



# **1      Portfolio**

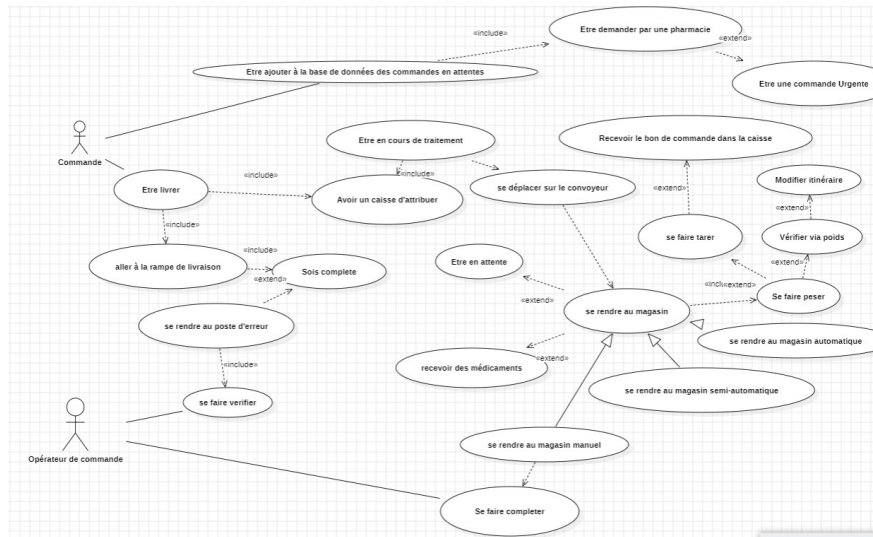
## 1.1 Compétence 1 : Réaliser

**Apprentissage critique** : Élaborer et implémenter les spécifications fonctionnelles et non fonctionnelles à partir des exigences

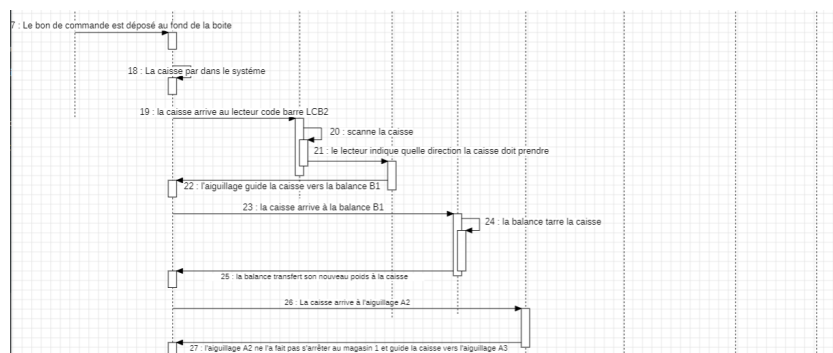
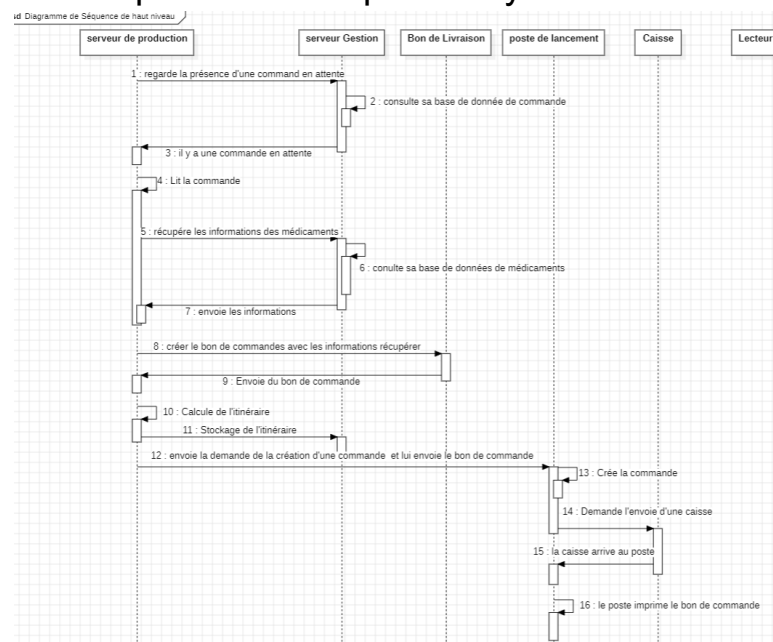
Dans le cadre du module R3.03 (UML), nous avons réalisé un projet en groupe de 4 étudiants, réparti sur plusieurs séances. L'objectif de ce projet était de comprendre et de modéliser le fonctionnement d'un atelier de conditionnement de médicaments utilisé par des sociétés de logistique pharmaceutique, à l'aide de diagrammes UML.

À partir d'un cahier des charges détaillé, nous avons dû produire plusieurs diagrammes pour représenter les différentes facettes du système. En tout, cinq diagrammes ont été réalisés : un diagramme de cas d'utilisation, un diagramme de classes, un diagramme de déploiement, un diagramme de séquence de haut niveau, et un diagramme de séquence de bas niveau. Chaque diagramme présentait des difficultés spécifiques à surmonter, en lien avec l'organisation industrielle décrite.

