**CHAPITRE I : INTRODUCTION GENERALE**

Actuellement le monde connait une avance technologie considérable dans tous les secteurs et cela en grande partie grâce à l’informatique qui est une science qui étudie les techniques du traitement automatique de l’information. Elle joue un rôle important dans le développement de l’entreprise et d’autres établissements. Ainsi la majorité des secteurs d’activité cherchent à tirer le plus grand profile des technologies de l’information.

Avant l’invention de l’ordinateur, on enregistrer toutes les informations manuellement sur des supports en papier, ce qui engendrait beaucoup de problème tels qu’une perte de temps considérable dans la saisie et la recherche, leur dégradation par une consultation fréquente et leur sécurisation etc.

Ainsi pour évoluer et accroitre leurs efficacités, plusieurs écoles sénégalaises ont besoin d’assimiler la culture de l’innovation portée par ces nouvelles technologies. Ainsi des applications informatiques sont utilisées dans presque tous les domaines pour résoudre des problèmes et offrir des services.

Informatisation des écoles sénégalaise constitue un enjeu affiché pour améliorer la gestion des activités d’une école. Cependant, la plupart de ces solutions informatiques ont été bâties suivant une architecture monolithique. Cette dernière présente beaucoup de limite parmi les quelle on peut citer : la limite technologie, les modifications couteuses, la tolérance aux pannes limitées, les difficultés liées à la maintenabilité et la scalabilité.

**Contexte** :

L’éducation et la formation sont des éléments essentiels pour le développement économique, social et culturel de toute société.

Ceci est encore plus valable pour les pays dont la population est majoritairement jeune comme le cas des pays africains en particulier le Sénégal.

L’éducation scolaire sont géré par des structure éducatives qui sont bien organisées.

Cependant, ces structures rencontrent énormément de problèmes pour faire véhiculer convenablement les informations entre les différents intervenants.

Très souvent la communication se fait via des affiches de toute personne n’ayant pas lu au moment opportun une affiche lui concernant, est susceptible de rencontrer des problèmes. A l’heur numérique, ces pratiques devraient normalement rester derrière nous, mais malheureusement, on constate que ce pratique persiste toujours.

A cote de cette situation, l’utilisation des smartphones et des ordinateurs portables au Sénégal se développe de plus en plus.

Selon le dernier rapport trimestriel sur le marché des télécommunications de l’autorité de Régulation des télécommunications et Post (ARTP), le taux de pénétration de la téléphonie mobile s’élevé à 109,70% au 30 juin 2015 (ARTP,2015).

Aujourd’hui la plupart des élèves des écoles sénégalaise possède des smartphones

Le réseau mobile, très accessible au Sénégal, nous permet d’avoir accès à l’internet à des couts relativement abordables et variables selon les opérateurs (OSIRIS,2013)

La question donc est : pourquoi ne profitons-nous pas de l’internet et de ces smartphones pour la communication entre l’administration, les élèves et les enseignant.

C’est dans ce cadre, que nous abordons notre étude qui consiste à concevoir et réaliser une application web permettant de gérer une structure de formation et d’avoir une communication instantanée, simple et sécurisée entre les acteurs pour une meilleur gestion des informations

Grace a l’outil que nous proposons d’élaborer, la communication passera par internet et non plus par des affiches. Il pourra servir :

1. Aux élèves de recevoir toutes les informations en temps réel sur leur ordinateur ou portable
2. Aux enseignant de recevoir et de faire passer des informations sur leurs enseignements.
3. L’administration, quant à telle, détiendra un outil simple de gestion des informations administrative et pédagogie telle que les convocations à des examens, une information urgente et a la possibilité de gérer les informations

Notre travail consiste à analyser les besoins de la structure de formation, faire une étude comparative des solutions existantes, concevoir notre système en tenant compte des insuffisances des autres solutions puis passer à la mise en œuvre des options et solutions retenues.

**Problématique** :

Apres avoir interrogé les personnels des écoles plus précisément le surveillant générale de l’école de Phare, nous avons constaté quelques anomalie et insuffisances dues à une gestion manuelle.

En effet, nous avons identifié les problèmes suivants :

* Volume important des informations traitées manuellement, ce qui engendre une perte de temps et parfois des erreurs dans établissements des documents
* Recherche difficile sur les registres qui engendre une perte de temps
* Possibilité d’erreur dans le remplissage des différents documents et registres
* Possibilité d’erreur dans les calculs des statistiques
* Un problème d’harmonisation dans le processus de gestion des demandes d’inscription en premier année.

**Objectif** :

Compte tenu de ses problèmes déduit dans notre analyse et dans le souci de trouver des solutions appropriées nous proposons de concevoir et de réaliser une plateforme de gestion d’une école. Afin d’y remédier tous ces problèmes, nous avons assigné à notre plateforme les objectifs spécifiques que nous allons détailler dans le chapitre suivant

* Gestion des élèves
* Gestion de la cantine
* Gestion du transport
* Gestion des parents ou tuteurs
* Gestion des emplois du temps
* Gestion des enseignants
* Gestion des activités scolaires
* Gestion des payements

**CHAPITRE II : CAHIER DE CHARGE ET CAPUTRE DES BESOINS**

**Introduction**

Ce chapitre a pour rôle de montrer les besoins du client, ainsi que les besoins fonctionnels et non fonctionnels de la future application, de préciser les livrable et les risque les plus critiques du projet

**Besoins du client**

Au cours de mon stage concernant le projet de fin d’études au sein du service informatique dans un cabinet de développement d’application, je souhaite établir la mise en œuvre d’un portail-web sur la gestion d’école qui comporte un ensemble de module :

1. **Gestion des élèves** :

Ce module contient quatre volets qui sont : l’inscription, la réinscription, les absences et les retards ainsi que l’exploitation des notes obtenues.

* **L’inscription qui se divise en deux parties**

Celle des élèves qui vont s’inscrire pour la première fois en première années du primaire. En effet, cet acte est statué par l’obligation d’avoir l’âge règlementaire de scolarisation qui est fixé en 6ans.

Les parents des intéressés doivent présenter à l’école d’un dossier (tirer du l’école) après avoir rempli la partie réservée à l’état civil concernant la date de naissance de l’élève. Parmi les pièces d’inscription, on trouve la photo d’identité de l’élève, la photocopie de la CIN des parents ou tuteurs, l’adresse.

En fin les parents ou les tuteurs doivent régler les frais d’inscription, les frais de scolarisation, les frais du transport et de la cantine (les deux dernier son facultatifs).

Les élèves nouveaux dans l’établissement ayant présenté un certificat de sortie réglementaire certifié homologué par la délégation régionale, ces élèves qui ont un niveau supérieur à la 1 er année du primaire peuvent bénéficier aussi d’une inscription à condition d’avoir une place vacante dans le niveau souhaité et remplir les conditions de l’inscription telle que la photo de l’élève, son acte de naissance et les informations personnelles, les coordonnées des parents ou tuteurs.

Le règlement des frais d’inscription, de scolarisation, des droits de transport et de la cantine sont obligatoires sauf les deux derniers points qui sont facultatifs

* **La réinscription des anciens élèves de l’école se fait si** :

L’élève est en situation réglementaire, c’est-à-dire si son nom figure sur la liste des élèves admis en classe supérieure ou ayant le droit au doublage.

La réinscription se fait après la présentation des pièces demandées et après le règlement de frais d’inscription des droits de scolarisation et des services désirés tel que le transport et la cantine.

* **Retards et absences**

Le traitement des absences et de retards des élèves est règlementé par le code interne de l’établissement.

Les absences et retards peuvent être justifier ou non, dans le deuxième cas : les parents ou tuteurs doivent signer un engagement pour la totalité de la responsabilité à l’école vis-à-vis de l’absence ou du retard non justifié de leur enfant.

L’exploitation et le traitement des notes obtenues par les l’élèves dans les contrôles contenus visent à suivre et à mieux contrôler le travail de l’élèves, son évolution ou son retard scolaire est des préoccupations de l’administration.

Les parents sont avisés aussi du comportement de leur enfant au moment opportun, ils sont convoqués à l’école pour décider de la prise d’une mesure éducative en prendre.

Exemple : Soutien de l’élève dans telle ou telle manière …

1. **Gestion des emplois du temps**

Ce module est divisé en deux parties, celui des enseignants et celle des élèves.

**Celui des Enseignants** : l’administration établis un emploi du temps cohérent répondant aux directives et aux normes des inscriptions officielles du ministère de l’éducation nationale pour chaque niveau et pour chaque discipline, en respectant le nombre d’heures pour le tel ou tel niveau concernant telle ou telle matière.

L’emploi du temps une fois établi par l’administration qui l’envoie à l’inspecteur de la matière pour visa.

**Celui des élèves** : l’emploi du temps doit être compatible non chargé (maximum 7 heur/jour), il doit comporter les matières autorisées à être enseignée dans ou tel niveau.

1. **Gestion des enseignants**

Deux catégories d’enseignant sont à mentionner :

**Les vacataires** : ceux qui travaillent par heure.

**Les permanents** : sont ceux qui sont engagés mensuellement par contrat.

La gestion des enseignants consiste à tenir compte de leur dossier professionnel (diplôme, ancienneté, ...). Elle consiste à tenir compte de leurs absences et retards (justifiés ou non) durant l’années scolaire.

1. **Gestion des parents**

Consiste à noter les informations relatives aux parents ou tuteurs des élèves inscrits, tels que : l’adresse familiale, numéro fixe.

L’intérêt et l’avantage de cette application vise à rester en contact avec l’administration de l’école pour donner une vision vivante sur le travail et le comportement journalier de l’élève.

1. **Gestion de la cantine**

Le programme de la cantine scolaire est présenté à l’avance sur un calendrier visé par l’hygiène scolaire.

La présentation du menu de la semaine suivant un emploi du temps précis qui ne perturbe pas le déroulement des cours. Ensuite, les élèves bénéficiaires de la cantine sous la surveillance des responsables (maitre de garde, accompagnateur, …)

1. **Gestion du transport**

Le programme du transport scolaire est présenté à l’avance sur un calendrier visé par l’administration de l’école.

