

Sprint1:Gestion Des Utilisateurs

1.Introduction:

Dans cette partie de la Sprint1, nous allons nous concentrer sur la gestion des utilisateurs. Cela comprend l'enregistrement de nouveaux collaborateurs, administrateurs et managers sur notre plateforme, ainsi que la mise en place d'un système d'authentification sécurisé en utilisant différents types d'authentification tels que OAuth et OAuth2. Nous allons également mettre en place un système de gestion des rôles et des autorisations pour chaque utilisateur de la plateforme. Enfin, nous allons prendre en compte les erreurs de saisie dans les formulaires et les contrôler pour garantir une meilleure expérience utilisateur.

Dans l'ensemble, cette partie est essentielle pour assurer la sécurité et la fiabilité de notre plateforme. Nous allons nous assurer que les utilisateurs sont enregistrés correctement, que leurs identités sont authentifiées et que leur accès est contrôlé de manière appropriée pour garantir la confidentialité et l'intégrité de notre système.

1.1Methodologies Adopter:

Nous allons adopter la methode agile:Relational Unified Process comme un prcessus de developpement logiciel

- **Methodologie RUP :**

Le **processus unifié (PU)**, ou « *unified process (UP)* » en anglais, ou « *Unified Software Development Process (USDP)* » est une famille de méthodes de [développement de logiciels orientés objets](#). Elle se caractérise par une démarche itérative et incrémentale, pilotée par les [cas d'utilisation](#), et centrée sur [l'architecture](#) et les [modèles UML](#). Elle définit un processus intégrant toutes les activités de conception et de réalisation au sein de cycles de développement composés d'une phase de création, d'une phase d'élaboration, d'une phase de construction et d'une phase de transition, comprenant chacune plusieurs itérations

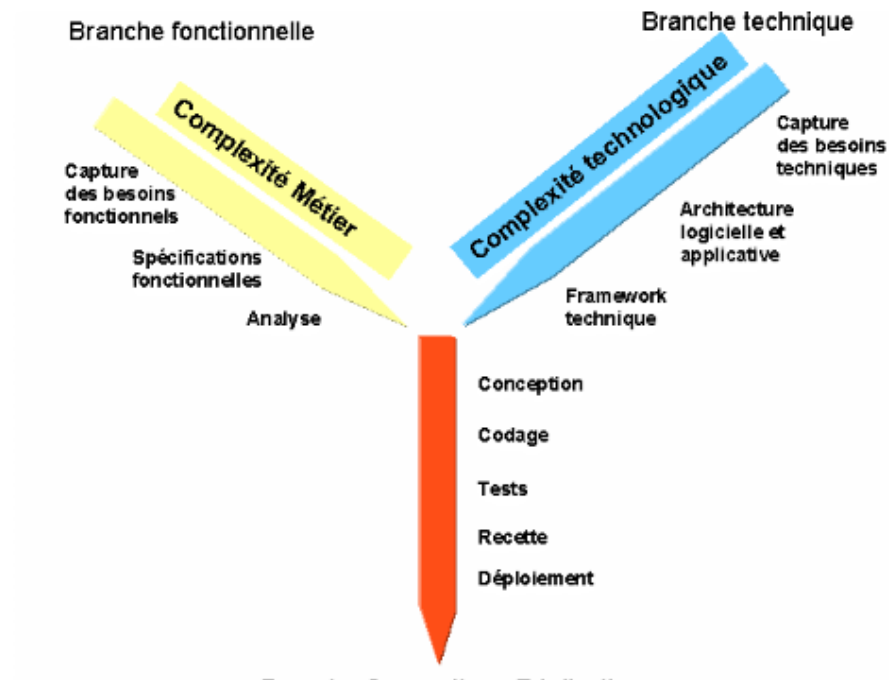


Figure1 -les phases d'un processus unifié

le processus unifié répond à la définition d'un **processus** métier. Il vise ainsi à assurer un cycle de développement avec des enchainements d'activités systématiques et répétables basés sur des artefacts bien définis, tout en facilitant l'intégration de nouvelles personnes dans les équipes.

2.Language de modelistaion :



Afin d'optimiser et sélectionner la méthodologie de conception ainsi que le projet a opté pour UML

Le langage UML (Unified Modeling Language) **est un moyen de représenter visuellement l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes.** Une application comporte des milliers de lignes de code, ce qui rend difficile la gestion des interactions et des hiérarchies.