Le diagramme de cas d'utilisation ci-dessous illustre les différentes interactions entre les acteurs (commande, opérateur) et le système. Il met en évidence les cas d'utilisation principaux comme la livraison, la gestion des erreurs ou encore la manipulation des commandes urgentes. Des relations telles que <<include>> et <<extend>> ont été utilisées pour refléter les comportements conditionnels ou obligatoires.



Nous avons également réalisé un diagramme de séquence de haut niveau, qui illustre le traitement automatisé d'une commande depuis sa détection par le serveur de production jusqu'à l'envoi de la caisse dans le système logistique. Ce diagramme montre l'enchaînement des interactions entre les composants logiciels et met en évidence l'automatisation des premières étapes du cycle de conditionnement.



Mais au-delà de la création de diagrammes UML, il y a aussi l'élaboration de cahiers des charges, comme lors de mon stage où j'ai travaillé à partir d'une base existante que j'ai dû compléter en y ajoutant de nouvelles fonctionnalités.

Voici un exemple :

#### 1. Introduction

- **Objectif du projet** : Développer un intranet pour améliorer la communication interne et la gestion des documents au sein de l'entreprise.
- **Contexte** : Présentation de la collectivité (mise à jour possible par Sce RH).

#### 2. Fonctionnalités Requises

- **Trombinoscope** :
  - Affichage des photos et des informations de contact des agents (mise à jour responsable services et RH).
  - Recherche par nom, service, ou poste.
- **Dépôt de Documents par Service** :
  - Système de gestion des documents avec accès par service.
  - Possibilité de télécharger, consulter les documents.

#### 3. Exigences Techniques

- **Plateforme** : Technologies web (HTML, CSS, JavaScript, etc.).
- **Sécurité** : Authentification et autorisation des utilisateurs, chiffrement des données. Sécurisé. (Active directory )
- **Compatibilité** : Fonctionnement sur différents navigateurs et appareils.

En effet, comme on peut le voir, le cahier des charges était partiellement complet. J'ai donc dû y ajouter certaines fonctionnalités qui n'avaient pas été évoquées initialement, afin d'améliorer l'accessibilité et l'expérience utilisateur.

## 1.2 Compétence 2 : Optimiser

**Apprentissage critique** : Comprendre les enjeux et moyens de sécurisation des données et du code .

Lors du semestre 3, nous avons suivi des cours portant sur la sécurité informatique et la cryptographie. Ces enseignements nous ont véritablement permis de comprendre l'importance de la sécurité, notamment au niveau du code, et d'explorer les principaux concepts liés au chiffrement.

Nous avons étudié plusieurs méthodes de chiffrement, telles que le chiffrement de Hill, le chiffrement symétrique et asymétrique ou encore le chiffrement affine avec attaque, chacune reposant sur des mécanismes et des niveaux de sécurité spécifiques.

On peut observer dans le code ci-dessous une utilisation du chiffrement symétrique combinée à un déchiffrement, le tout reposant sur l'algorithme RSA :

```
def rsa_encrypt_decrypt():
    p = 53
    q = 11
    e = 3
    n = p * q
    phi = (p - 1) * (q - 1)
    print(f"p = {p}, q = {q}, e = {e}")
    print(f"n = p * q = {n}")
    print(f"φ(n) = (p-1) * (q-1) = {phi}")

    def modinv(a, m):
        g, x, y = extended_gcd(a, m)
        print(f"Calcul de l'inverse modulaire de {a} mod {m}:")
        if g != 1:
            print(f"Pas d'inverse, car le PGCD de {a} et {m} est {g}")
            return None
        else:
            print(f"L'inverse modulaire est {x % m}")
            return x % m

    def extended_gcd(a, b):
        if a == 0:
            return (b, 0, 1)
        else:
            g, y, x = extended_gcd(b % a, a)
            return (g, x - (b // a) * y, y)

    d = modinv(e, phi)
    print(f"\nClé publique (n, e): ({n}, {e})")
    print(f"Clé privée d: {d}")
    message = "JEVOUSAIME"
    print(f"\nMessage original: {message}")

    def text_to_numbers(text):
        numbers = [str(ord(c) - ord('A') + 1).zfill(2) for c in text if c.isalpha()]
        numbers_str = '0' + ''.join(numbers)
        return numbers_str
    numbers_str = text_to_numbers(message)
    print(f"Message en nombres avec 0 ajouté: {numbers_str}")

    def split_blocks(numbers_str, block_size=3):
        return [numbers_str[i:i+block_size] for i in range(0, len(numbers_str), block_size)]
    blocks = split_blocks(numbers_str)
    print(f"Blocs avant chiffrement: {' '.join(blocks)}")
    for block in blocks:
        if int(block) >= n:
            print(f"Attention: bloc {block} >= n ({n})")
    def encrypt_block(m, e, n):
        encrypted = pow(int(m), e, n)
        print(f"Chiffrement de {m} : {m}^{e} mod {n} = {encrypted}")
        return encrypted

    encrypted_blocks = [str(encrypt_block(block, e, n)) for block in blocks]
    print(f"Blocs chiffrés: {' '.join(encrypted_blocks)}")

    def decrypt_block(c, d, n):
        decrypted = str(pow(int(c), d, n).zfill(3))
        print(f"Déchiffrement de {c} : {c}^{d} mod {n} = {decrypted}")
        return decrypted
    decrypted_blocks = [decrypt_block(block, d, n) for block in encrypted_blocks]
    print(f"Blocs déchiffrés: {' '.join(decrypted_blocks)}")

    def numbers_to_text(blocks):
        full_number = ''.join(blocks)[1:]
        text = ''
        for i in range(0, len(full_number), 2):
            code = int(full_number[i:i+2])
            if 1 <= code <= 26:
                text += chr(code + ord('A') - 1)
        return text
    decrypted_message = numbers_to_text(decrypted_blocks)
    print(f"Message déchiffré: {decrypted_message}")
    sa_encrypt_decrypt()
```

