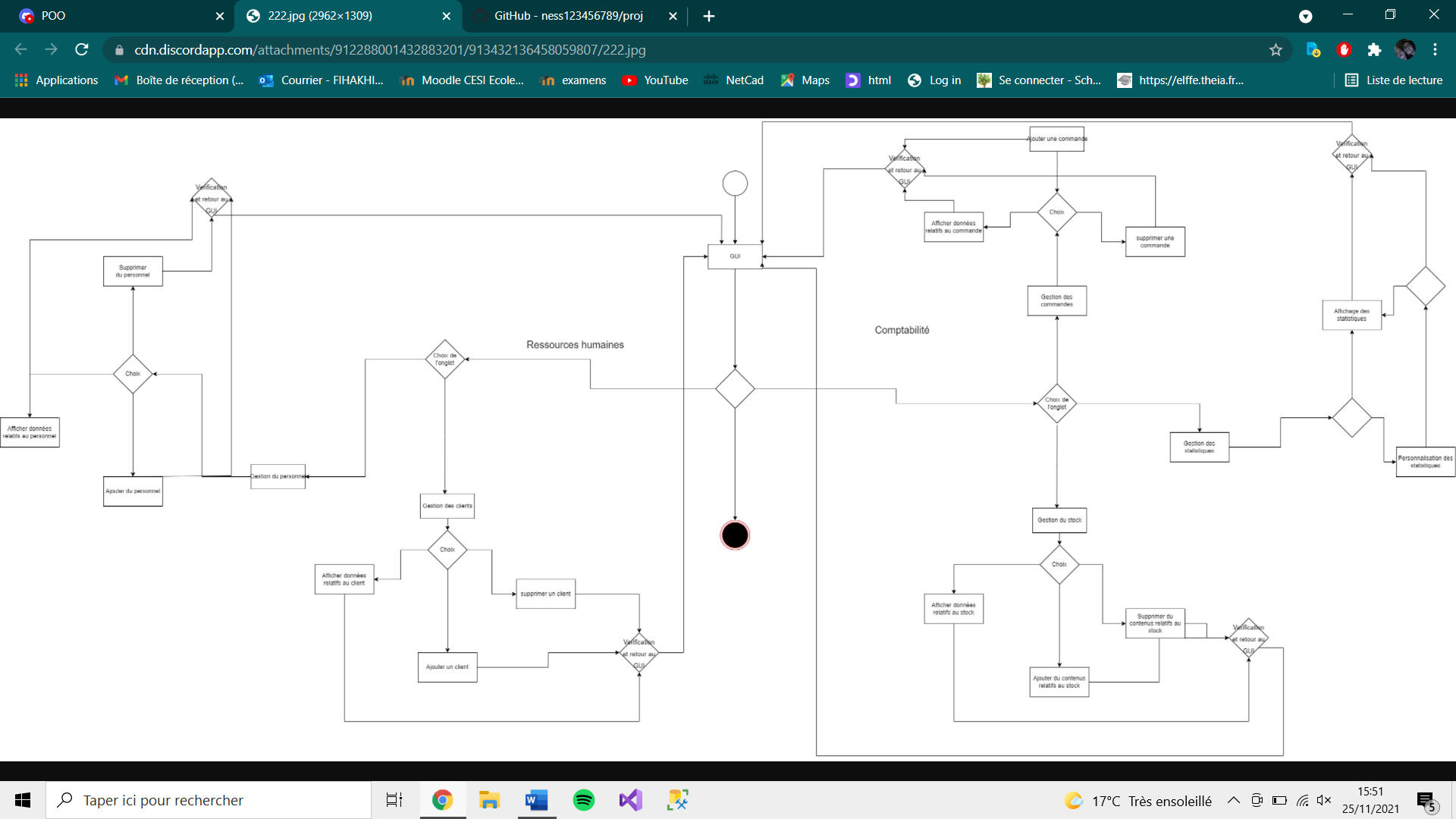
C’est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

*Représentation :*

**

*En zoomant :*





*Explication :*

Nous avons choisi de faire ce dernier car il va nous aider à représenter la gestion du personnel en détail et l’exprimer d’une manière temporelle.