1. **Gestion des activités scolaire**

Parmi les activités scolaires, on trouve les activités artistiques, les activités écologiques et les excursions …

Chaque activité possède un leader ou chef de groupe, ce dernier est le responsable direct vis-à-vis de l’administration qui lui demande d’établir un programme à réaliser suivant un calendrier trimestriel, semestriel ou annuel. Ce calendrier doit comporter et préciser les différents intervenants, les bénéficiaires, les dates d’exécution, les ressources et les moyens disponibles.

A là de l’activité, le responsable doit établir un rapport détaillé sur l’exécution de telle ou telle activité, les inconvénients et les bienfaits de l’activité, comme il doit présenter un dossier illustré par des photos, des CD montrant la réalisation concrète de l’activité (classement des dossiers de l’activité dans le patrimoine culturel de l’école).

La réalisation des activités scolaire vise à casser la routine scolaire à dynamiser et faire renouveler l’effort scolaire de l’élève.

1. **Gestion des payements**

Les parents ou tuteurs peuvent payer en distants selon le service établis par l’administration de l’école ou bien se présentés physiquement au niveau de l’administration de l’école.

1. **Gestion des messages**

La messagerie reste le moyen de communication le plus efficace. La procédure de lancement d’une campagne SMS est très facile.

Un module de communication par SMS pour communiquer facilement avec les parents d’élèves afin de les notifier pour :

* Des absences des élèves
* Des rappels de paiement.

La communication par SMS présente plusieurs avantages :

* L’utilisateur du téléphone portable est averti instantanément.
* Le SMS est un moyen percutant susceptible d’attirer l’attention plus qu’un email.
* Le message est lu immédiatement

**Les besoins fonctionnels**

Avant d’impose une solution il faut se tourner vers le demandeur, pour aboutir de manière structurée à la solution. En effet le but du projet est de satisfaire le besoin. Il faut exprimes clairement les objectifs à atteindre du projet, afin d’éviter toute confusion entre nous et le demandeur.

Le futur système doit permettre à l’utilisateur de gérer un ensemble de modules, dont on trouve :

* Gestion des élèves
* Gestion des emplois du temps
* Gestion des enseignants
* Gestion des parents
* Gestion de la cantine et du transport
* Gestion les activités scolaires
* Gestion des payements
* Gestion des messages

**Les besoins non fonctionnels**

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d’objectifs spécifiques que doit atteindre le système.

**La maintenabilité :** le code doit être compréhensible par simple lecture, notamment en respectant les bonnes pratiques de développement.

**Exploitabilité :** les impacts en termes de performances doivent être pris en compte lors de développement, ainsi que la consommation des ressources qui doit être minimisée.

**Capacité fonctionnelle et convivialité**: les composants développés doivent respecter les spécifications fournies par le client. Le système doit être facilement utilisable et disposer d’interfaces conviviales, notamment par respect des règles d’ergonomie.

**Sécurité :** chaque utilisateur, pour accéder à l’application, est obligé de s’authentifier par un nom d’utilisateur et un mot de passe. Il ne pourra accéder qu’aux pages qui lui sont permis par son profile ou les droits d’accès qui lui sont affectés par l’administrateur.

Le mot de passe ne sont enregistré dans la base de données qu’après un chiffrement (MD5)

**Livrables du projet**

**Chapitre 2 :Concepts fondamentaux d’un micro service**

Dans ce chapitre, nous nous concentrerons sur les concepts de base liés à notre travail. Nous présenterons d’abord les concepts liés à l’architecture des microservices. Puis nous décrirons par la suite l’architecture de microservices. Enfin nous expliquerons le concept d’application cloud native.

1. **Les concepts liés à l’architecture de microservices**
2. **Domain Driven Design (DDD)**

Le développement logiciel désigne le processus consistant à bâtir des applications informatiques. Le concept de logiciels est un art, et comme tout art elle ne peut pas être enseignée et apprise comme une science précise, au moyen de théorèmes et de formules. Nous pouvons découvrir des principes et des techniques utilis

**Chapitre 3** **: Démarche du projet**

1. **Introduction**

Le présent chapitre décrit la conduite du projet. Il présente dans un premier lieu le choix de sa démarche, puis le principe de cette démarche, les rôles du **SCRUM** et en fin l’organisation du projet.

1. **Le choix de la méthode de gestion du projet**

Le choix de la méthode de développement s’est porté vers la méthode **SCRUM**.

SCRUM est la méthode Agile la plus utilisée parmi les autres méthode Agile. Et de fait, la plus éprouvée.

D’autre part, SCRUM est un processus itératif et incrémental, représente Framework de développement logiciel agile pour la gestion du développement des produits.

Il définit « une approche souple, stratégie de développement de produits holistiques et permet aux équipes de développement de s’organiser comme une unité pour atteindre un objectif commun ».

L’une des particularités de SCRUM est que pendant le développement de produits, les clients peuvent changer d’avis sur ce qu’ils veulent et ont besoin (souvent appelé la volatilité des exigences)

1. **Principe de SCRUM**

SCRUM est une méthode agile dédiée a la gestion de projet. Cette méthode de gestion a pour objectif d’améliorer la productivité de son équipe.

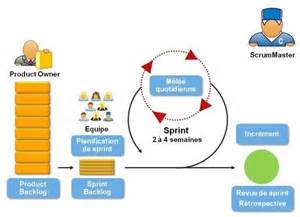
La méthode SCRUM implique que le projet progresse à travers la mise en place de série de « sprints ». A chaque lancement d’un sprint, une réunion de planification est organisée afin que chaque membre de l’équipe puisse s’engager sur le nombre de taches qu’il pourra exécuter, ainsi que sur la création du « sprint blacklog », qui est la liste globale des taches à réaliser lors du sprint.

Chaque jour du sprint, tous les membres de l’équipe (ainsi que le responsable de produit et le SCRUM Master) doivent assister à la réunion SCRUM quotidienne. Cette dernière ne doit pas durer plus de 15 minutes, et permet aux membres de l’équipe de partager aux autres ce qu’ils ont fait la veille, ce sur quoi ils travaillent le jour même, ainsi que l’identification de tout problème pouvant entraver le bon déroulement du sprint. Cette réunion permet ainsi de synchroniser tous les membres de l’équipe.

La fin d’un sprint est marquée par une session de débriefing permettant de présenter le travail achevé au responsable produit, et de partager des informations pouvant influer sur le print suivant.

Voilà un schéma qui représente le processus de la méthodologie SCRUM, avec un détail de chaque étape :

Image.



1. **Product backlog**

Les utilisateurs constituent un produit de backlog, qui va être compose de toutes les demandes de fonctionnalités priorisées. C’est pour cela que dans le produit backlog, on a le dépile par le haut, et le haut du produit backlog représente les fonctionnalités les plus demandées et les plus urgentes, qu’il va falloir réaliser en premier.

1. **Sprint planning :**

Comme l’indique le schéma, on va sélectionner les PBI « les produits de backlog items » qui sont les plus demandés par l’utilisateur, c’est-à-dire qui est en haut de la pile. Puis dans le sprint planning on va décomposer les PBIs en tache auxquelles on va donner une durée en heures.

1. **Spring exécution :**

Une fois on a fait ce **pring planning** c’est-à-dire qu’on a demandé l’ensemble des taches qui vont être réalisées pendant cette itération ce qu’on appelle un sprint. Eh bien là, on va exécuter l’itération, dans sprint exécution, et c’est là ou on a une équipe de projet, une équipe de développement qui vont réaliser les incréments de fonctionnalités.

1. **Daily scrum :**

Tous les jours on a un petit évènement qui s’appelle le **dialy scrum** pendant le quel on passe en revue ce qui a été développé la veille, ce qui sera développé demain et comment on va s’y prendre.

Cet évènement très court qui prend peu de temps « point rapide sur ce qui s’est déroulé la veille, sur ce qui doit se dérouler aujourd’hui puis suivant le travail »

1. **Potentially shippable product increment :**

Une fois qu’on a fait l’ensemble des itérations, normalement on devrait avoir vingt itérations à peu près égales, maintenant on va avoir probablement un ou plusieurs incréments de fonctionnalités qui sont potentiellement intégrables dans le produit.

C’est pour cela qu’on parle en anglais de **Potentially shippable product increment,** c’est-à-dire à la fin de mon itération je ne vais pas forcement livrer quelque chose, en tous cas, j’ai quelque chose qui devrait pouvoir être mis en production, peut être à la fin de cette itération, peut-être la fin d’une autre itération, mais c’est quelque chose de tangible.

Donc on va montrer tout cela aux utilisateurs pour avoir le retour hi-media, et d’ailleurs une fois qu’on a montré aux utilisateurs ce qui convient de développer, ils vont pouvoir donner l’accord, faire des modifications ou ajouter de nouveau PBIs.

1. **Sprint Review :**

L’évènement suivant c’est sprint review, dans le quel on passe en revue ce qui a été réalisé pendant l’itération avec l’utilisateur et puis le passer à ce moment-là avec ce qui « est demandé par les utilisateurs, de ce qu’il va falloir encore ajouter au produit backlog, ou supprimer. C’est à ce moment-là qu’on fait la revue de ce que le sprint a délivré.

1. **Sprint retrospective :**

C’est la dernière étape dans cette méthodologie, est là on passe en revue les éléments de la méthodologie Scrum dans l’entreprise « ce qui a fonctionné ou ce qui n’a pas fonctionné, en fait, c’est une revue de la méthodologie ».

1. **Les rôles du SCRUM :**

SCRUM définit seulement trois rôles :

* **Le Product Owner** : qui porte la vision du produit à réaliser et travailles-en interaction avec l’équipe de développement. Il s’agit généralement d’un expert du domaine métier du projet.
* **L’Equipe de Développement** : qui se charge de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables. Elle est pluridisciplinaire et peut donc encapsuler d’autres rôles tels que développeur, architecte logiciel, DBA, analyste fonctionnel, graphiste/ergonome, ingénieur système.
* **Le SCRUM Master** qui doit maitriser SCRUM et s’assurer que se dernier est correctement appliqué. Il a donc un rôle de coach a la fois auprès du Product Owner et auprès de l’équipe de développement. Il doit donc faire preuve de pédagogie. Il est également chargé de s’assurer que l’équipe de développement est pleinement productive.

Analyse et conception :

Dans cette partie, on va analyser et modéliser les besoins du client avec le langage de modélisation UML.

L’activité d’analyse et de conception permet de traduire les besoins fonctionnels et les contraintes issues de cahier de charge et de la spécification des exigences dans un langage plus professionnel et compréhensible par tous les individus intervenants dans la réalisation et l’utilisation de l’application.