Analyse et spécification des Besoins

Introduction

L'ANALYSE et la spécification des besoins représentent la première phase du cycle de développement d'un logiciel. Elle sert à identifier les acteurs réactifs du système et leur associer chacun l'ensemble d'actions avec lesquelles il intervient dans l'objectif de donner un résultat optimal et satisfaisant au client.

//plus de detail ICI

2.Specification des Besoins :

2.1.Identification des besoin fonctionnels

- **Enregistrement des utilisateurs** : Les utilisateurs doivent être en mesure de s'inscrire sur la plateforme en fournissant leurs informations personnelles telles que leur nom, leur adresse e-mail, leur mot de passe et d'autres informations pertinentes. Le formulaire d'inscription doit être clair, convivial et contrôler les erreurs de saisie.
- **Authentification des utilisateurs** : Les utilisateurs doivent pouvoir s'authentifier sur la plateforme en utilisant différentes méthodes d'authentification telles que OAuth et OAuth2. Les informations d'identification doivent être stockées de manière sécurisée et les protocoles d'authentification doivent être implémentés de manière sécurisée.
- **Gestion des rôles et des autorisations** : Les utilisateurs doivent être associés à des rôles et des autorisations appropriés en fonction de leurs responsabilités et de leur niveau d'accès à la plateforme. Les rôles et les autorisations doivent être définis de manière à garantir que les utilisateurs ont accès uniquement aux fonctionnalités et aux données auxquelles ils ont droit.
- **Enregistrement des différents types d'utilisateurs** : La plateforme doit permettre l'enregistrement de différents types d'utilisateurs tels que collaborateurs, administrateurs et managers. Les informations spécifiques à chaque type d'utilisateur doivent être collectées et stockées de manière appropriée.
-

- **Contrôle des erreurs de saisie** : Les formulaires de saisie doivent être conçus pour contrôler les erreurs de saisie et les messages d'erreur doivent être clairs et précis. Les erreurs de saisie doivent être contrôlées pour éviter les entrées incorrectes et garantir l'exactitude des données stockées.

En résumé, les spécifications des besoins pour la partie de gestion des utilisateurs de la Sprint1 consistent à garantir la sécurité et l'exactitude des informations stockées sur la plateforme en permettant l'enregistrement de différents types d'utilisateurs, l'authentification des utilisateurs, la gestion des rôles et des autorisations, ainsi que le contrôle des erreurs de saisie dans les formulaires.

2.2. Identification des besoins non fonctionnels

Quand les besoins fonctionnels expriment les fonctionnalités concrètes du produit, les besoins non fonctionnels sont des indicateurs de qualité de l'exécution des besoins fonctionnels.

La norme [ISO/CEI 25000](#), relative à la qualité du logiciel, liste un certain nombre de qualités qui peuvent être approchées comme autant de catégories de besoins non fonctionnels.

- **Sécurité** : La plateforme doit garantir la sécurité des informations personnelles des utilisateurs en utilisant des protocoles d'authentification sécurisés et en stockant les données de manière appropriée.
- **Convivialité** : Les formulaires d'enregistrement et d'authentification doivent être conviviaux, avec des messages d'erreur clairs et des instructions pour faciliter la saisie des informations.
- **Performance** : La plateforme doit être rapide et réactive pour garantir une expérience utilisateur fluide, en minimisant les temps de chargement et en optimisant les temps de réponse.
- **Disponibilité** : La plateforme doit être disponible en tout temps, avec une haute disponibilité pour assurer la continuité des services.
- **Extensibilité** : La plateforme doit être extensible pour permettre l'ajout de nouvelles fonctionnalités et l'évolution de la plateforme avec les besoins futurs.
- **Documentation** : La documentation doit être complète et précise pour faciliter la maintenance, la mise à jour et l'évolution de la plateforme.

Ces besoins non fonctionnels sont importants pour garantir que la plateforme est sécurisée, facile à utiliser, performante, disponible en tout temps et capable de s'adapter aux besoins futurs, tout en offrant une documentation claire et précise pour faciliter la maintenance et l'évolution de la plateforme.