Grâce à ces fonctions, il est possible de transformer un texte clair en texte chiffré, puis de le déchiffrer à volonté, à condition de connaître les clés utilisées.

```
Message original: JEVOUSAIME
Message en nombres avec 0 ajouté: 010052215211901091305
Blocs avant chiffrement: 010 052 215 211 901 091 305
Attention: bloc 901 >= n (583)
Chiffrement de 010 : 010^3 mod 583 = 417
Chiffrement de 052 : 052^3 mod 583 = 105
Chiffrement de 215 : 215^3 mod 583 = 557
Chiffrement de 211 : 211^3 mod 583 = 52
Chiffrement de 901 : 901^3 mod 583 = 318
Chiffrement de 091 : 091^3 mod 583 = 335
Chiffrement de 305 : 305^3 mod 583 = 347
Blocs chiffrés: 417 105 557 52 318 335 347
```

```
Déchiffrement de 417 : 417^347 mod 583 = 010
Déchiffrement de 105 : 105^347 mod 583 = 052
Déchiffrement de 557 : 557^347 mod 583 = 215
Déchiffrement de 52 : 52^347 mod 583 = 211
Déchiffrement de 318 : 318^347 mod 583 = 318
Déchiffrement de 335 : 335^347 mod 583 = 091
Déchiffrement de 347 : 347^347 mod 583 = 305
Blocs déchiffrés: 010 052 215 211 318 091 305
Message déchiffré: JEVOURIME
```

En effet, on peut constater que le texte n'est pas exactement le même. Comme nous avons un chiffre supérieur à 583, le déchiffrement ne s'est pas effectué correctement, ce qui a renvoyé d'autres caractères.

Le chiffrement RSA offre l'avantage d'une sécurité renforcée grâce à l'utilisation d'une paire de clés (publique et privée), ce qui permet un échange sécurisé des données sans nécessiter de clé secrète partagée à l'avance.

Toutefois, un inconvénient majeur réside dans la gestion des clés, notamment la nécessité de protéger rigoureusement la clé privée. Si celle-ci est perdue ou compromise, il devient impossible de déchiffrer les messages, ou pire, un tiers peut accéder à des informations sensibles.



## 1.3 Compétence 3 : Administrer

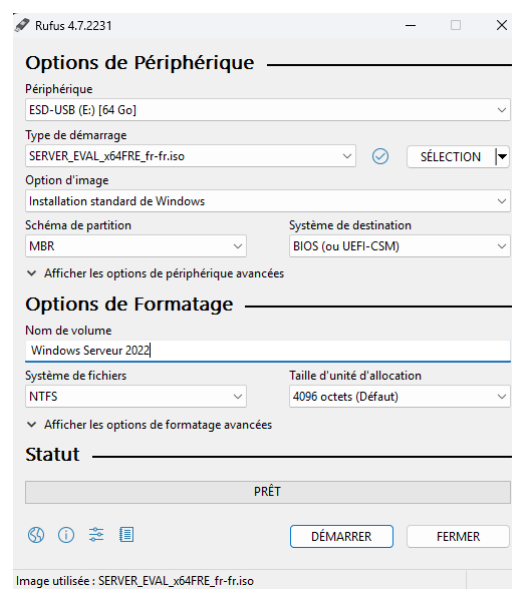
**Apprentissage critique** : Utiliser des serveurs et des services réseaux virtualisés

Lors de mon stage, j'ai eu la chance de pouvoir configurer un serveur entièrement à partir de zéro. Un collègue m'a généreusement montré les différentes étapes nécessaires à cette configuration, ce qui m'a permis d'enrichir mon portfolio.

La mise en place d'un serveur suit plusieurs étapes précises qu'il est essentiel de respecter pour garantir son bon fonctionnement. Un serveur peut être livré « nu », sans disque dur ni configuration, nécessitant une préparation complète, ou bien pré-configuré avec certains paramètres déjà en place, ce qui facilite grandement la tâche lors de la configuration.