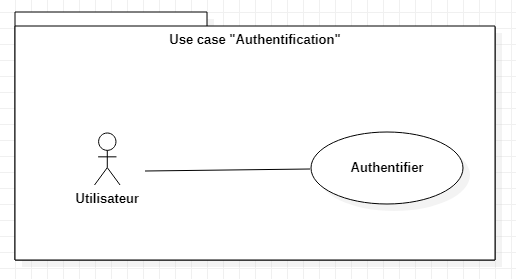
# Identification des acteurs du système

Pour ce système de gestion scolaire, nous avons identifiés les acteurs suivants :

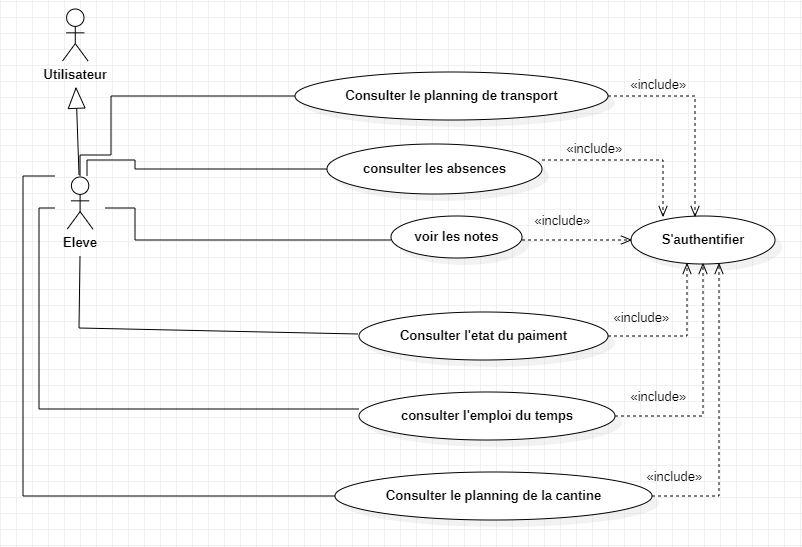
1. Directeur d’école
2. Professeur
3. Secrétaire
4. Surveillant
5. Comptable
6. Caissier
7. Elève
8. Tuteur

# Modélisation des diagrammes de cas d’utilisation

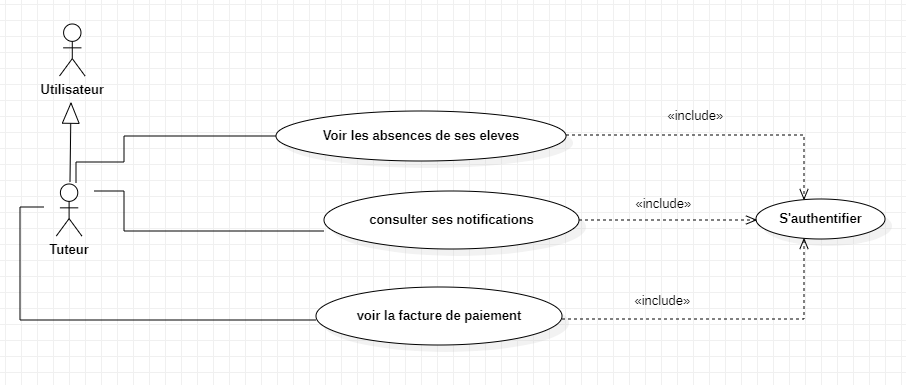
1. Authentification



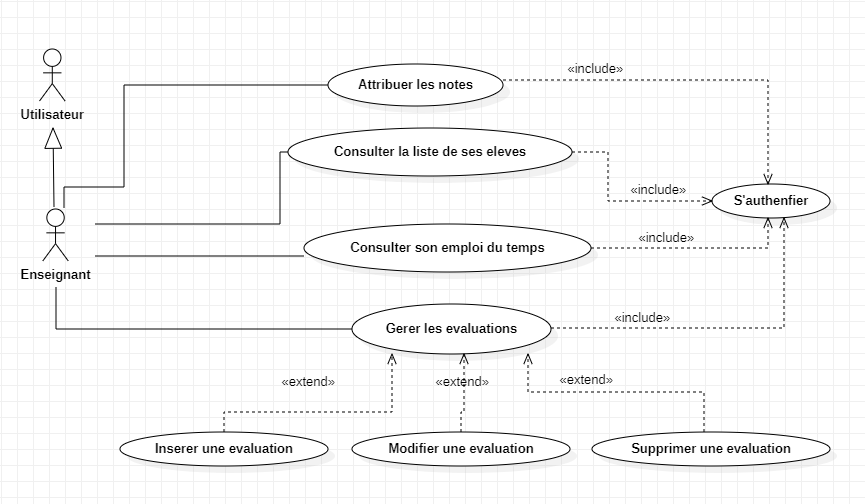
1. Elève



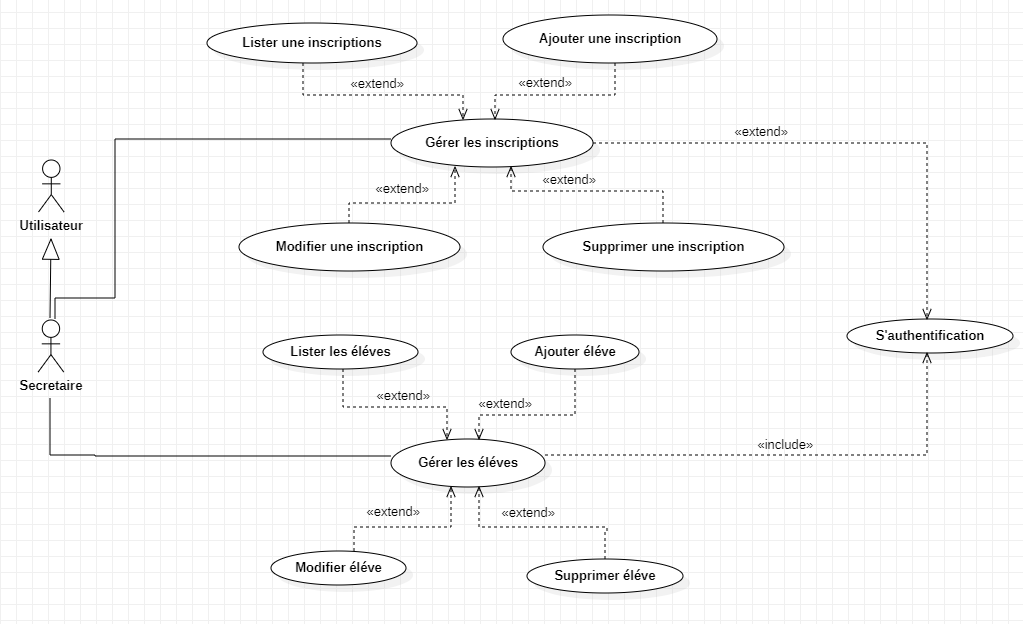
1. Tuteur :

