

UNIVERSITE DE YAOUNDE 1

FACULTE DES SCIENCES

BP 812 Yaoundé



UNIVERSITY OF YAOUNDE 1

FACULTY OF SCIENCES

P.O Box 812 Yaounde

INF3055 – Conception Orientée Objet

THEME : APPLICATION DE GESTION DES REQUETES

Membres du Groupe

CHIN SUILA JOYCE	19M2570
DJEUMENI DJOMBISSIE LEVINNE CLEMENCE	19M2077
FEJUO LAURA FEWAIN	18T2430
NDEMAFO NKENANG FLAVIE DAVILA	19M2267
NGOUNE TSAKA WILFRIED BAUDOUIN	19M2325

Superviseur

Dr VALERY Monthe

1) RECUEIL DES BESOINS

Besoin brutes recueillis

Lors du recueil des besoins de notre projet voici les besoins brutes que nous avons pu avoir


- Toute personne sur l'application doit avoir un compte.
- Chaque utilisateur doit pouvoir se connecter.
- Regarder l'état de la requête
- Envoyer une requête
- Traiter une requête
- Etre notifié (lors de l'envoi ou du traitement)
- Choisir le type de la requête
- On doit savoir à qui on envoie la requête
- C'est pas tout le monde qui traite une requête
- Les requêtes sont envoyées avec les motifs
- On peut voir l'historique des requêtes traitées ou envoyées
- Les requêtes peuvent être classées par (type (.....) ou état (traitée ou non-traitée))
- Un enseignant peut vouloir voir un étudiant
- Les requêtes s'envoient à toute heure
- Pour écrire une requête il faut avoir un compte SR (SoftRequest)
- Il y'a un système d'orientation pour savoir le type de requête à envoyer.
- On peut modifier un compte
- Une requête déjà envoyée ne peut plus être modifier
- Chaque utilisateur peut supprimer son historique
- L'application doit fonctionner en pc comme en mobile

2) ANALYSE DES BESOINS ET SPECIFICATIONS DU SYSTEME

1^{er} besoin : Chaque utilisateur doit avoir un compte


Fonctionnalité : Créer un compte / s'inscrire

Spécification :

 Pour un étudiant

Créer un formulaire comportant les champs

- Matricule (obligatoire)
- Nom et prénom de l'étudiant (obligatoire)
- Mot de passe (obligatoire)
- Matricule
- Filière

 Pour un enseignant

Créer un formulaire comportant les champs

- Grade

- Département

Après envoi de la requête, l'on vérifie les différentes informations remplies puis on les stocke dans la base de données. Une page d'accueil s'affiche par la suite.

2^{ème} besoin : chaque utilisateur doit pouvoir se connecter

Fonctionnalité : Se connecter

Spécification :

Un formulaire qui a deux champs

- Email
- Mot de passe

Après envoi de la requête, l'on vérifie la validité des informations dans la BD, en cas de succès, on ouvre la page d'accueil.

3^{ème} besoin : chaque étudiant peut envoyer une requête

Fonctionnalité : Quel est le type de la requête ?

Spécification :

Types de Requêtes

Un étudiant peut faire une requête pour :

1. Absence de note

Ici plusieurs champs sont mis en jeux :

Infos_etudiants (l'étudiant étant celui émet cette requête, renseignent tous les infos relative à lui). Infos_enseignant (le récepteur de la requête émit par l'étudiant).

Pièce jointe(ou l'on observe des documents relative à la vérification : fichier externe...)

2. Absence de paiement

Mettant en relation deux acteurs principaux :

- L'étudiant : qui est celui qui émet la requête renseignant (nom_etudiant, matricule, filière et faculté)
- Le Doyen : qui est celui à qui la requête est adressée

En plus de ces acteurs nous avons aussi les champs

Pièce jointe : contenant, la photocopie certifiée du reçu de paiement.

3. Activation de matricule

Nous observons des cas d'activation de matricule lors du tirage des fiches d'inscription académique; pour des personnes qui effectuent le paiement des frais académiques en deux tranches puisqu'il possède deux numéros de transactions différents il faut donc appliquer le processus d'activation de matricule pour être en possession d'une fiche intègre. Pour cela deux acteurs mis en jeux :

- Etudiant et Doyen

Comme pièce jointe nous avons :
matricule_etudiant

4. Blocage de matricule

Les différents champs observés sont :

- infos_etudiant (émetteur)
- centre de calcul (récepteur)
- motif : voyage, changement d'école,....