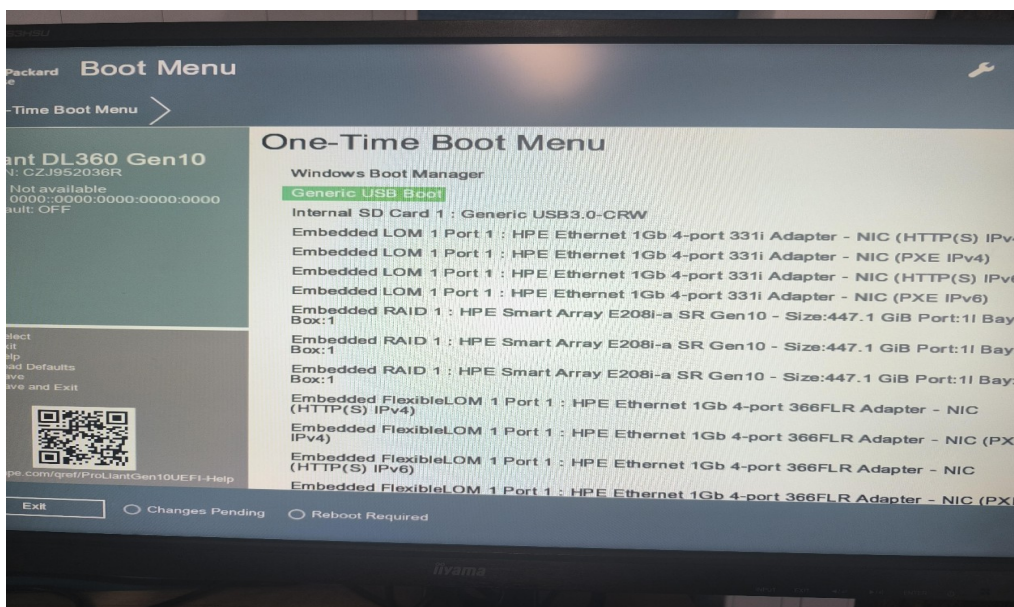
La première étape consiste à créer une clé USB bootable contenant un fichier ISO, nécessaire à l'installation de Windows sur le serveur. Ce fichier, souvent volumineux, peut prendre un certain temps à être téléchargé. Pour le transfert et la préparation de la clé, on utilise Rufus, un outil gratuit et open source pour Windows. Ce logiciel permet de formater facilement une clé USB et de la rendre amorçable, afin de lancer l'installation de manière fiable sur le serveur.

Voici à quoi ressemble l'interface de Rufus :



On constate que l'interface de Rufus est à la fois simple et complète. Il suffit de sélectionner le fichier ISO et de le placer au bon endroit : le logiciel se charge ensuite automatiquement de rendre la clé USB amorçable.

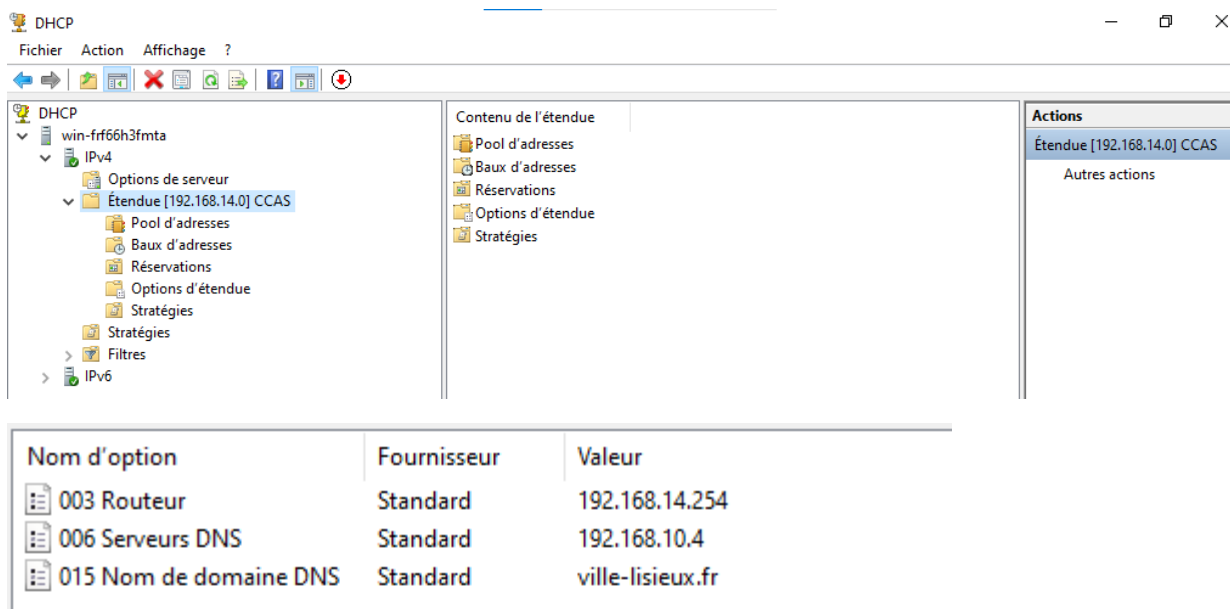
Cependant, cette étape ne représente que le début du processus de configuration du serveur. Une fois la clé insérée dans la machine, on accède au menu de démarrage (boot), comme le montre la capture d'écran suivante.



Mais ce n'est pas la seule étape. En effet, pour qu'un serveur fonctionne correctement, il est indispensable de configurer et

d'installer le service DHCP, qui permet d'attribuer automatiquement des adresses IP aux appareils du réseau.