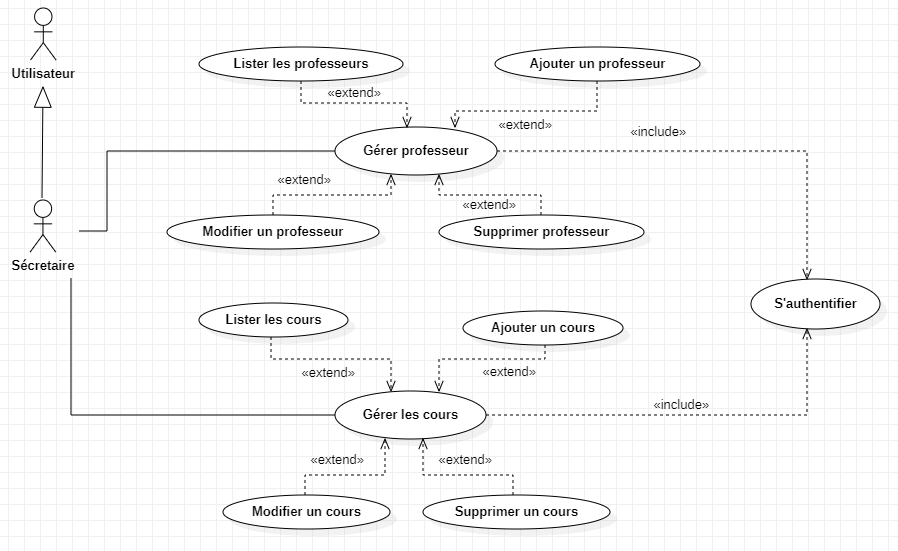


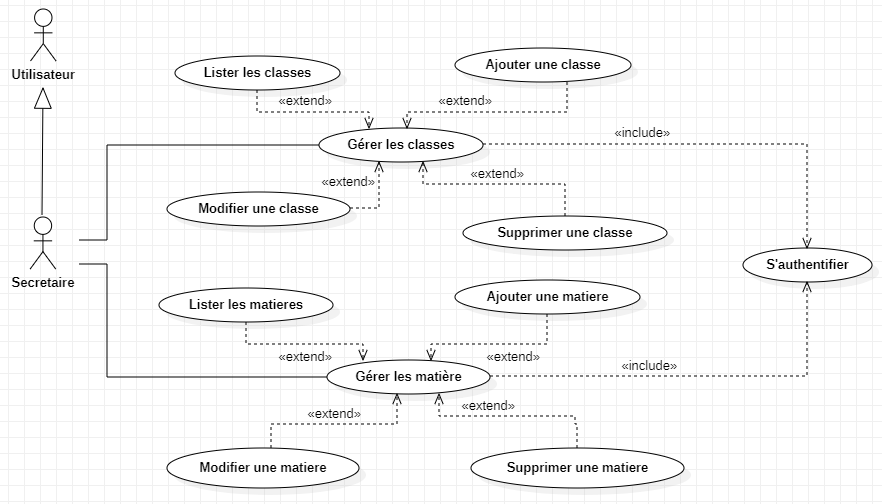
1. Enseignant

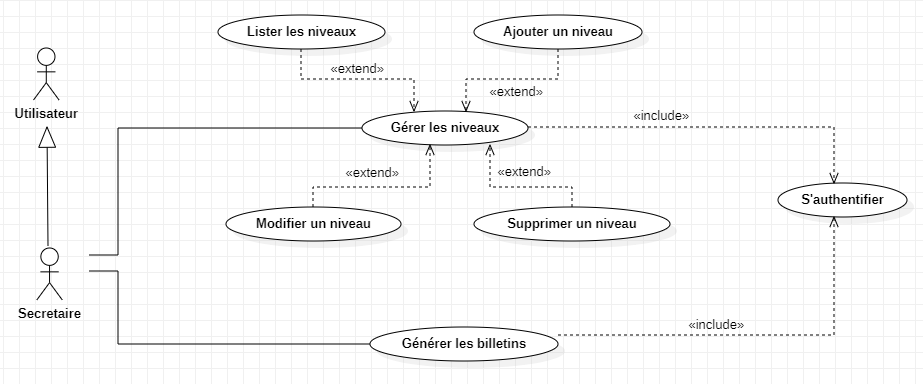


1. Secrétaire :









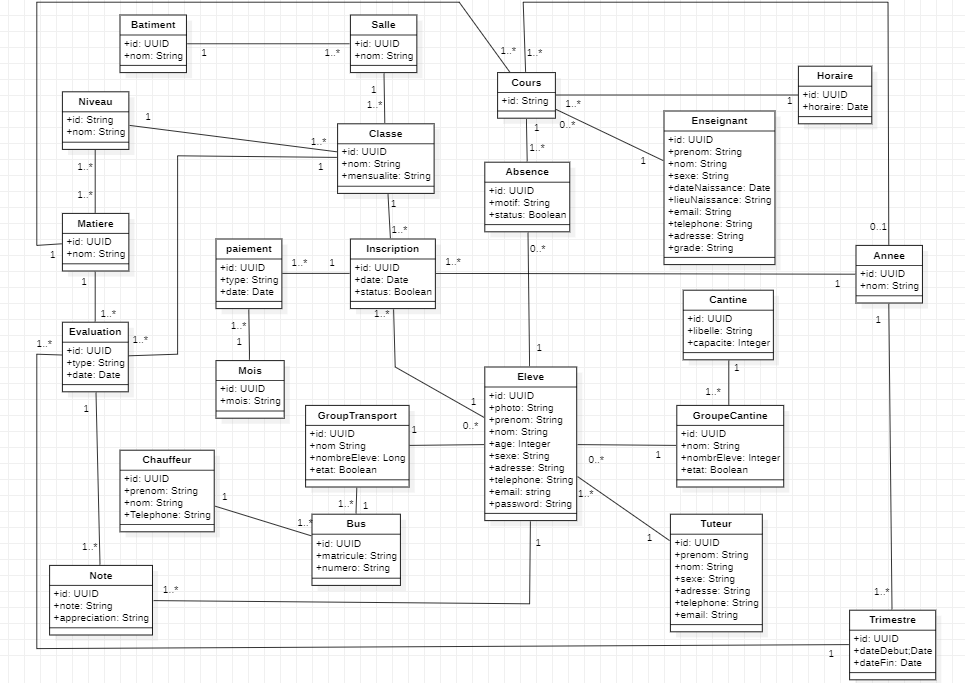
1. Directeur :

# Modélisation globale du Diagramme de classe

C’est un diagramme qui représente la vue statique des objets pages, son intérêt majeur est de modéliser les entités d’un système. Il représente l’architecture conceptuelle du système. Autrement dit, il exprime les relations existantes entre les pages client et serveur.

Le diagramme ci-dessous exprime la structure statique du système en termes de classe et des relations entre ces classes. Ce diagramme permet de représenter l’ensemble des informations finalisée qui sont gérées par le domaine. Ces informations sont structurées c’est à dire qu’elles sont regroupées dans des classes.

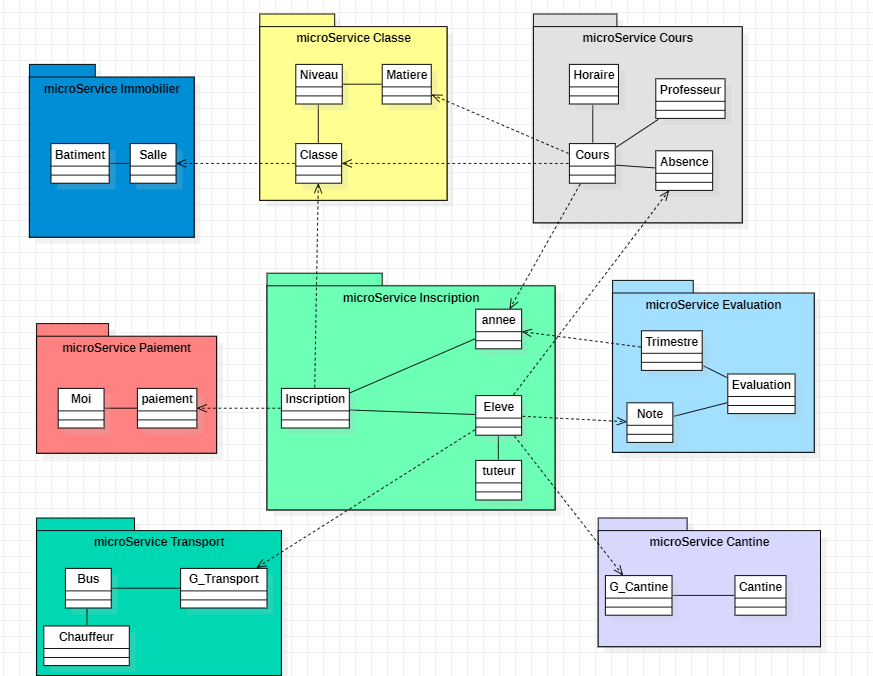
Avant de découper notre application en micro services, nous allons faire en premier temps une modélisation globale système. Le diagramme de classe de notre application est représenté par la figure ci-dessous



# Division de l’application Gestion école en micro services

Pour diviser notre application en micro services, nous définissons d’abord nos contextes bornés.

Chaque contexte identifié représente un micro service à développer. Dans la figure ci-dessous, nous présenterons de manière simplifiée notre carte de mapping ou chaque contexte est représenté par une couleur différente.

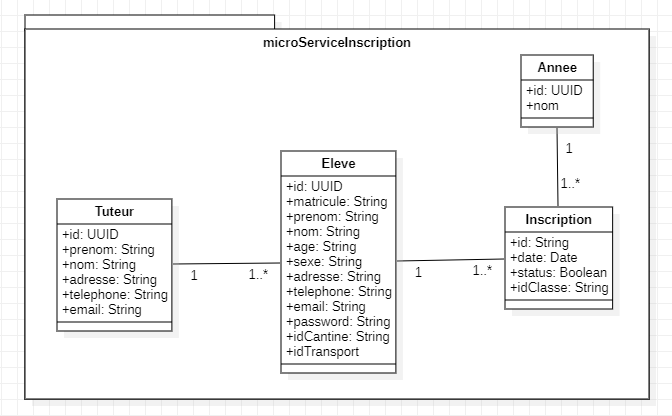


Ce schéma a été fait à laide du pattern bounded context. Utilisation de ce pattern bounded context nous a permis d’identifier huit (8) micro services plus le micro service de la gestion des utilisateurs et leurs privilèges et celui de la gestion des notifications.

Les huit (8) micro services sont :

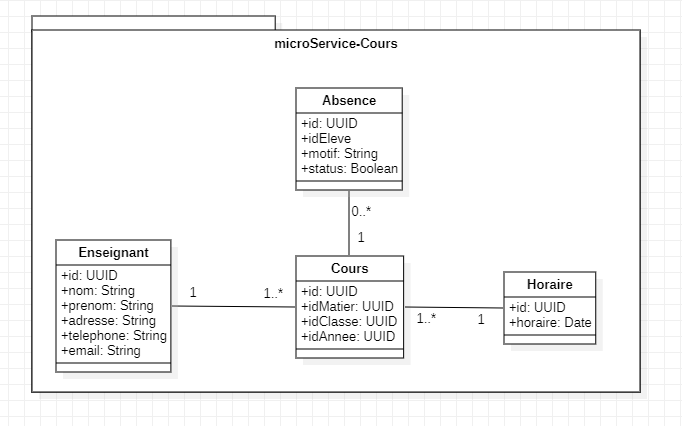
1. **Microservice inscription-service**

Il permet de gérer les inscriptions des élèves, de suivre si un élève est inscrit ou pas. Il permettra d’ajouter, de supprimer, de mettre à jour et d’afficher les élèves inscrits.



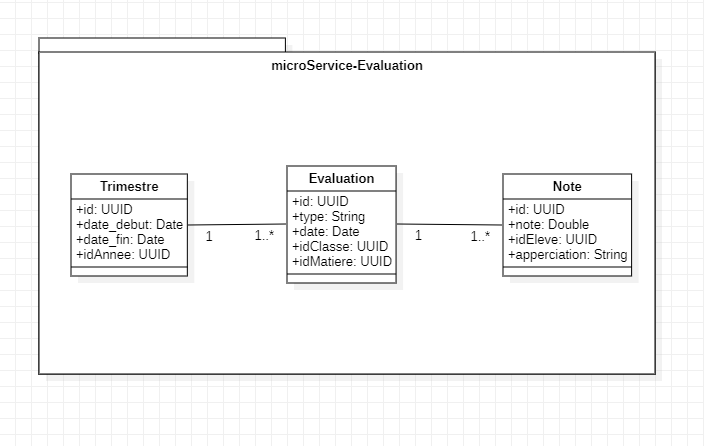
1. **Microservice cours-service**

Elle nous permet de créer des cours de supprimer et de mettre à jour. Elle permettra de gérer les professeurs, les horaires, et les absences.