5. Changement de filière

Les différents champs mis en jeux sont :

- Filière : nom_new_filiere
- Motif : ne se sent plus aise où il se trouve
- Infos_étudiant (émetteur)

Pièce jointe : reçu de paiement

- Doyens (récepteur)

6. Matricule erroné

Nous avons ici :

- Le doyen et l'enseignant (qui sont les destinataires de la requête)
- L'étudiant (celui qui émet la requête)

Pièce jointe : nous pouvons observer de toute sorte (fichier externe et objets)

7. Requête personnalisée

Pour des étudiants n'appartenant pas aux différents requêtes précédentes, nous avons créé une nouvelle requête appelée requête personnalisée possédant les champs :

- Motif

Pièce jointe

REMARQUE : nous constatons couramment que les étudiants adressent des requêtes à la haute hiérarchie universitaire mais bien aussi des demandes. Nous pouvons en observer plusieurs objets de demande à savoir :

- a) *Obtention du relevé de note*
- b) *Diplôme académique*
- c) *Fiche de soutenance*
- d) *Rattrapage à l'examen (CC, SN, TP)*
- e) *Obtention du certificat de scolarité*

Les différents champs utilisés sont ;

- Infos étudiant (émetteur)
- Doyens, enseignant (récepteurs)

Pièce jointe : reçu de paiement

I. Envoie de la requête

Qui est l'étape qui suit la rédaction de la requête; elle est matérialisée par un pour **envoyer**.

NB : nous avons créé notre site de telle sorte que la communication se fait uniquement via le site de l'envoi au traitement de la requête (ou dans le site...)

II. Traitement de la requête

Qui est un scénario simple et précis :

4^{ème} besoin : Etre notifié lors de la réception d'une requête / demande

Fonctionnalité : Notifier l'acteur (Etudiant ou Administration)

Spécification :

- A chaque requête / demande envoyée par un étudiant, le destinataire doit être notifié dans sa boîte email.
- A chaque réponse envoyée par un administrateur, le destinataire doit être notifié dans sa boîte email.

5^{ème} besoin : les requêtes sont envoyées par type

Fonctionnalité : Choisir le type de la requête

Spécification : Un menu comportant les différents types de requêtes sera mis à disposition pour que l'émetteur puisse choisir.

6^{ème} besoin : Envoyer une requête à un destinataire

Fonctionnalité : Lier chaque requête à son destinataire

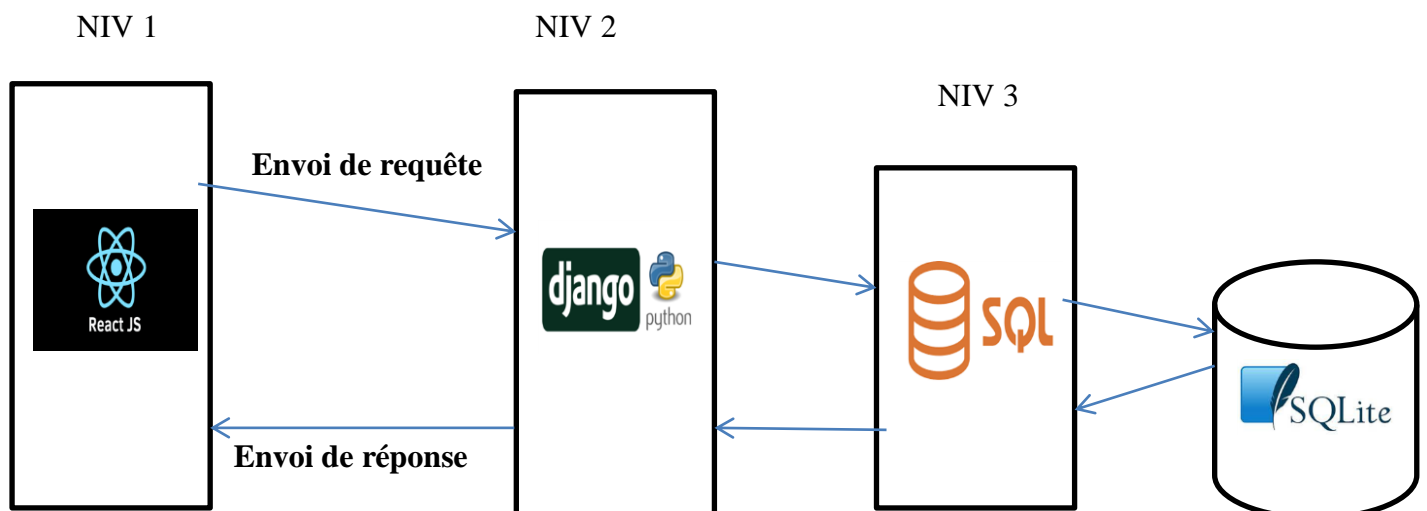
Spécification :

- Lors de l'envoi d'une requête, on stock le nom de l'émetteur et du destinataire dans la table requête.
- On sait exactement pour chaque requête son destinataire.