Lors de cette configuration, il faut définir un nom pour le serveur, lui attribuer une adresse IP fixe, puis exclure une plage d'adresses IP qui ne doivent pas être utilisées (car elles sont déjà attribuées ou réservées). Il est également nécessaire de configurer le serveur DNS en spécifiant son adresse IP, ainsi que celle de la passerelle (routeur). Enfin, il faut renseigner le nom du domaine auquel le serveur est rattaché. Voilà à quoi ressemble le dhcp finit :



Nom d'option	Fournisseur	Valeur
003 Routeur	Standard	192.168.14.254
006 Serveurs DNS	Standard	192.168.10.4
015 Nom de domaine DNS	Standard	ville-lisieux.fr

On constate donc que la mise en place d'un serveur peut sembler rapide, mais elle devient vite complexe si l'on ne maîtrise pas bien les notions liées aux adresses IP ou si l'on ne connaît pas les étapes précises de la configuration.

## 1.4 Compétence 4 : Gérer

**Apprentissage critique** : Assurer la sécurité des données (intégrité et confidentialité)

Dans une entreprise, les informations et les identifiants des personnes sont très importants. Pour les protéger, il existe plusieurs méthodes, comme le hachage des mots de passe. Cela consiste à transformer le mot de passe en une suite de caractères mélangés, souvent avec l'ajout d'un petit élément aléatoire. Ainsi, même si quelqu'un accède à la base de données, il ne pourra pas lire les vrais mots de passe directement.

Cette protection est obligatoire pour les entreprises, car elle fait partie des règles imposées par le RGPD (Règlement Général sur la Protection des Données), une loi européenne qui vise à garantir la sécurité et la confidentialité des données personnelles.

Lors de mon stage, j'ai eu à gérer l'enregistrement des identifiants de connexion de nombreuses personnes. Pour garantir la sécurité de ces données, j'ai utilisé la technique du hachage, qui permet de protéger les mots de passe en les rendant illisibles. Cela évite qu'ils puissent être facilement récupérés en cas d'accès non autorisé à la base de données.

Voici un exemple du hachage d'un mot de passe dans une base de donnée :

Supprimer	95	dupont	jean	NULL	jdupont@ville-lisieux.fr	NULL	NULL	6	\$2y\$10\$utEtkhy6lIAOpIR9zQ7T.oE7QJep10VF4QWZl6NQL...	NULL
Supprimer	101	CAILLET	Enzo		SVC-	[CN=MAICOINF-		22	\$2y\$10\$5voTDGkht7ZqlwzQAyAlvudM3mzFd3dW5sNihWguwg...	Ville - DSI - Stagiaire
					INFORMATIQUE	IMP,OU=Imprimantes,OU=Groupes,OU=Vil...				

Et voila un exemple de ligne de code pour le hashage de mot de passe en php :

```

<?php
// Connexion à la base de données avec PDO
$dns = 'mysql:host=localhost;dbname=nom_de_la_base;charset=utf8mb4';
$utilisateur = 'nom_utilisateur';
$motDePasseDB = 'mot_de_passe';

try {
    $pdo = new PDO($dns, $utilisateur, $motDePasseDB);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Erreur de connexion : " . $e->getMessage());
}

$email = $_POST['email'] ?? '';
$motDePasse = $_POST['mot_de_passe'] ?? '';
$nom = $_POST['nom'] ?? '';

// Hachage du mot de passe
$motDePasseHache = password_hash($motDePasse, PASSWORD_DEFAULT);

$sql = "INSERT INTO utilisateurs (email, mot_de_passe, nom) VALUES (:email, :mot_de_passe, :nom)";
$stmt = $pdo->prepare($sql);

$stmt->bindParam(':email', $email);
$stmt->bindParam(':mot_de_passe', $motDePasseHache);
$stmt->bindParam(':nom', $nom);

// Exécution de la requête
if ($stmt->execute()) {
    echo "Utilisateur enregistré avec succès.";
} else {
    echo "Erreur lors de l'enregistrement de l'utilisateur.";
}
?>

```

Le hachage des mots de passe lors de la connexion d'un utilisateur à une base de données constitue une mesure de sécurité indispensable.

Il permet de protéger efficacement les informations d'authentification, même en cas de compromission de la base de données, en empêchant l'accès direct aux mots de passe réels. Cette approche renforce la sécurité globale du système en limitant les risques liés aux intrusions et aux tentatives de piratage.

Par ailleurs, les cours de R3.11 nous ont permis d'approfondir les principes liés au RGPD (Règlement Général sur la Protection des Données). Ils nous ont aidés à mieux comprendre l'importance de la sécurité et de la bonne gestion des mots de passe. Le hachage, par exemple, s'inscrit dans cette démarche en empêchant que les

données personnelles soient stockées en clair, et en les rendant inaccessibles aux personnes non autorisées.

Toutefois, même si un mot de passe haché peut sembler inviolable, il existe des techniques permettant de tenter de le récupérer. Si un attaquant parvient à obtenir un hash, il peut essayer de le déchiffrer à l'aide de méthodes comme l'attaque par force brute ou les tables arc-en-ciel (Rainbow Tables), qui utilisent des bases de données pré-calculées pour retrouver les mots de passe d'origine.