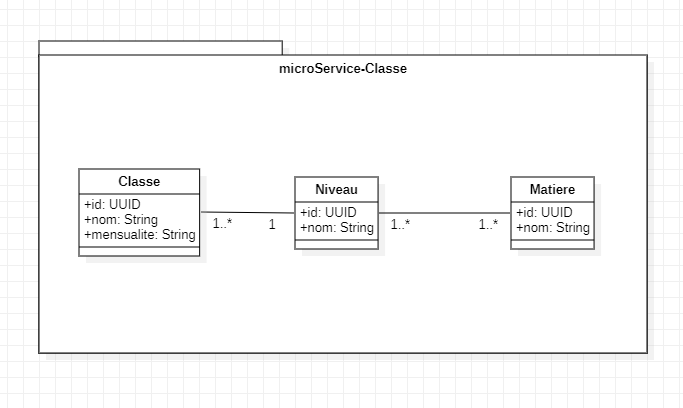
1. **Microservice évaluation-service**

Il permet de gérer les évaluations c’est dire d’ajouter, modifier et supprimer une évaluation. Elle permettra aussi de gérer les trimestres et les notes



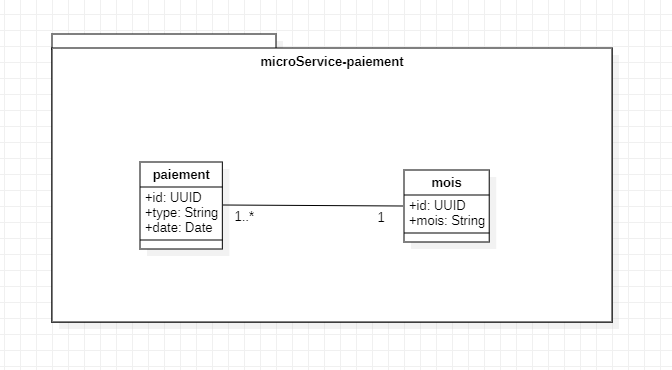
1. **Microservice classe-service**

Il permet d’ajouter, modifier et supprimer une classe. Il permettra aussi de gérer les niveaux et les matières.



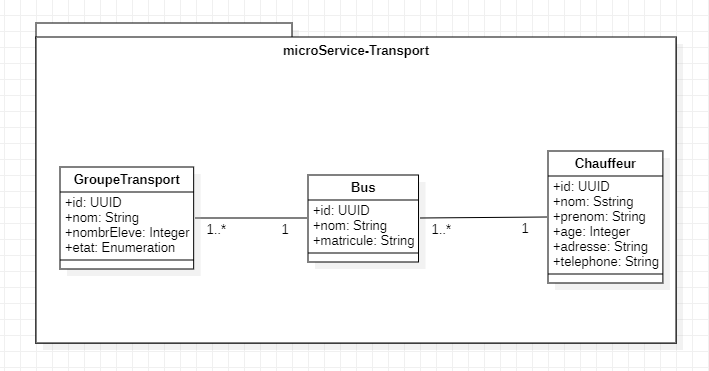
1. **Microservice paiement-service**

Ce microservice permet de gérer les paiements des élèves ou des parents d’élèves. Il permettra aussi de gérer les mois.



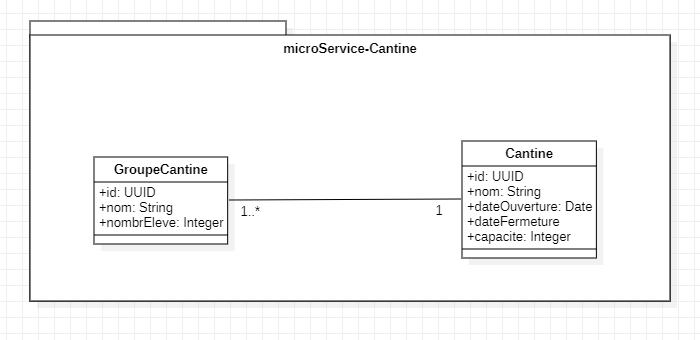
1. **Microservice transport-service**

Il permet de gérer tous les bus de l’école le nombre de personne pour chaque bus et la zone. Il permettra aussi de gérer les chauffeurs surtout la gestion des groupes de transport.



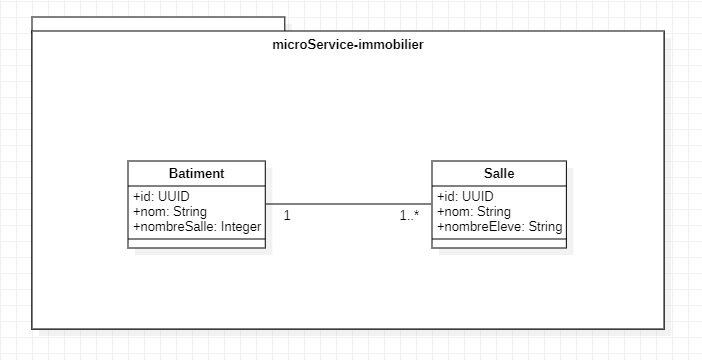
1. **Microservice cantine-service**

Permet de gérer les cantines et les nombre de groupe de cantine qui peut passer heure d’ouverture et heure fermeture. Il permettra de gérer les groupes de cantine



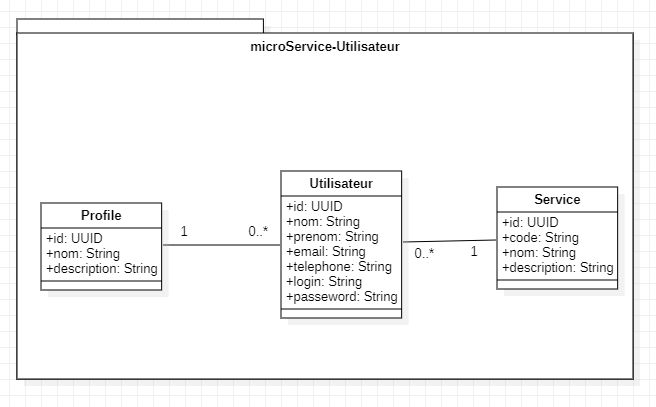
1. **Microservice immobilier-service**

Il gérer tous les patrimoines immobiliers de l’école comme les bâtiments et salles. Il permet aussi de savoir pour bâtiment le nombre de salle et pour chaque salle le nombre d’élève qu’elle peut contenir donc facilite la répartition des élèves dans les salles.



1. **Microservice utilisateur-service**

Il nous permet de créer des utilisateurs du système pour chaque service. Il permettra d’ajouter, de supprimer, de mettre à jour et d’afficher les utilisateurs. En sert de bénéficier de toutes les fonctionnalités de la solution selon leur profil et service.



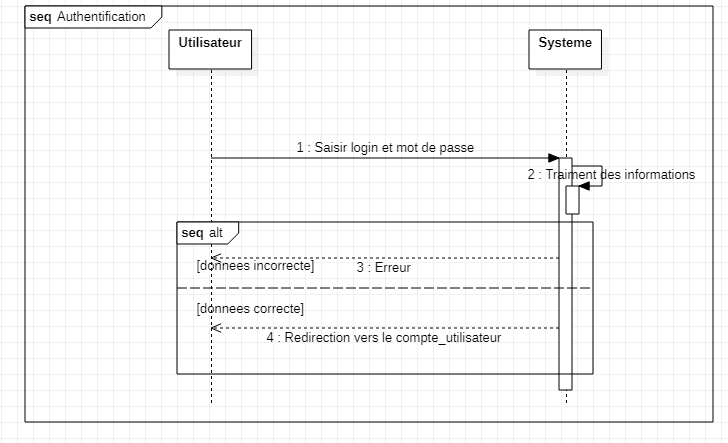
1. **Microservice notification-service**

Il permet de gérer les notifications c’est-à-dire les messages envoie par le système vers les parents d’élève ou les élevés. Il permet d’ajouter de supprimer et de modifier une notification.

# Diagramme de séquences.

1. **Authentification**

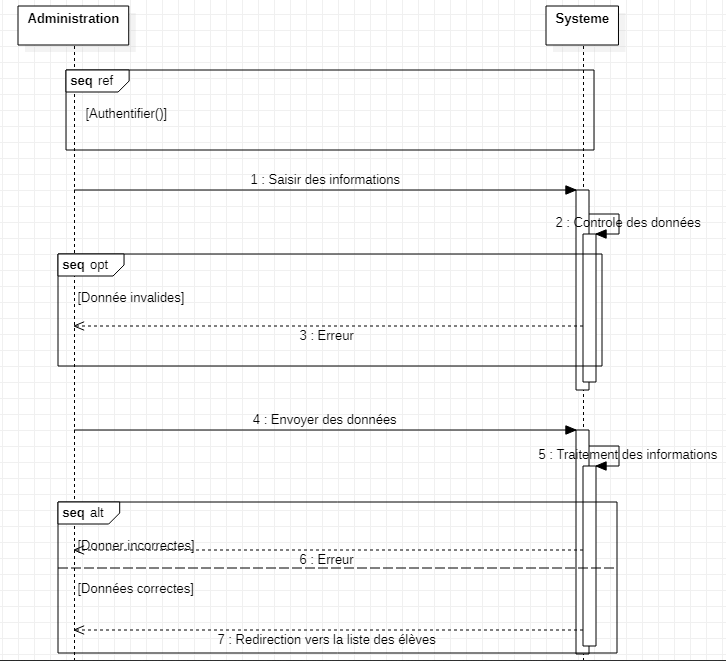
* **Titre**: authentification
* **Objectif** : permet à chaque utilisateur de l’application, de s’authentifier vers ses rôles, à travers un login et un mot de passe
* **Acteurs** : utilisateur-système



**La description des enchainements** :

1. Saisir le login et le mot de passe
2. Envoyer login et mot de passe
3. Traitement des informations envoyés
4. En cas d’erreur, authentification est rejetée
5. Si non, redirection d’utilisateur vers ses rôles
6. **Inscription d’un élève**

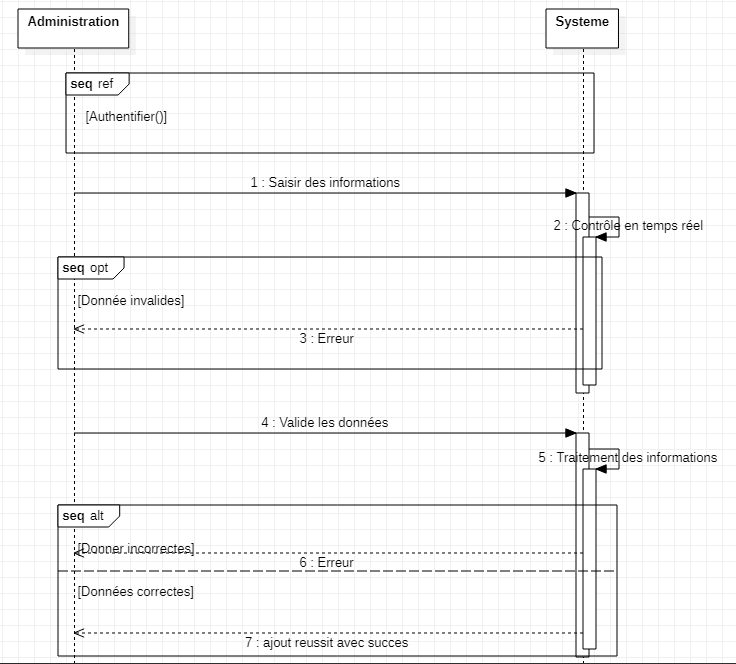
* **Titre** : Inscription d’un élève
* **Objectif** : Permet à l’administration d’ajouter toutes les informations cernant l’élève.
* **Acteurs** : Administration