3) CONCEPTION

○ Conception Architectural

• Architecture du System



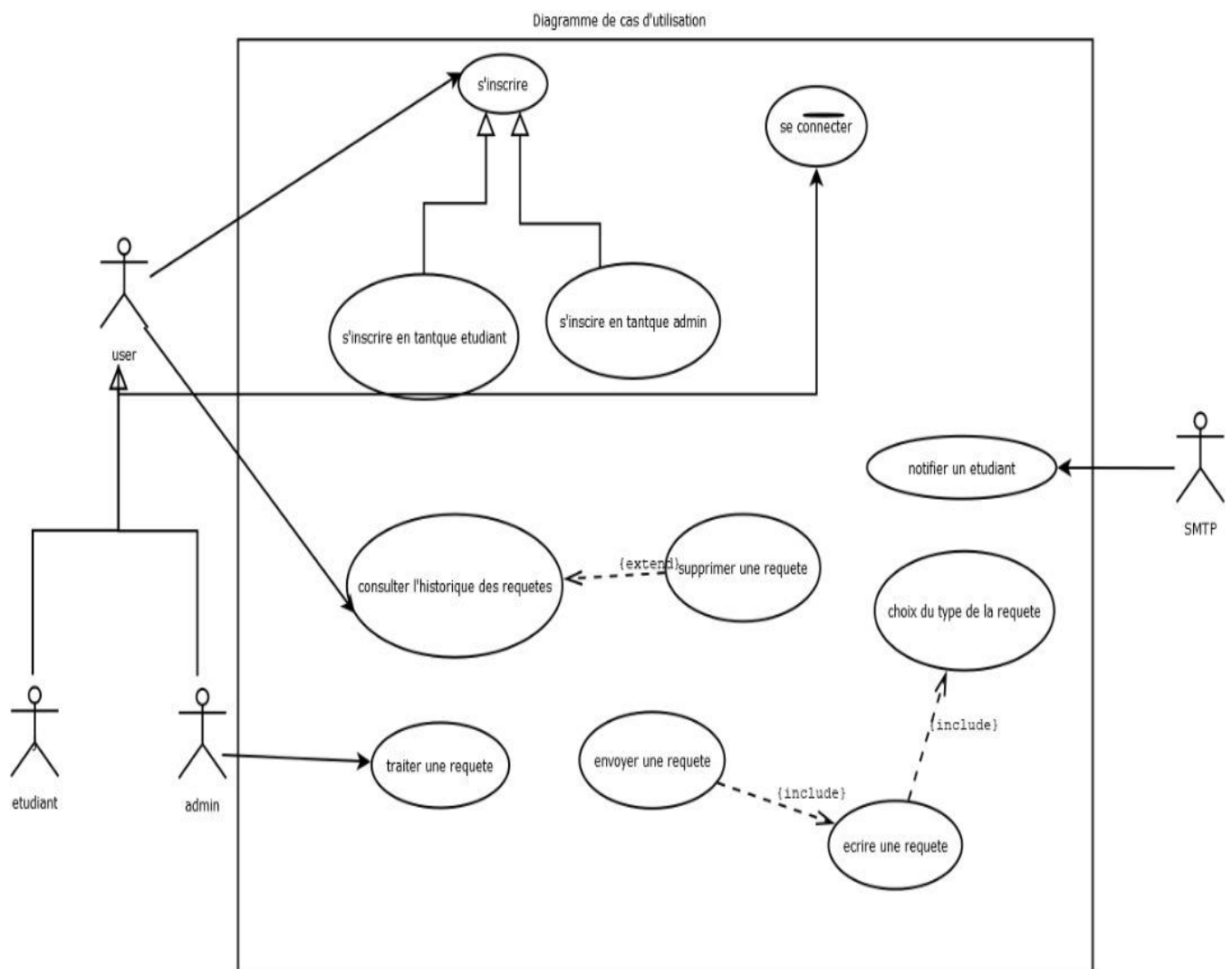
Client

HEROKU (serveur d'application) SQL lite

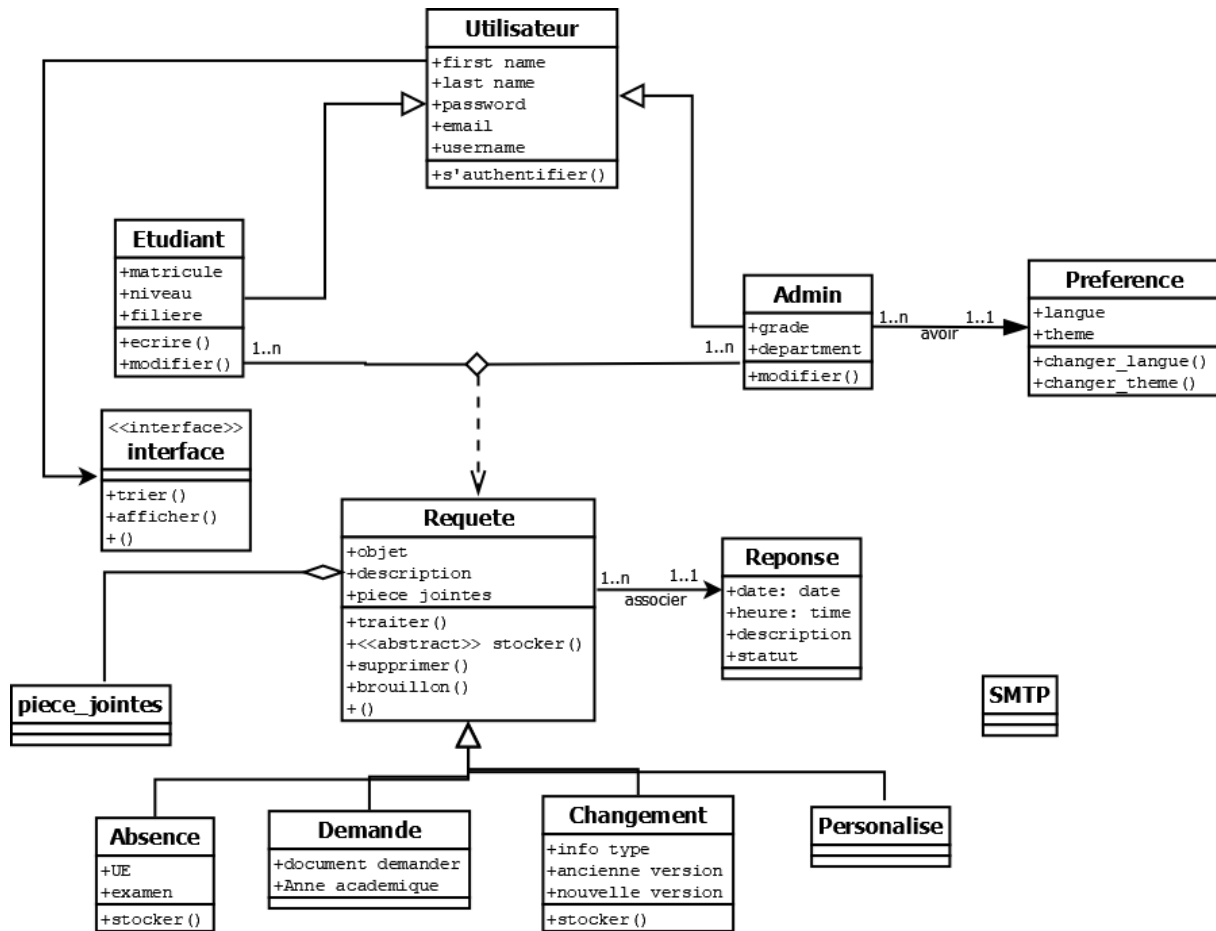
Le diagramme ci-dessus illustre l'interaction entre les différents niveaux du système. En un premier temps, l'utilisateur via le NIV1 envoie une requête au NIV 2 qui interroge le NIV 3 pour fournir une réponse à l'utilisateur. Pour ce faire, le NIV 3 génère une réponse, la renvoie au NIV 2 qui traite la réponse et la présente à l'utilisateur.