3. Analyse des besoins :

3.1. Identification des acteurs :

Un acteur est une personne physique ou morale¹ prenant part ou affecté par l'action ou le projet en question. Il faut donc commencer par bien préciser par rapport à quelle action ou suite d'actions on cherche à déterminer qui sont et ce que sont les acteurs.

Pour notre system ,on peut distinguer trois acteurs principaux

- Administrateur
- Manager
- Colaborateur

L'Administrateur: ils seront en charge d'administrer et donc de gérer la plateforme. Ils installent, mettent à jour et paramètrent le LMS en prenant en compte les besoins des colaborateur . L'administrateur est en quelque sorte le chef d'orchestre de la plateforme de formation.

Le Manager : le manger s'occupe de favoriser la montee en comptance la circulation des bonne pratique en intern et fixer des objectif clairs et mesuralbe

Colaborateur : se sont experts métiers à partager leurs savoirs et leurs bonnes pratiques (pas de secret ,les enseignants et les enseignants sont des apprenants.)

3.2 Analyse des besoins des acteurs pour le développement du site SkillUp:

Dans cette analyse, nous allons examiner les besoins des trois acteurs principaux pour le développement du partie de Gestion des utilisateur SkillUp: l'administrateur, le Manager et les Collaborateurs.

Acteur 1 : **L'administrateur**

- En tant qu'administrateur, je veux pouvoir gérer les utilisateurs et les autorisations d'accès au site.
- En tant qu'administrateur, je veux pouvoir ajouter, modifier ou supprimer des pages et des contenus du site.
-
- En tant qu'administrateur, je veux pouvoir générer des rapports sur les activités des utilisateurs.

Acteur 2 : **Le Manager**

- En tant que Manager, je veux pouvoir favoriser la montée en compétence des collaborateurs en partageant les bonnes pratiques en interne.

- En tant que Manager, je veux pouvoir fixer des objectifs clairs et mesurables pour chaque collaborateur.
- En tant que Manager, je veux pouvoir suivre la progression des collaborateurs à travers des tableaux de bord et des rapports personnalisés.

Acteur 3 : **Les Collaborateurs**

- En tant que membre, je veux pouvoir créer un compte et me connecter facilement au site.
- En tant que membre, je veux pouvoir gérer mon profil et modifier mes informations personnelles.

En résumé, les visiteurs souhaitent une navigation intuitive et une recherche efficace, les membres veulent un accès facile à leur compte et des notifications pertinentes, et les administrateurs ont besoin de fonctionnalités de gestion avancées. En prenant en compte ces besoins des acteurs, nous pouvons élaborer des User Stories plus précises pour chaque fonctionnalité du site.