## 1.5 Compétence 5 : Conduire

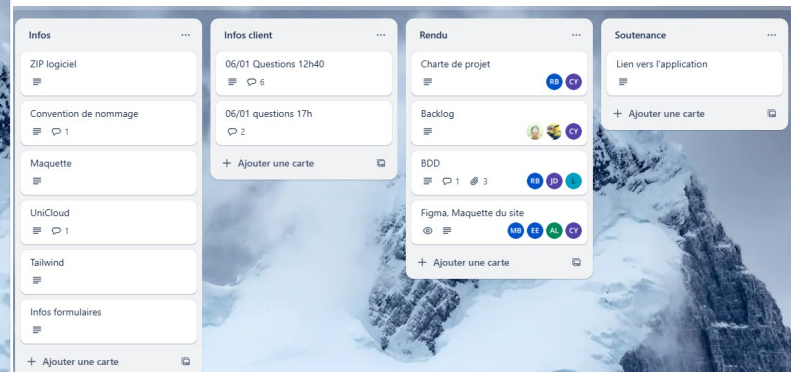
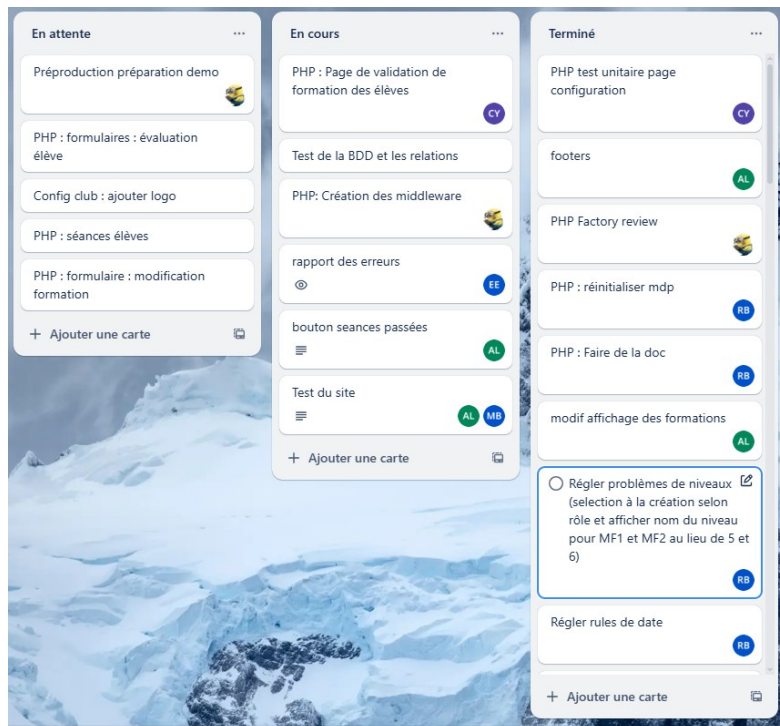
**Apprentissage critique** : Définir et mettre en oeuvre une démarche de suivi de projet

Dans le cadre de la SAE du semestre 3, nous avons été amenés à développer un site web pour un centre de plongée. Ce projet avait pour objectif de répondre à une demande spécifique d'un client, en respectant précisément ses besoins et ses attentes. Il nous était toutefois possible de proposer des alternatives techniques lorsque certaines solutions étaient trop complexes ou peu adaptées à nos compétences de développement à ce stade.

Afin d'assurer une bonne organisation et un déroulement efficace du projet, nous avons utilisé Trello, un outil collaboratif de gestion de projet. Trello repose sur un système de tableaux divisés en listes de cartes, chaque carte représentant une tâche à accomplir. Ces tâches peuvent être attribuées à des membres de l'équipe, accompagnées de descriptions, d'échéances, de pièces jointes et de commentaires, facilitant ainsi la collaboration.

Ce mode de fonctionnement nous a permis de visualiser clairement l'ensemble des tâches à réaliser, de répartir les responsabilités de manière équitable, de suivre l'avancement du projet en temps réel ou encore d'améliorer la communication et la coordination au sein de l'équipe.

Voici un exemple de tableau Trello utilisé lors du projet :