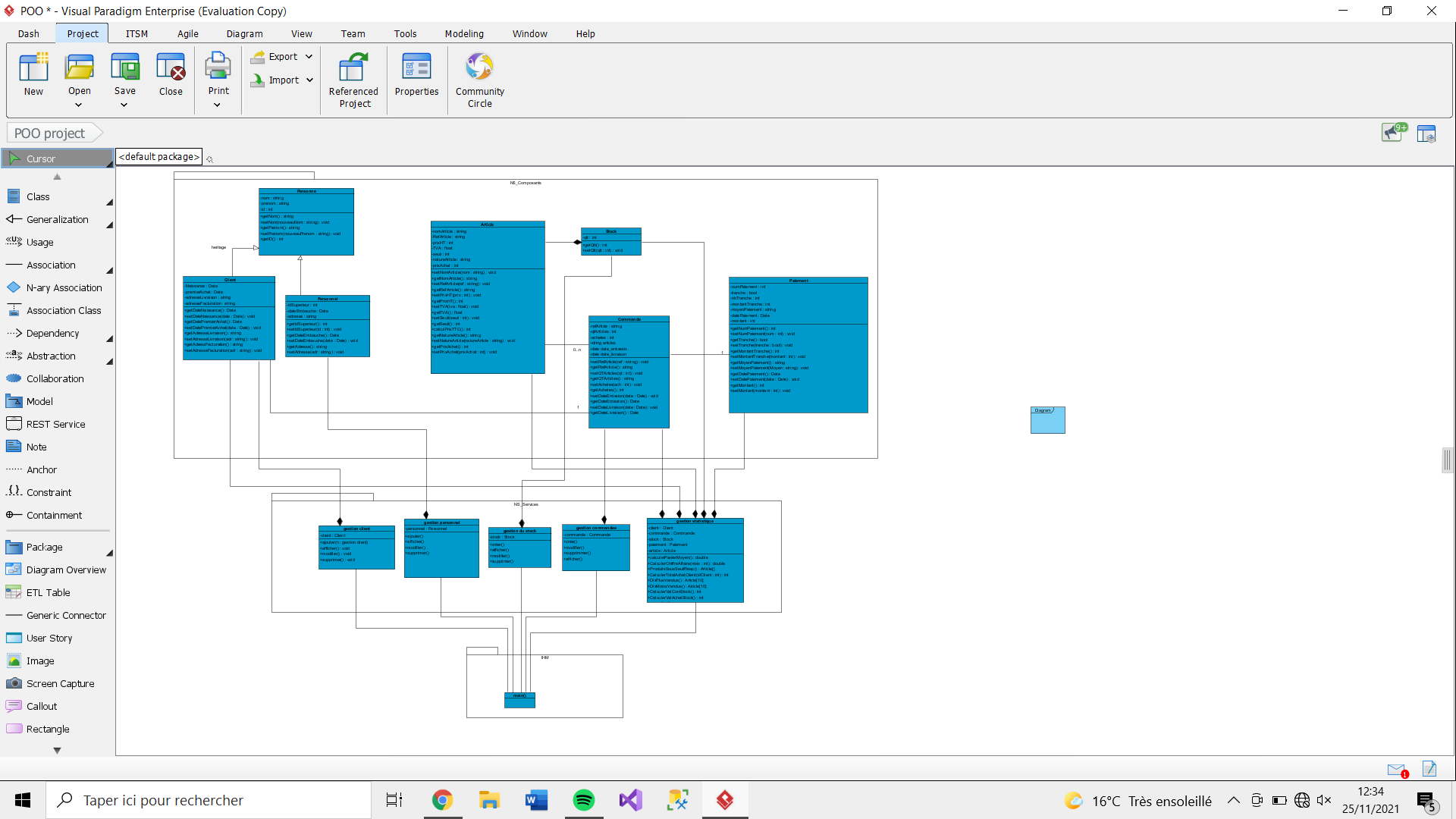
En analysant le diagramme nous pouvons remarquer que à la suite de la saisie du nom et du prénom une vérification aura lieu, si le compte existe déjà il affichera les informations de la personne, puis il y’aura le choix entre soit supprimer les