3.2 Diagrams de cas d'utilisation :

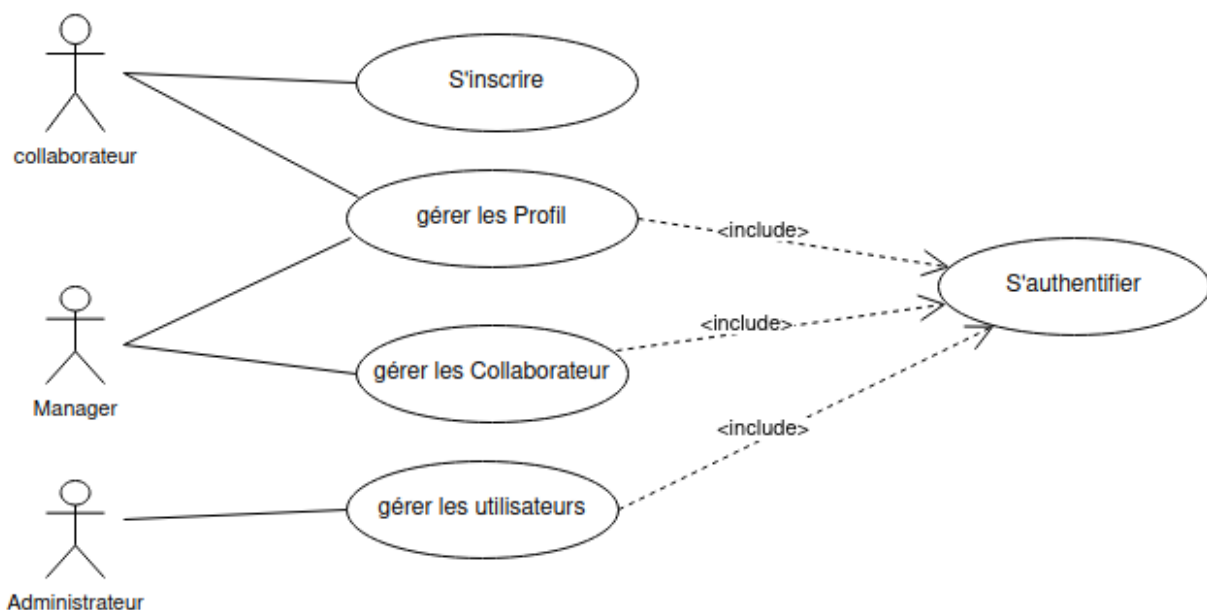


Figure 1-Cas d'utilisation <Gestion utilisateur Global>

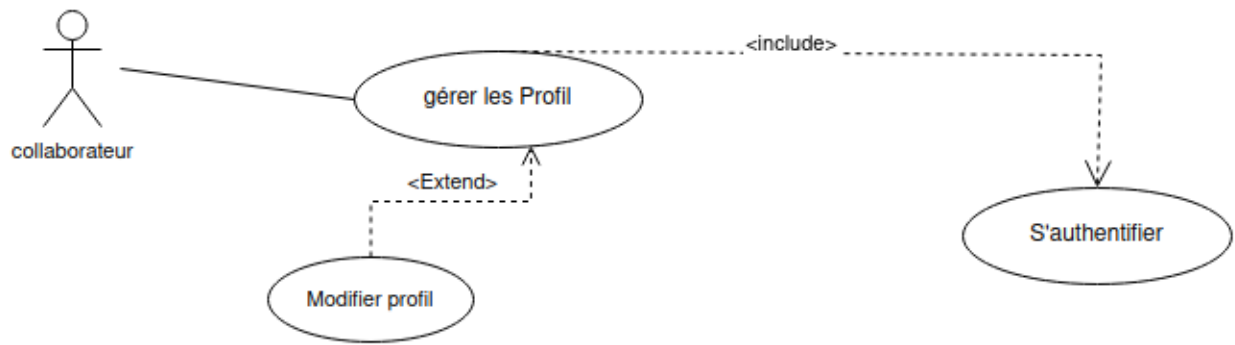


Figure 2-Cas d'utilisation <gerer profile>

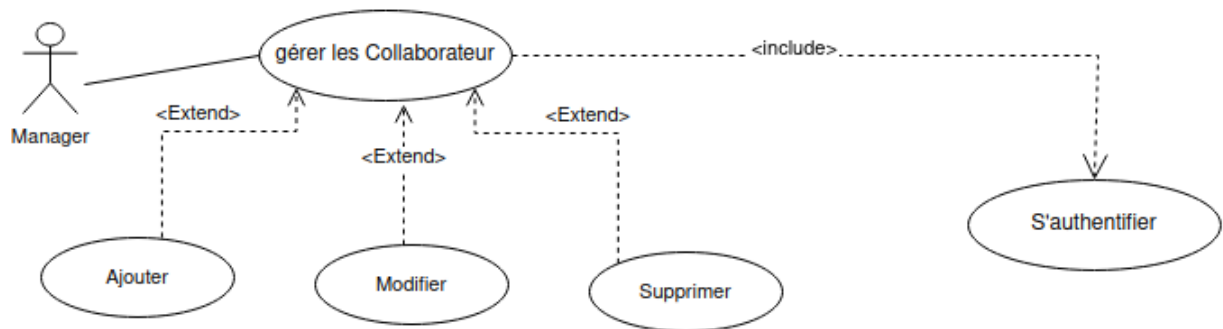


Figure 3-Cas d'utilisation <gerer les collaborateur>

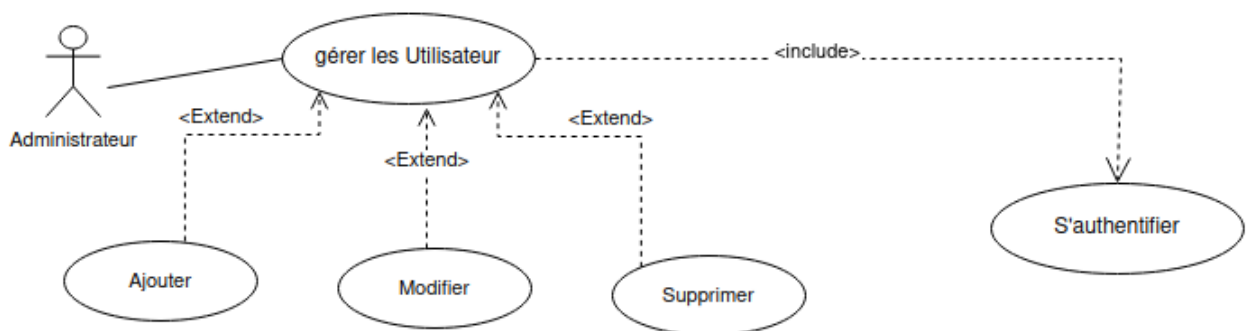


Figure 4-Cas d'utilisation <gerer les Utilisateur>

3.2 Description textuel de cas d'utilisation :

Acteurs : Administrateur, Manager, Collaborateur

Description : Ce cas d'utilisation permet aux acteurs de gérer les utilisateurs et les autorisations d'accès au site.

Pré-conditions :

Les acteurs sont connectés à la plateforme SkillUp.

L'administrateur dispose des droits d'administration nécessaires.

Le Manager dispose des droits de gestion des collaborateurs nécessaires.

Scénario principal :

1. L'administrateur accède à la page de gestion des utilisateurs.

2. L'administrateur peut effectuer les actions suivantes :

- Ajouter un nouvel utilisateur : l'administrateur entre les informations de l'utilisateur (nom, prénom, adresse e-mail, rôle) et définit les autorisations d'accès appropriées.

- Modifier un utilisateur existant : l'administrateur modifie les informations de l'utilisateur ou les autorisations d'accès.

- Supprimer un utilisateur existant : l'administrateur supprime l'utilisateur et toutes les informations associées.

3. Le Manager accède à la page de gestion des collaborateurs.

4. Le Manager peut effectuer les actions suivantes :

- Ajouter un nouvel utilisateur : le Manager entre les informations de l'utilisateur (nom, prénom, adresse e-mail, rôle) et définit les autorisations d'accès appropriées.