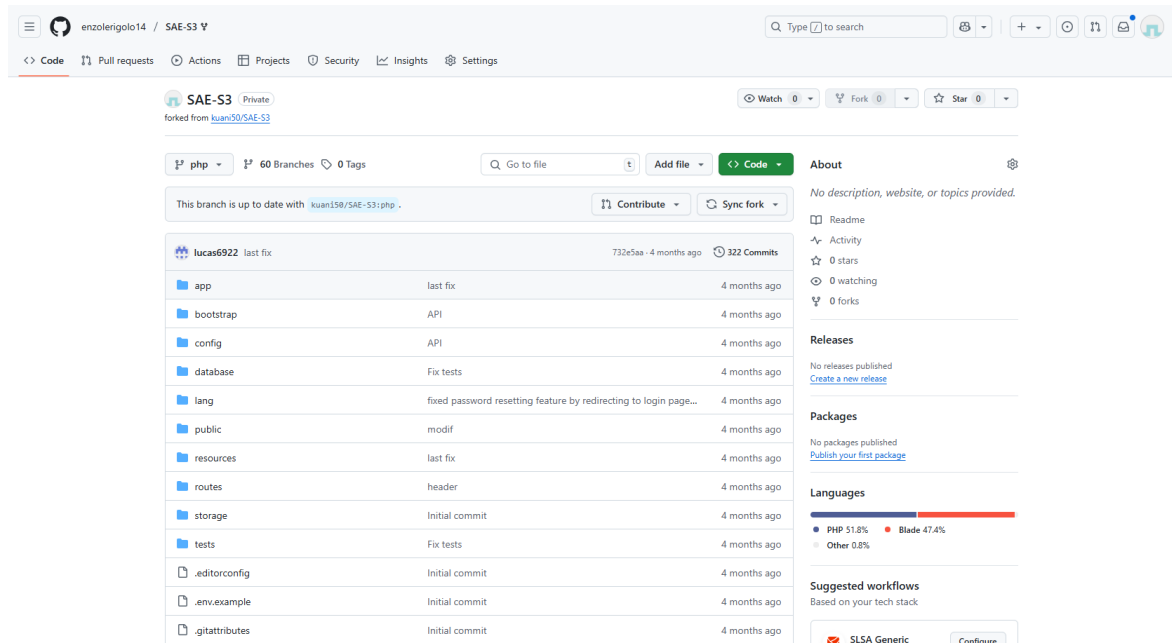


On peut donc voir que l'outil est clairement indispensable pour la bonne gestion d'un projet. Mais ce n'est pas le seul outil qui est très utilisé pour une bonne gestion des projets, en effet énormément de personnes utilisent le logiciel git hub qui est indispensable pour le partage de code en commun et le travail en commun.

GitHub est un outil particulièrement complet pour le développement collaboratif. Il permet non seulement de stocker et versionner le code source, mais aussi de suivre l'évolution d'un projet, de collaborer efficacement en équipe, et de centraliser les contributions de chacun.

L'interface de GitHub est à la fois claire et intuitive, comme en témoigne la capture d'écran ci-dessous :





GitHub ne se limite pas à l'hébergement ou au suivi du code source, c'est également un véritable outil de gestion de projet. En plus de permettre le travail collaboratif sur un même dépôt, il offre des fonctionnalités avancées très utiles pour organiser efficacement les différentes étapes du développement.

Par exemple, il est possible de créer des tableaux de type Kanban directement sur la plateforme, à la manière de Trello. Ces tableaux permettent de répartir visuellement les tâches selon leur état d'avancement (à faire, en cours, terminé), ce qui facilite la coordination entre les membres de l'équipe. Cette fonctionnalité intégrée à GitHub offre ainsi une vue d'ensemble claire sur le déroulement du projet, sans avoir à recourir à un outil externe.

Par ailleurs, la gestion des branches dans GitHub permet une répartition efficace du travail. Chaque membre de l'équipe peut développer une fonctionnalité ou corriger un bug sur sa propre branche, sans perturber le travail des autres. Cette organisation

favorise une meilleure autonomie, tout en simplifiant l'intégration des contributions grâce aux demandes de fusion (pull requests).

Comme le montre la capture ci-dessous, l'interface de GitHub est conçue pour rendre ces fonctionnalités facilement accessibles et intuitives à utiliser, ce qui en fait un outil complet pour le suivi technique et organisationnel d'un projet web.

Branch	Updated	Check status	Behind / Ahead	Pull request
php	now		(Default)	🗑️ ...
views	now		261 / 45	🗑️ ...
validation-level	now		262 / 8	🗑️ ...
user-list	now		102 / 0	🗑️ ...
user-creation	now		300 / 0	🗑️ ...
training-modification	now		235 / 2	🗑️ ...
training-creation	now		233 / 0	🗑️ ...
test-sote	now		20 / 1	🗑️ ...
stutend-formation-display	now		217 / 0	🗑️ ...
student-seance	now		220 / 0	🗑️ ...
student-monitoring	now		265 / 1	🗑️ ...
skill-creation	now		143 / 0	🗑️ ...
session-assignment	now		305 / 0	🗑️ ...
session-assignement	now		246 / 0	🗑️ ...

L'utilisation de Trello et GitHub nous a permis d'organiser efficacement notre travail. Trello a facilité la gestion des tâches et la répartition entre les membres, tandis que GitHub a centralisé le code tout en offrant des outils puissants de collaboration et de suivi de projet. Ensemble, ces outils ont grandement contribué à la réussite du projet.

## 1.6 Compétence 6 : Collaborer

**Apprentissage critique** : Comprendre la diversité, la structure et la dimension de l'informatique dans une organisation (ESN, DSI,...)

Comprendre le rôle de chacun dans une entreprise est essentiel pour bien s'intégrer dans un environnement professionnel. C'est pourquoi, au cours de mes deux années de BUT, nous avons eu l'occasion d'échanger avec différents intervenants, notamment un membre de la DSI de l'Université de Caen, ce qui nous a permis d'avoir un aperçu concret de l'organisation des services informatiques. Un échange avec un étudiant en troisième année d'alternance nous a également permis de mieux comprendre le fonctionnement de l'alternance, ses avantages, ses exigences et le rythme à tenir entre l'université et l'entreprise.

Ces deux entretiens nous ont apporté des informations très utiles, en particulier sur l'alternance, élément clé de la validation du BUT en troisième année.

Mais nos découvertes ne se sont pas limitées aux interviews. Nous avons aussi participé à des réunions au CEMU, où nous avons endossé différents rôles (étudiants, professeurs, responsables de BUT). Ces mises en situation nous ont permis de mieux comprendre le fonctionnement d'un département universitaire et les responsabilités de chacun au sein de l'IUT.

Ces mises en situation nous ont permis de nous glisser dans le rôle de différents acteurs du département, comme les étudiants, les professeurs ou encore le responsable du BUT.

Mais notre apprentissage ne s'est pas limité aux interviews et aux rencontres. Nous avons également eu l'opportunité de travailler sur

Odoo, un outil de gestion administrative utilisé par de nombreuses entreprises. Ce logiciel nous a permis de découvrir concrètement comment sont gérées certaines tâches essentielles comme la facturation, les bons de commande ou encore les bons de livraison.

Nous avons dû réaliser un projet autour de cet outil, ce qui s'est révélé très enrichissant. Même si les consignes étaient limitées, les travaux dirigés effectués en amont nous ont donné les bases nécessaires pour progresser de manière autonome. Cela nous a permis d'apprendre à chercher par nous-mêmes, à expérimenter, et à avancer à notre rythme.

Cependant, Odoo n'est pas un logiciel facile à prendre en main. L'interface est dense, avec de nombreux onglets, sous-onglets, et parfois encore d'autres menus qui apparaissent selon les actions effectuées. Les débuts peuvent donc être déroutants, mais avec de la pratique, on en comprend progressivement le fonctionnement.

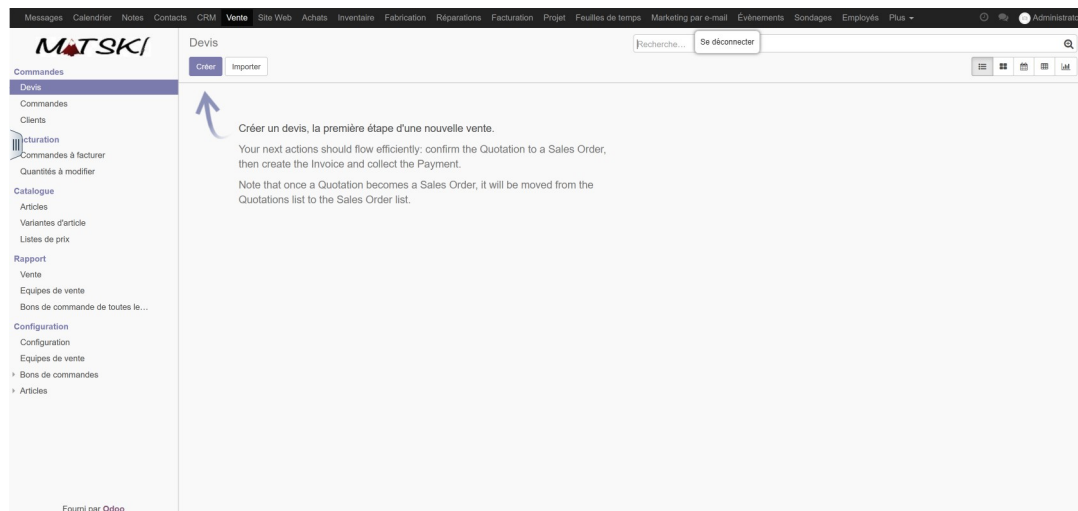
Voici ci-dessous une capture d'écran de la page d'accueil d'Odoo :



Voici à quoi ressemble la page d'accueil lorsqu'on se connecte à Odoo. On y trouve une panoplie d'onglets en haut, chacun ayant une

signification et une importance particulière dans la gestion de l'entreprise.

Voici un exemple d'onglet un peu plus complexe, avec de nombreux sous-menus qui rendent la navigation moins évidente au début, mais qui montrent la richesse fonctionnelle du logiciel.



Les interviews et les échanges réalisés au cours de l'année nous ont permis de mieux comprendre le fonctionnement interne d'un département et les réalités de l'alternance. En parallèle, le travail sur Odoo nous a offert une première immersion concrète dans la gestion administrative d'une entreprise. Ces deux expériences ont été très formatrices et complémentaires dans notre parcours.

## **2 Conclusion**

## **2.1 Conclusion**

Pour conclure, cette année a été particulièrement enrichissante. Au-delà de l'apprentissage du code, nous avons acquis de nombreuses compétences essentielles à la vie en entreprise. Nous avons découvert la gestion et le partage de code, la répartition des tâches entre collègues dans le cadre d'un projet, ainsi que l'importance des droits d'accès et de la sécurité des mots de passe. Nous avons également abordé des notions clés comme le chiffrement, qui permet de protéger les données grâce à l'utilisation de clés de sécurité.

Mais surtout, cette année nous a permis de développer nos compétences en communication, que ce soit lors des réunions ou présentations de projets. Savoir écouter, exprimer ses idées clairement, collaborer efficacement ou encore rendre compte de son travail sont autant de compétences que nous avons pu améliorer, et qui sont indispensables dans le monde professionnel.