**La description des enchainements** :

1. Authenfication
2. L’inscription d’un élève doit répond aux conditions d’inscription établie par l’école
3. Saisir les informations de l’élève.
4. Contrôle des données en temps réel en de duplication.
5. Validation de la saisie.
6. Traitement des informations envoyé.
7. En cas d’une anomalie, l’inscription est rejetée en précisant l’erreur effectuée.
8. Si non, l’inscription est effectuée avec succès avec redirection d’utilisateur vers la liste d’élèves.
9. **Diagramme de séquence « Ajout d’un enseignant »**

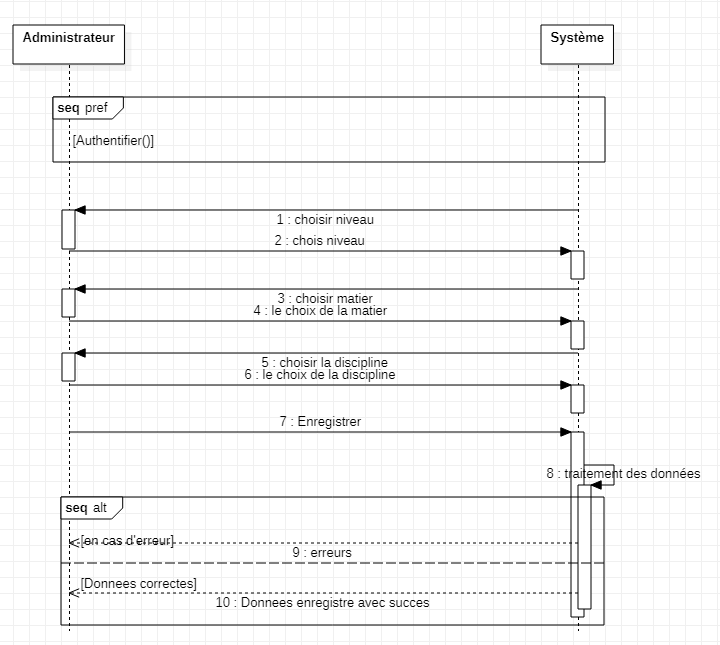
* **Titre** : Ajouter enseignant
* **Objectif** : Permettre à l’administration d’ajouter toutes les informations concernant l’enseignant.
* **Acteurs** : Administration



Description des enchainements :

1. Authentification
2. L’ajout d’un enseignant doit répond aux conditions de recrutement établie par la direction régionale.
3. Saisir les informations concernant l’enseignant
4. Contrôle des données en temps réel, en cas de duplication.
5. Validation de la saisie.
6. Traitement des informations envoyées.
7. En cas d’une anomalie, l’ajout est rejeté en précisant l’erreur effectuée
8. Si non, l’ajout est réussi avec succès avec un redirection de l’administration vers la liste des enseignants.
9. **Affectation des disciplines au niveaux**

* **Titre** : Affectation des disciplines aux niveau
* **Objectif** : Permettre à l’administration d’affecter à chaque niveau ses disciplines, en précisant le coefficient de chaque discipline.
* **Acteurs** : administration – système

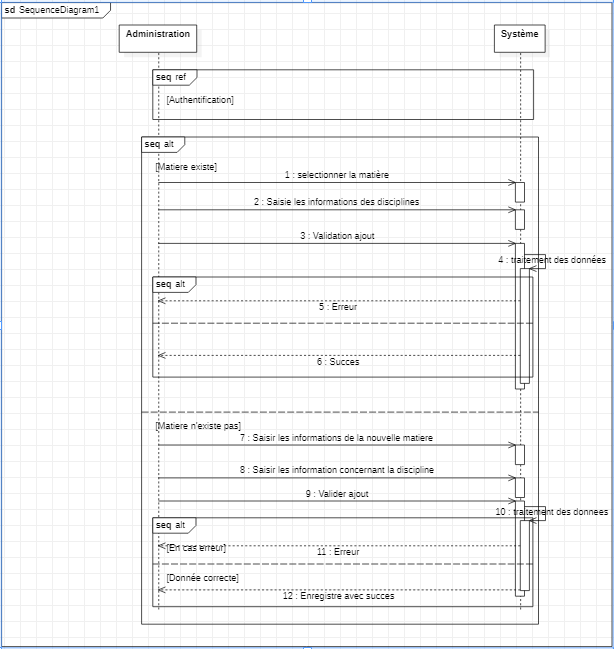


**Description des enchainements** :

1. Authentification
2. Choisir le niveau
3. Choisir la matière, ce qui vous donne les disciplines
4. Choisir la discipline et attribuer un coefficient.
5. Valider des choix
6. En cas d’une erreur, l’ajout est rejeté en précisant l’erreur effectuée
7. Si non, l’ajout est effectué avec succès.
8. **Ajouter discipline**

**Titre :** Ajouter discipline

**Objectif :** Permettre à l’administration un ajout rapide des disciplines de chaque matière.

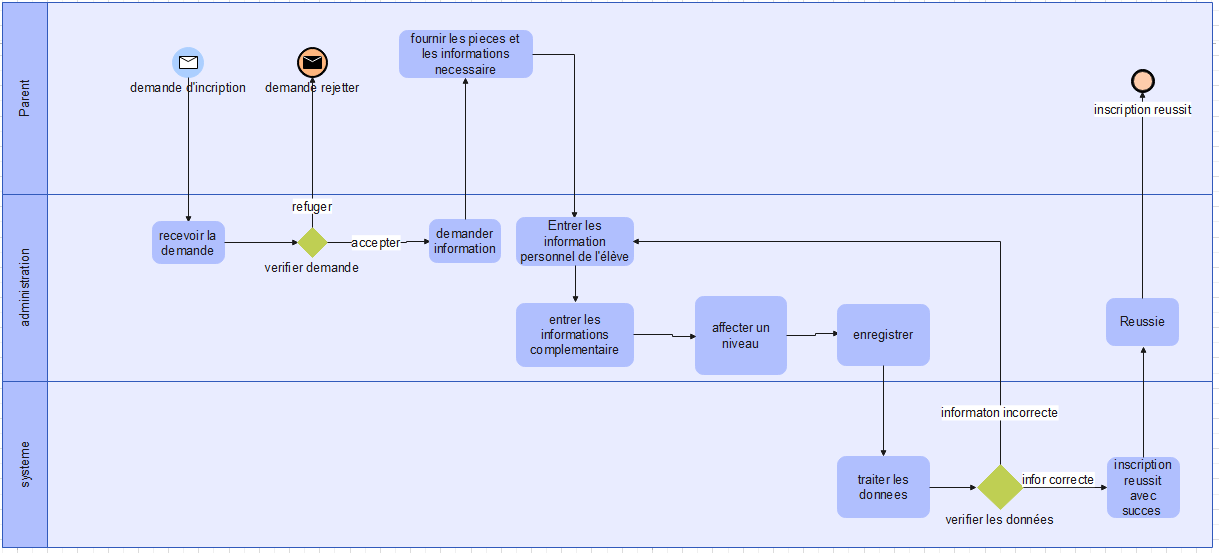
**Acteurs :** administration

**Description** :

1. Authentifier
2. Si la matière existe
3. Sélectionner la matière
4. Saisir les informations des disciplines
5. Validation d’ajout
6. Traitement des informations par le système
7. En cas d’erreur, l’ajout est annulé
8. Si non, l’ajout est effectué avec succès avec la direction d’utilisateur vers la liste des disciplines.
9. Si la matière n’existe pas :
10. Saisie les informations concernant la matière
11. Saisie les informations concernant les disciplines.
12. Validation de l’ajout
13. Traitement des informations
14. En cas d’erreur, l’ajout est annulé.
15. Si non, l’ajout est effectué avec succès avec redirection d’utilisateur vers la liste des disciplines.

# Diagramme d’activités

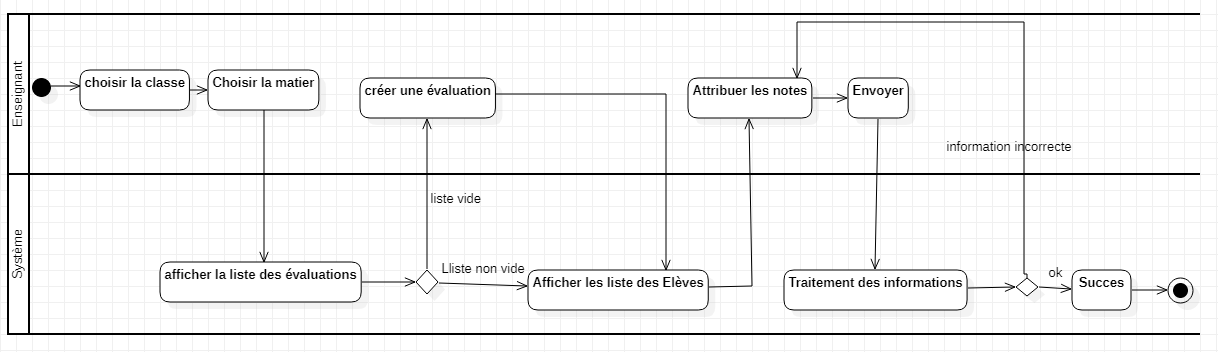
1. Inscription élèves



Description du processus de diagramme d’activité :

* L’élève demande l’inscription dans un niveau
* La demande est reçue par l’administration
* L’administration vérifie les conditions d’inscription de l’élève
* Si l’élève ne réponde pas aux conditions de l’établissement donc la demande est rejetée
* Sinon l’administration demande l’information nécessaire à l’élève
* L’élève doit fournir les pièces et les informations nécessaires pour l’inscription
* L’administration donne les informations personnelles de l’élève.
* L’administration introduit les informations complémentaires et celle concernant la santé de l’élève.
* L’administration affecte le niveau et valide l’inscription
* Le système traite les informations envoyées
* En cas d’erreurs, le système refuse l’inscription demandant à l’administration de vérifier l’erreur
* Sinon, l’inscription est effectuée avec succès
* Le système envoie un message que l’inscription est réussie à administration
* L’administration communique le message à l’élève

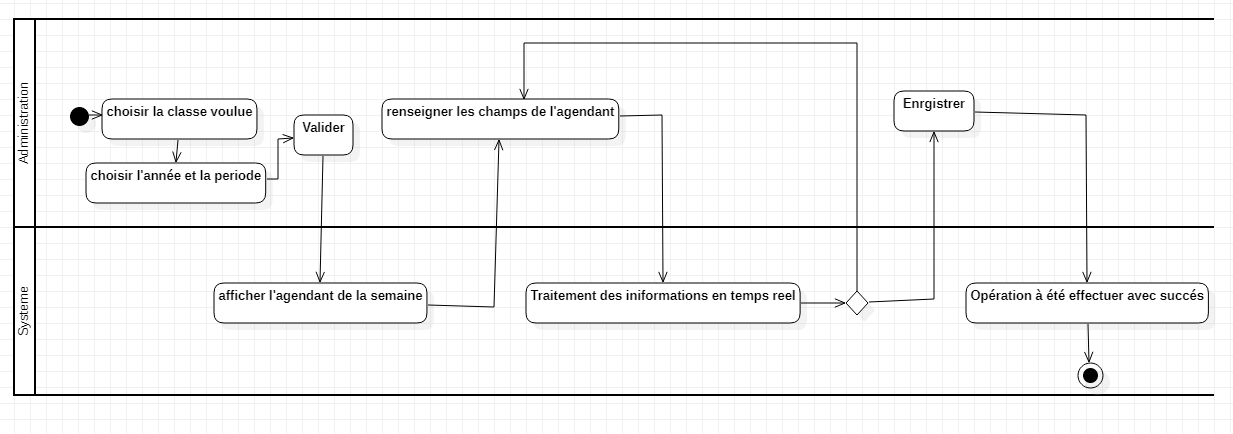
1. **Affectation des notes**



Description du processus de diagramme d’activité « Affecter les notes d’élève »

* L’enseignant Choisi la classe et la matière
* Le système affiche la liste des évaluations
* L’enseignant vérifie si la liste des évaluations est vide
* Si oui, l’enseignant créer l’évaluation
* Sinon le système affiche la liste des élèves
* L’enseignant attribut pour chaque élèves une note puis enregistre les informations
* Le système traite les informations
* Cas d’erreur le système notifies l’enseignant
* Sinon l’affectation est réussie.

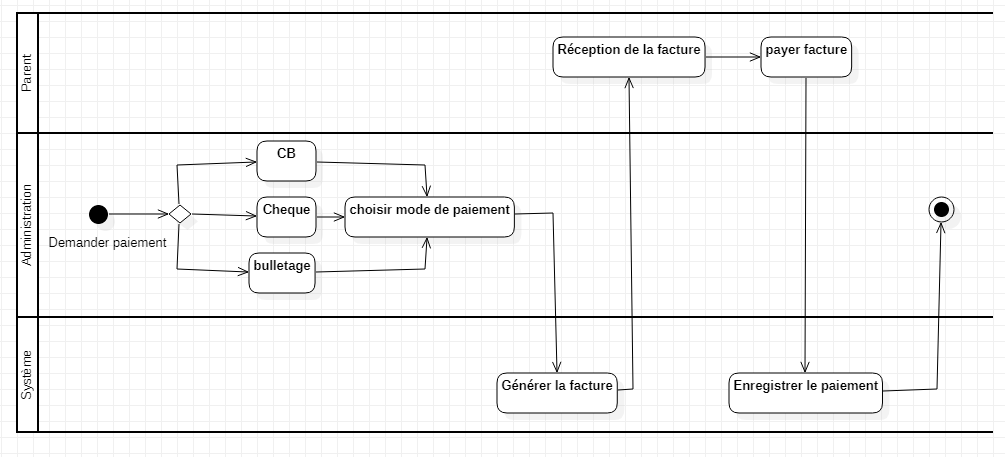
1. Emploi du temps



Description du processus de diagramme d’activité « création emploi du temps »

* L’administration choisi la classe voulue
* Puis choisi l’année et la période puis il valide
* Le système affiche l’agenda de la semaine du lundi jusqu’au samedi
* L’administration renseigne les champs de l’agenda
* Le système contrôle les informations en temps réel
* En cas d’erreur, le système demande à l’utilisateur de corriger l’erreur
* Si non, l’administration envoie les données et le système enregistre l’agenda

1. Paiement



Description du processus de diagramme d’activité

* L’administration choisi le moyen de paiement
* Il enregistre le mode de paiement
* Le système génère la facture
* Et envoie la facture aux parents
* Les parents reçoivent les factures générées par le système
* Les parents paient les factures
* Puis les systèmes enregistrer les paiements

# Diagramme de classes

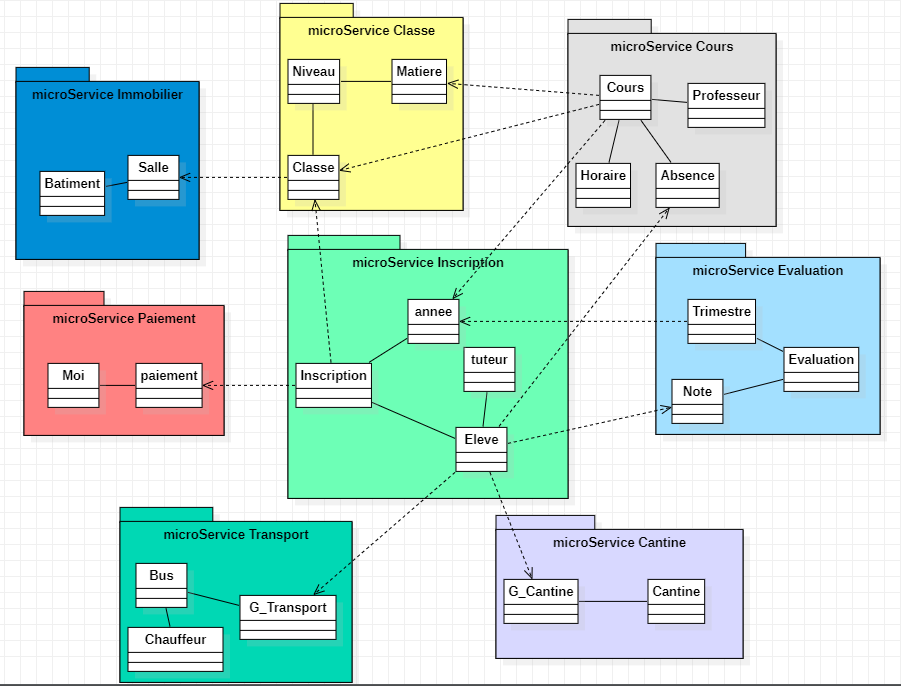
1. Modélisation globale du diagramme de classe

Avant de découper notre application en micro service, nous allons faire en premier temps une modélisation globale du système. Le Diagramme de classe de l’application

1. **Division de l’application Gestion Ecole en micro service**

Pour diviser notre application en micro Services, nous définissons d’abord nos contextes bornés.

Chaque contexte identifié représente un micro service à développer. Dans la figure ci-dessous, nous présenterons de manière simplifiée notre carte de mapping ou chaque contexte est présenté par une couleur différente.



Ce schéma a été fait à l’aide du pattern bounded context. L’utilisation de ce pattern bounded context nous a permis d’identifier cinq (5) micro services plus le micro service de la gestion des notifications et celui de la gestion des utilisateurs et leur privilège.

Ces Cinq (5) micro services sont :

1. Micro service inscription-serve
2. Micro service cours-service
3. Micro service évaluation-service
4. Micro service classe-service
5. Micro service paiement

# Chapitre 4 : Développement des micro services

Dans ce chapitre, nous ferons d’abord le choix des technologies à utiliser pour la réalisation de l’application. Ensuite nous ferons la conception et la réalisation de chaque micros service de notre application.

1. **Choix technologiques**

Dans une architecture micro service, chaque micro service peut être développé avec une technologique différente. Ici il ne s’agit pas de présenter les technologies à utiliser dans chaque micro service mais nous présenterons les technologies globales que nous allons utiliser pour réaliser notre projet.

1. **Choix des frameworks de back-end**

La liste est longue mais nous allons juste exposer les frameworks en vogue.

1. **Spring boot**



C’est un Framework java populaire pour l’écriture de micro services. Il fournit divers projets d’extension sous spring cloud pour créer des micro Service à pile complète. Spring Boot permet de créer des systèmes a grande échelle en démarrant une architecture simple à partir d’un certain nombre de composant collaboratifs. Il peut être utilisé pour construire un système à petite ou grande échelle. Spring boot est très facile à intégrer à d’autre Framework populaire également en raison de l’inversion de contrôle. Spring dispose de divers modules pour intégrer facilement aux bases de données populaire. Il fournit diverses fonctionnalités pour gérer les pannes dans le système distribué.

1. **Quarkus**



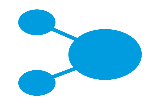
C’est un framework native pour le cloud, conçus en premier par le Red Hat pour l’écriture d’applications java. Quarkus a été imaginé pour permet le développement d’application java dites cloud-natives, ou « kubernetes native » pour reprendre leurs termes. L’objectif de Quarkus est de faire de java une plate-forme leader dans kubernetes et les environnements sans serveur tout en offrant aux développeurs un modelé de programmation réactif et impératif unifié pour répondre de manière optimale à un plus large éventail d’architectures d’application distribuées.