- Modifier un utilisateur existant : le Manager modifie les informations de l'utilisateur ou les autorisations d'accès.

- Supprimer un utilisateur existant : le Manager supprime l'utilisateur et toutes les informations associées.

5. Le Collaborateur accède à la page de gestion de son profil.

6. Le Collaborateur peut effectuer les actions suivantes :

- Modifier son profil : le Collaborateur modifie ses informations personnelles.

7. Les acteurs enregistrent les modifications apportées.

Extensions :

Si l'acteur ne dispose pas des droits nécessaires, il ne peut pas accéder à la page de gestion des utilisateurs ou de gestion de profil.

Si l'acteur tente de créer un utilisateur avec des autorisations d'accès inappropriées, un message d'erreur s'affiche et l'utilisateur n'est pas créé. Si l'acteur ne dispose pas des droits nécessaires, il ne peut pas accéder à la page de gestion des utilisateurs ou de gestion de profil.

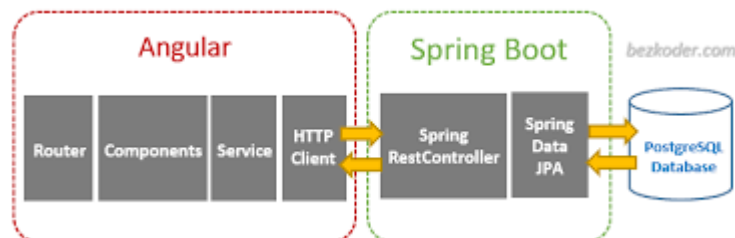
Si l'acteur tente de créer un utilisateur avec des autorisations d'accès inappropriées, un message d'erreur s'affiche et l'utilisateur n'est pas créé.

Conception

Introduction:

La partie conception pour le sprint de gestion des utilisateurs. Dans cette étape cruciale du processus de développement de notre application, nous allons explorer les différentes fonctionnalités et besoins de notre système de gestion des utilisateurs.

1. Architecture MVC:



L'architecture MVC (Modèle-Vue-Contrôleur) est une approche de développement de logiciels qui divise une application en trois parties principales : le modèle, la vue et le contrôleur. Cette architecture permet de séparer les différentes responsabilités de l'application et de faciliter sa maintenance, son évolutivité et sa cohérence.