- **Conception détaillée**

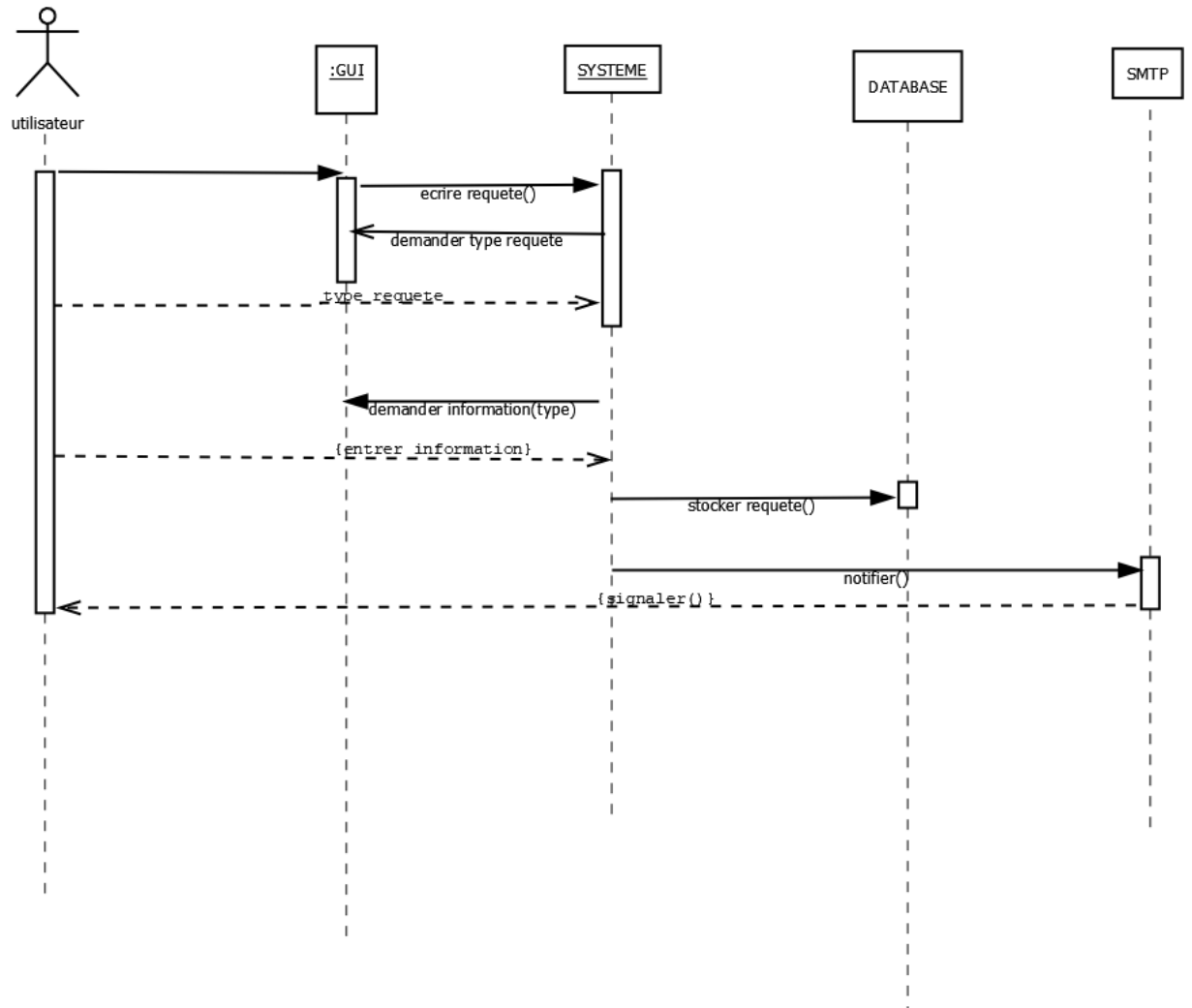
- **Diagramme de cas d'utilisation**



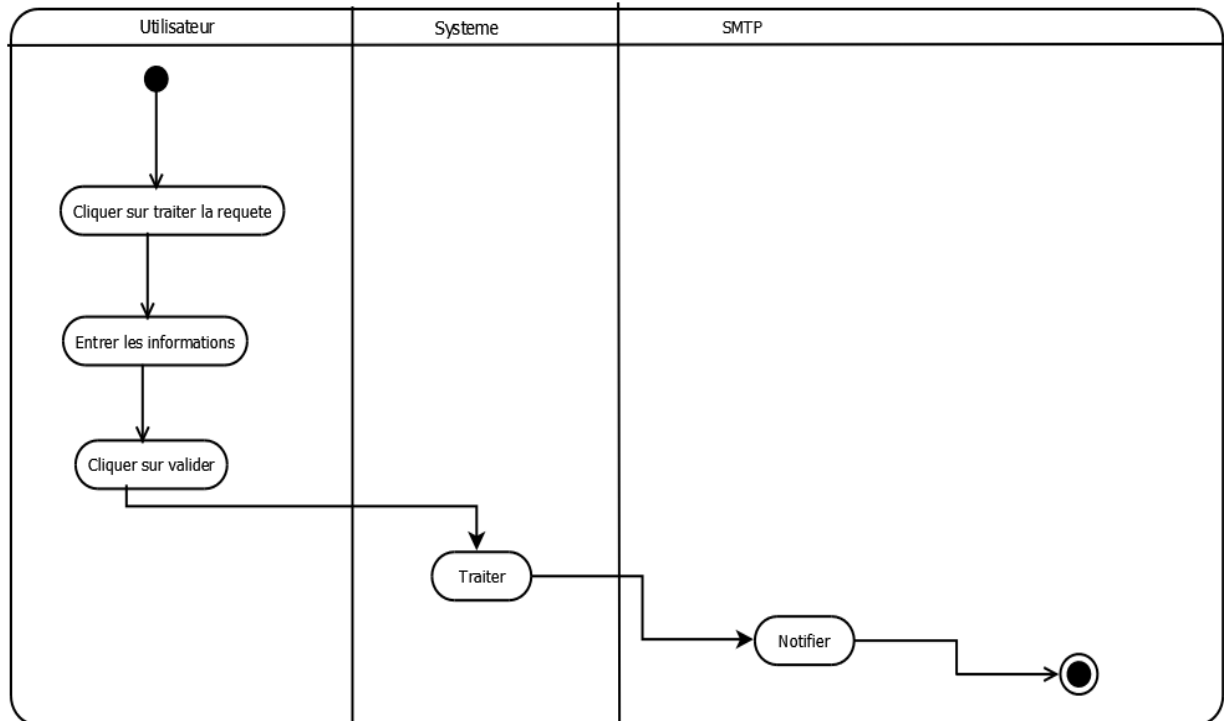
- Diagramme de classe



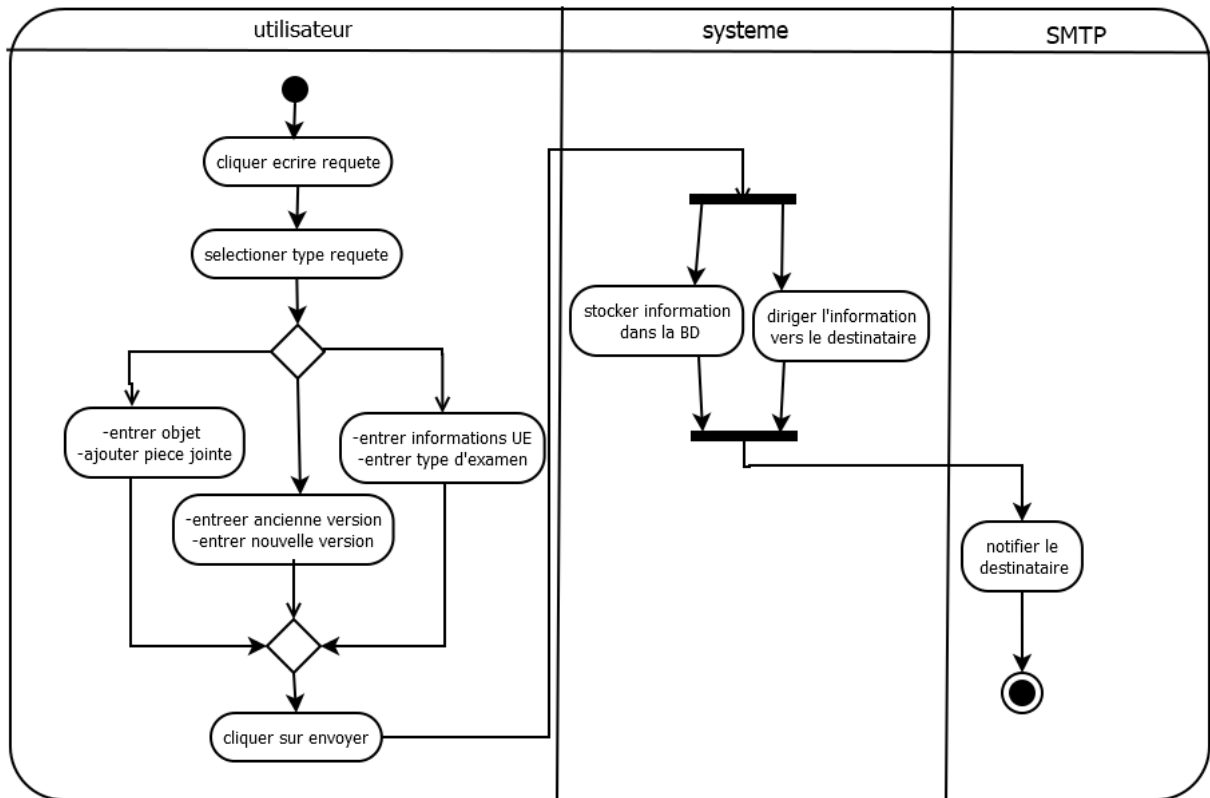
- Diagramme de sequence



- Diagrammes d'activité
 - Traiter la Requête



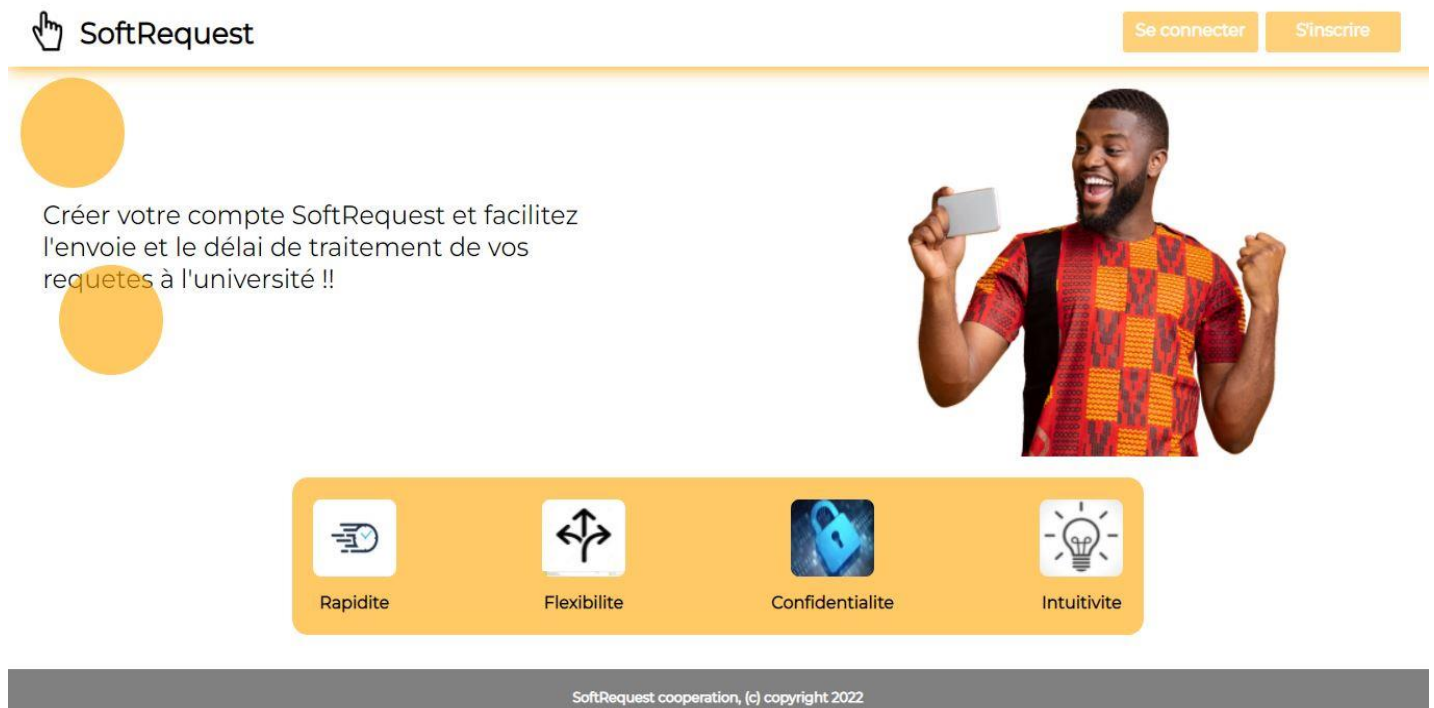
➤ Ecrire une requête



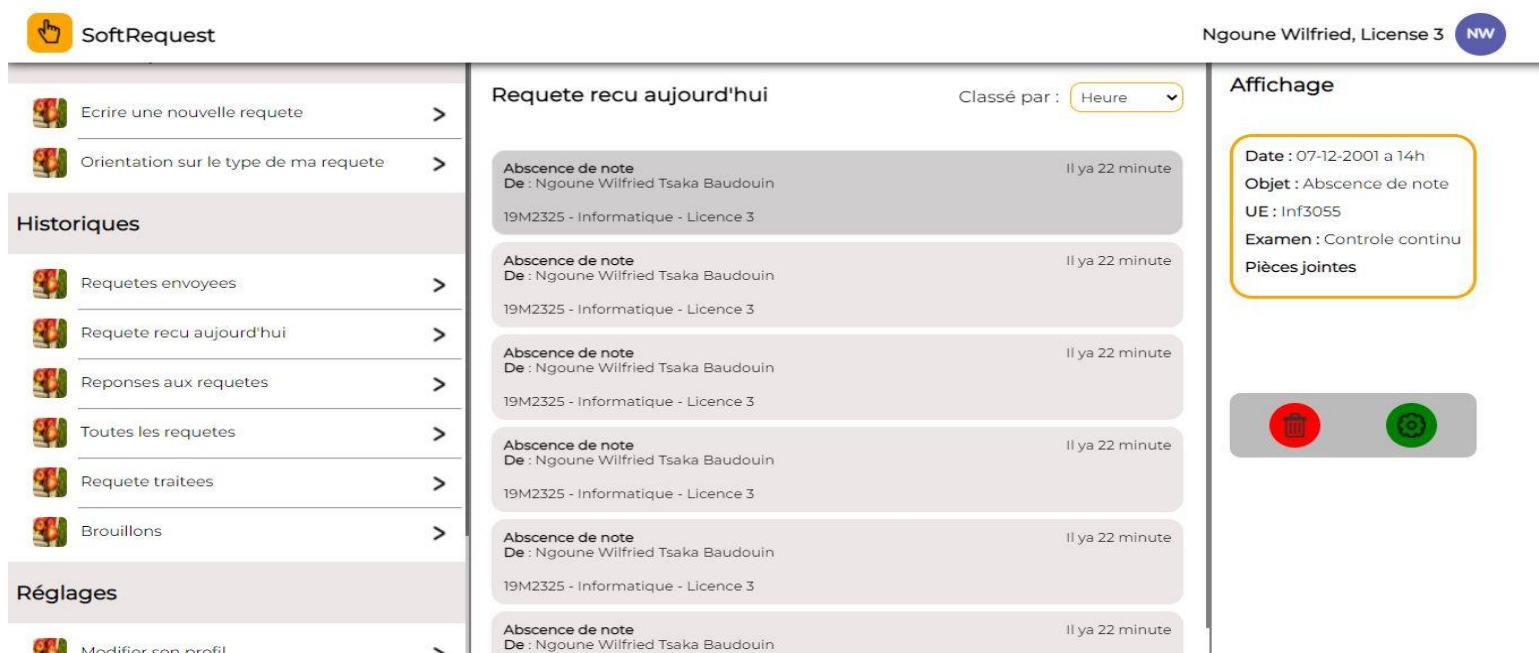
- **Interfaces(IHM)**

Ci-dessous sont deux exemplaires de nos interfaces : la page d'accueil et le dashboard.

➤ **Page d'accueil**



➤ **Dashbord**



4) Implémentation

- **Intégration**

Le backend et le frontend sont liés via un client http nommé axios et donc pour avoir les données venant de la base de données, il suffit de faire une requête axios en précisant le lien du backend.

5) Déploiement

Le projet est construit de deux applications différentes : le frontend et le backend.

- **Déploiement du frontend**

Le frontend a été déployé sur Heroku et le déploiement est basé sur git.

- **Déploiement du backend**

Le backend a été déployé sur Heroku et le déploiement est de même basé sur git et donc, mettre à jour le projet en ligne consiste à faire un git push.