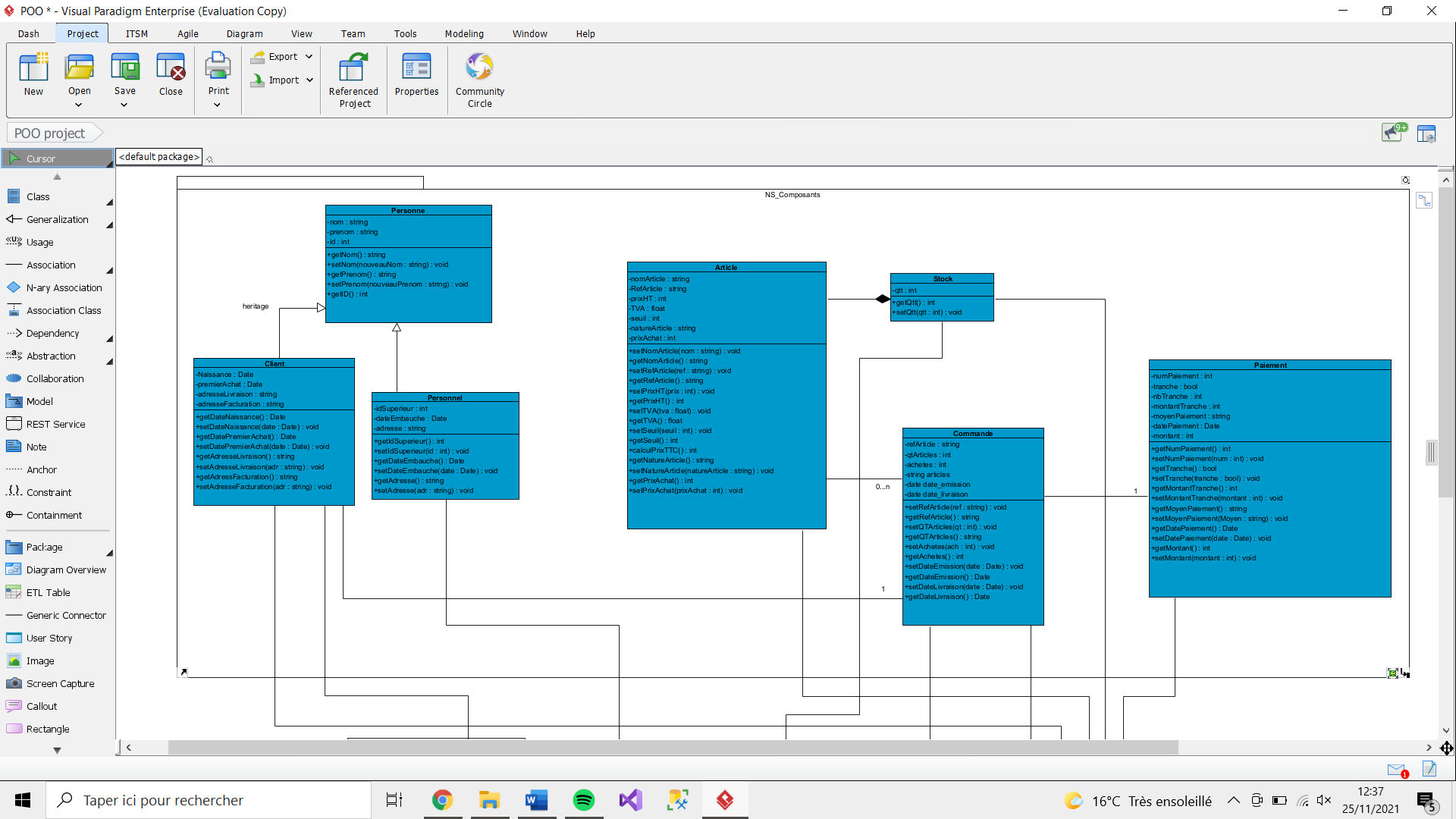
informations ou modifier ces dernières. Si dans le cas contraire la personne n’existe pas, un nouveau compte sera créé où le nom/prénom/adresse/date d’embauche de la personne seront demandés et puis tout en respectant le déroulement temporel les informations à la fin devront toutes être enregistrées, l’état final correspond à la fin de la consultation.

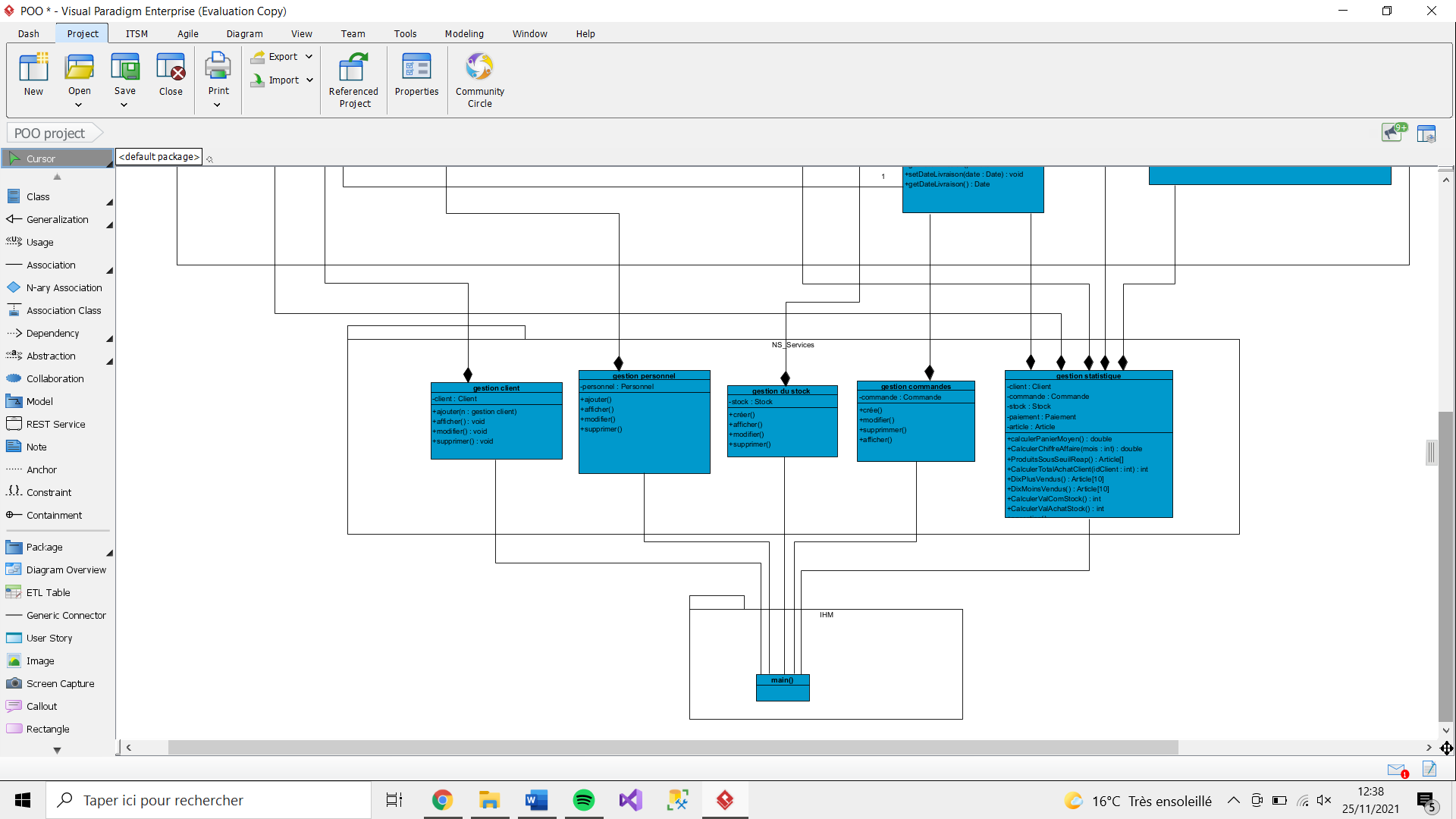
Diagramme de classes :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

*Représentation :*

*En zoomant :*



*Explication :*

Dans ce diagramme, nous avons modéliser les composants ainsi que les différentes couches qui constituent le système : les composants, les services et le « main ».

1. **Environnement de développement opérationnel :**

Une image contenant texte

Description générée automatiquement*Les logiciels utilisés :*

Visual Studio : Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles.

Nous allons coder en C++ pour créer l’interface graphique.

Une image contenant texte

Description générée automatiquement

Jmerise : est un logiciel dédié à la modélisation des modèles conceptuels de données (MCD) pour Merise il permet la généralisation et la spécialisation des entités, la création des relations et des cardinalités ainsi que la généralisation des modéles logiques de données(MLD) et des script SQL.



SQL Server : Microsoft SQL Server est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft.



Draw.io : est un logiciel pour modéliser les différents diagrammes.

Environnement GitHub :

Afin de bien communiquer entre nous, surtout pour la partie code quand chacun d’entre nous va ajouter sa partie, pour un travail simultané.

Voilà le lien :

<https://github.com/ness123456789/proj>

1. **Conclusion :**

Nous avons présenté dans ce livrable les différents diagrammes de notre projet.

Ces diagrammes nous apporterons de l’aide pour modéliser notre interface graphique ainsi que la base de données.