Dans le contexte de votre projet, vous pouvez utiliser Spring Boot comme serveur backend, Angular comme frontend et PostgreSQL comme base de données. Dans notre projet le model MVC :

- Le modèle représente la couche de données de votre application. Dans votre cas, les données sont stockées dans PostgreSQL. Vous pouvez créer des modèles de données pour chaque entité de votre application (User, Role, Permission, profile.) et utiliser des frameworks comme Spring Data pour accéder à la base de données.

- La vue représente l'interface utilisateur de votre application. Dans votre cas, Angular est responsable de la création de l'interface utilisateur. Vous pouvez créer des composants Angular pour chaque vue de votre application (la page de connexion, la page de profil utilisateur, la page d'administration...) et utiliser des services Angular pour communiquer avec le contrôleur.
- Le contrôleur représente la couche de logique métier de votre application. Dans votre cas, Spring Boot est responsable de la gestion de la logique métier. Vous pouvez créer des contrôleurs Spring pour chaque fonctionnalité de votre application (par exemple, la gestion des utilisateurs, la gestion des rôles, la gestion des permissions) et utiliser des annotations comme @RequestMapping pour gérer les requêtes HTTP.

En résumé, l'architecture MVC est un moyen efficace de structurer votre application en séparant les différentes responsabilités et en simplifiant la maintenance et l'évolutivité de votre code. En utilisant Spring Boot comme backend, Angular comme frontend et PostgreSQL comme base de données, vous pouvez créer une application robuste et scalable.

1. Vue dynamique du systeme:

1. 1 Diagramme de sequence:

Principe

La caractéristique principale d'OAuth 2.0 comme d'OpenID Connect client au fait fait que l'utilisateur n'a plus besoin de s'inscrire sur chaque application à laquelle il veut accéder car la procédure de connexion (la fourniture du login et du mot de passe) se passe sur le **serveur d'authentification**. Il est donc nécessaire de ne s'inscrire qu'une seule fois. On parle d'**inscription unique** (Single Sign On, SSO).

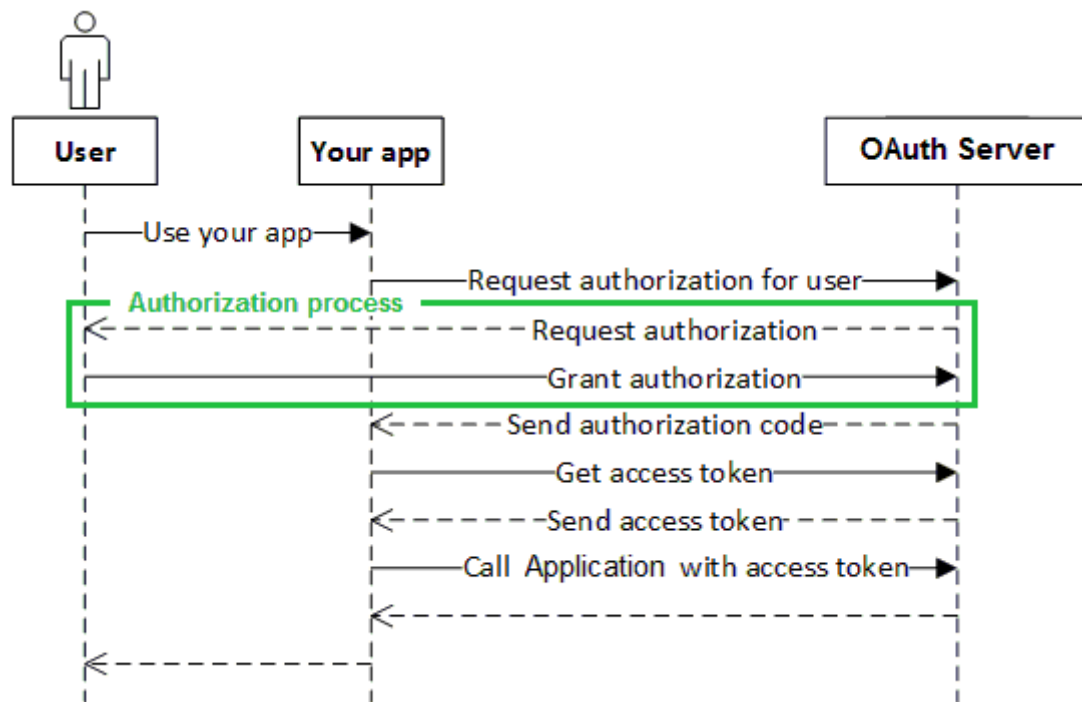


Figure4: L'application ne voit que des jetons, pas les identifiants personnels.