1. **Vert.x**



C’est une boite à outils de développement logiciel open source, réactive et polyglotte des développeurs d’Eclipse. La programmation réactive est un paradigme de programmation, associé à des flux asynchrones, qui répondent à tous les changements ou évènement. De même, Vert.X utilise un bus d’évènements, pour communiquer avec différentes parties de l’application et transmet les évènements, de manière asynchrone, aux gestionnaires lorsqu’ils sont disponibles. Nous l’appelons polyglotte en raison de sa prise en charge de plusieurs langage JVM et non JVM tel que Java, Groovy, Ruby, python et JavaScript.

1. **Moleculer**

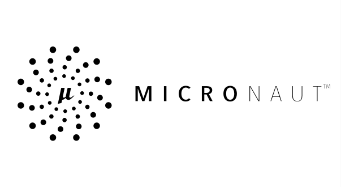


C’est un framework de micro Service intéressant. Comme NodeJS devient populaire, ce framework est préférable si vous êtes un développeur JavaScript. Moleculer est un framework de micro services rapide, moderne et puissant pour NodeJS. Il vous aide à créer des services efficaces, fiables et évolutifs.

Quelques caractéristiques principales :

* Prise en charge de l’architecture évidentielle avec un équilibrage
* Registre de services intégré et découverte de service dynamiques
* Requêtes et éléments a charge équilibrée
* De nombreuses fonctionnalités de tolérance aux pannes
* Solution de mise en cache intégré

1. **Micronaut**



C’est un framework de micro services full stack moderne, basé sur JVM, conçu pour créer des applications de micro services de modulaires et facilement testables.

Micronaut vise à fournir tous les outils nécessaires pour créer des applications de micro services complètes, notamment :

* Injection de dépendances et inversion de contrôle (IoC)
* Valeur par défaut sensibles et configuration automatique
* Configuration et partage de configuration
* Découverte de service
* Routage http
* Client http avec équilibrage de charge coté client

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Spring boot | Quartus | Vert.x | Moleculer | Micronaut |
| Langage utilisé | Java | Java | Polyglotte | NodeJS | Java |
| Portabilité | Oui | Oui | Oui | Oui | Oui |
| Licence | Apache Software  Licence 2.0 | Apache Software  Licence 2.0 | Apache  Software  Licence 2.0 | licenceMIT | Apache  Software  Licence 2.0 |

Pour ce projet notre choix est porté sur le framework spring avec son environnement spring cloud qui fournit des implémentations Discovery Client pour les registres populaires tels que Eureka, Consul, Zookeeper et même le système intègre de kubernetes.

NB : Ce choix concerne seulement la réalisation du service de découverte, du serveur de configuration et la passerelle. Chaque micro service sera développé avec la technologie qui vient le mieux.

1. **Choix des frameworks de front-end**
2. **VueJs**



C’est un framework javascript utilisé pour créer des interfaces utilisateur pour les applications web. Il a été créé par Evan You en 2014 et convient le mieux aux applications monopages, aux composants asynchrones, aux prototypes et au rendu coté serveur. Si vous êtes déjà familiarisé avec JavaScript, vous pouvez facilement utiliser Vue.js, car vous n’avez pas besoin de connaitre des tonnes de JSX ou ES2016 pour commencer. La dernière version de Vue comprend l’API de composition, la modification globale de l’API de montage et des fragments. Certaines entreprises notables qui utilisent Vue.js sont Facebook pour le marketing Newsfeed, le produit Portfolio d’Adobe et l’interface de Grammarly.

1. **Angular**

C’est un framework populaire. Créer par Misko Hevery et Adam Abrono en 2009 en tant que projet parallèle, il est actuellement maintenu par Google. Sa version actuelle, publiée le 03 novembre 2021 s’appelle Angular 13 et est entièrement construite avec TypeScript.

Le framework propose un chargement différentiel, une syntaxe de chargement différé et des API de génération et d’espace de travail. Avec Angular, vous pouvez utiliser du HTML pour vos modèles et lier le HTML pour exprimer clairement vos composants. Il vous aide également à réduire la quantité de code que vous devez écrire avec sa liaison de données et son injection de dépendances, le tout dans le navigateur. Certaines entreprises notables avec Angular dans leur pile technologique sont Google, Udemy et Amazon

1. **React**

C’est une bibliothèque frontale très populaire créée par un ingénieur logiciel Facebook, Jordan Wakle en 2011. Elle est utilisée pour créer des interface utilisateur dynamiques et est très appréciée par les développeurs JavaScript. React a une communauté en ligne florissante, une excellente documentation et même des cours sur son site web. Quelque temps après sa sortie, l’équipe React à créer react Native, un framework pour le développement mobile hybride. React rend les pages d’une manière qui les rend dynamiques et réactive à l’entrée d’un utilisateur. Il s’agit d’une bibliothèque JavaScript open source qui permet de créer d’impressionnantes d’interfaces coté utilisateur, rapides et conviviale pour le référencement. Certaines fonctionnalités de React.js incluent des composants réutilisables, une liaison de données unidirectionnelle et le DOM virtuel. Certaines applications notables créées avec react sont Facebook (page d’accueil), Instagram (géolocalisation et marquage).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Vue.js | Angular | React |
| Langage utilisé | TypeScript | TypeSccript | JavaScript |
| Scalabilité | Faible | Haute | Haute |
| Communauté de développeur | Petite | Grande | Très grande |
| Licence | MIT | MIT | MIT |
| Performance | Haute | Moyenne | Haute |

Pour notre projet, nous avons porté notre choix sur le framework angular qui dispose d’une documentation bien détaillée. Angular dispose aussi d’une grande communauté très disponible.

C’est un framework qui met à la disposition des développeurs beaucoup de librairies faciles à utiliser.

1. **Choix de message brocker**

Les files d’attente de messages sont l’épine dorsale de tout système distribué. Dans les systèmes avancés, une seule application ne peut être responsable de l’ensemble de d’opération. Au contraire, plusieurs applications sont interdépendantes pour exécuter leurs taches et atteindre l’objectif du système dans son ensemble. Les applications individuelles reposent sur le transfert de données pour communiquer avec d’autres application et c’est la que l’on a besoin des files d’attente de messages se fait sentir.

1. **Apache Kafka**

Apache Kafka a été développé à l’origine par LinkedIn puis incubé à apache Software Foundation. Il s’agit d’une plateforme de streaming de données open source écrite en Scala et java. Bien qu’à l’origine conçus comme une file d’attente d’emails, le logiciel est rapidement devenu populaire en tant que plate-forme de diffusion d’événements, traitant actuellement des milliards d’évènements chaque jour. Chaque fois que vous avez besoin d’une file d’attente de messages ou d’un système de courtier de messages, Kafka est un bon choix.

**Caractéristiques :**

* Tout comme un système de messagerie ou un système de mise en file d’attente de messages, Apache Kafka permet de publier et de s’abonner à des flux d’enregistrements.
* Les flux d’enregistrements sont stockés à l’aide d’une approche tolérante aux pannes.
* Il est évolutif car il permet le découplage des applications.
* Il offre un débit élevé pour gérer les transferts de données en temps réel.

1. **ActiveMQ**

ActiveMQ est un service de messagerie open source populaire écrit en java. Il a été créé à l’origine par LogicBlaze en 2004 en tant que courtier de messages open source, puis donné à Apache Software Foundation en 2007. Comme tous les autres courtiers de messages, il agit comme une plate-forme de communication entre plusieurs applications existant sur différents serveurs ou écrites dans différentes langues. Il implémente Java Message Service (JMS) et prend en charge plusieurs protocoles de messagerie, notamment AMQP et MQTT. La dernière mis à jour été publiée en juin 2020.

**Caractéristiques :**

* Plusieurs protocoles de connexion sont pris en charge.
* Les techniques de verrouillage au niveau des lignes de la base de données, le système des fichiers et d’autres modes sont utilisés pour la haute disponibilité.
* Outre l’authentification simple et authentification JAAS, ActiveMQ propose également une API pour les plug-ins d’authentification personnalisés.
* Outre la mis à échelle verticale, une fonctionnalité intégrée de mis à échelle horizontale, appelée Réseau de courtiers, est également prise en charge.
* Prend en charge plusieurs protocoles de transport, notamment STOMP, REST, et OpenWire.

1. **RabbitMQ**

RabbitMQ est un autre courtier de message open source largement utilisé, employé par de nombreuses entreprises à travers le monde. Il est écrit en Erlang et a été conçu à l’origine pour le protocole AMQP (Advanced Message Queuing Protocol), mais a été mis à jour pour prendre en charge d’autres protocoles, notamment STOMP et MQTT. La dernière version 3.8.9 est sortie en octobre 2019.

**Caractéristique** :

* Plusieurs techniques de messagerie sont prises en charge, notamment la messagerie pub-sub, point à point et demande-réponse. Vous pouvez également écrire le votre et le nourrir sous forme de plugin
* Des modes de communication synchrones et asynchrones sont disponibles
* Les accusés de réception garantissent un service fiable
* La haute disponibilité est assurée par la réplication des files d’attente sur plusieurs nœuds d’un cluster, garantissant que les messages ne sont pas perdus même en cas de panne matérielle.
* L’interface utilisateur de gestion permet un contrôle et une surveillance conviviaux des fonctions du coutier de messages

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Apache Kafka** | **RabbitMQ** | **ActiveMQ** |
| **Open source** | Oui | Oui | Oui |
| **Ecrit en** | Scala | Erlang | Java |
| **Protocoles** | Protocole binaire sur TCP | AMQP, MQTT, HTTP, WebSockets | AMQP, AUTO, MQTT, REST, etc. |
| **Langage**  **Supportés** | Java, PHP, Python, Go, Haskell, C#, Ruby, Node.js, Ocaml, etc. | Java, Python, PHP, Ruby, JavaScript, etc. | Java, C, C++, Ruby, Perl, PHP, etc. |

Pour notre projet, nous avons porté notre choix sur apache Kafka.

Pourquoi Apache Kafka ? Le principale avantage de l’utilisation de Kafka est qu’il peut utiliser des partitions pour paralléliser le système des sujets, et chaque partition peut être hébergée sur un ordinateur diffèrent. Les consommateurs peuvent accéder aux sujets en parallèle. Vous pouvez également autoriser plusieurs utilisateurs accéder à plusieurs partitions, ce qui facilite l’expansion. Cette fonctionnalité unique améliore considérablement le débit du traitement des files un grand nombre de charges de travail. Par exemple, LinkedIn utilise Kafka pour gérer 300 milliards d’évènements utilisateur chaque jour.