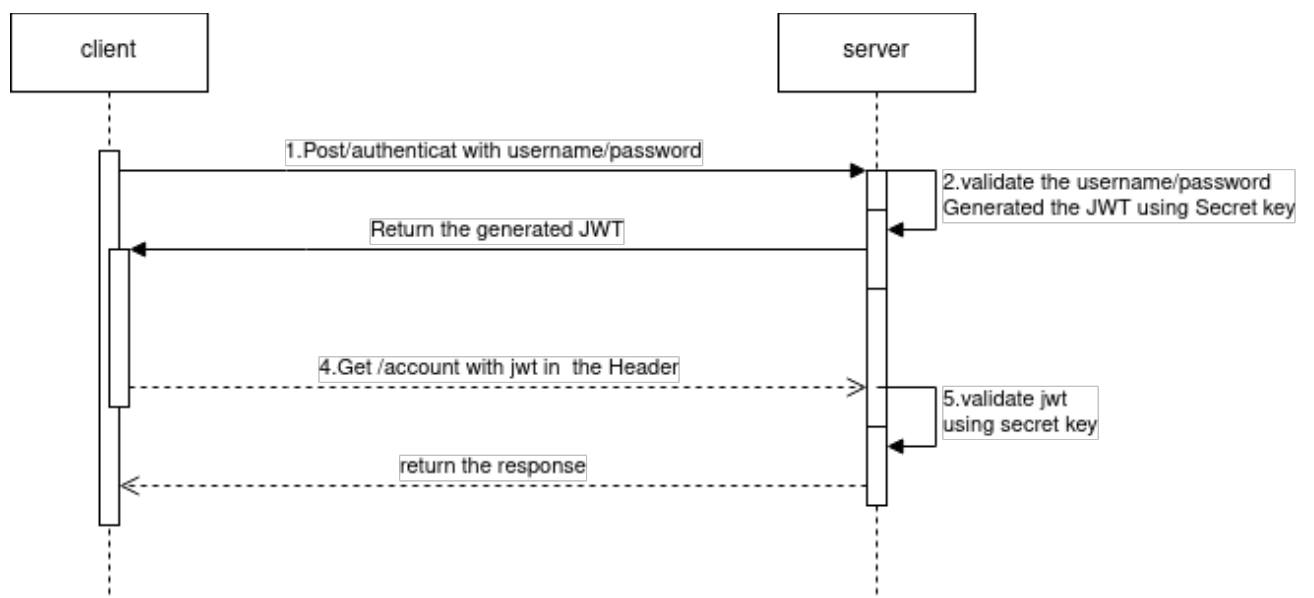


Figure5 : L'authentification avec JWT.

Principe

Le diagramme de séquence décrit le processus d'authentification d'un utilisateur avec JWT (JSON Web Token) dans une application web.

Le processus commence par une requête ;

- POST vers l'endpoint `"/authenticate"`, contenant le nom d'utilisateur et le mot de passe de l'utilisateur dans le corps de la requête.
- Le serveur valide ensuite le nom d'utilisateur et le mot de passe, et s'ils sont corrects, génère un JWT à l'aide d'une clé secrète.
- Le JWT est renvoyé dans le corps de la réponse au client, qui le stocke localement.
- Le client utilise ensuite le JWT pour accéder aux ressources protégées en l'envoyant dans l'en-tête des requêtes ultérieures au serveur.
- Lorsque le serveur reçoit une demande avec un JWT dans l'en-tête, il valide le jeton en utilisant la même clé secrète qui a été utilisée pour le générer. Si le jeton est valide, le serveur accorde au client l'accès à la ressource demandée et renvoie la réponse appropriée.

Dans l'ensemble, le diagramme de séquence illustre une façon sûre et efficace d'authentifier les utilisateurs et de leur accorder l'accès aux ressources protégées dans une application web à l'aide de JWT.

1. 2 Diagramme de class:

