

Projet de session

Etienne Donneger , Anass Abouzakaria , Jean-Baptiste Edouard

Cadre du projet

Description

Contexte	Un journal scientifique demande la production d'une solution en Java, qui va permettre de gérer le processus de soumission et la publication d'articles scientifiques.
Problème à résoudre	<p>Il faut réaliser la conception ainsi que l'implémentation du projet afin de traiter les besoins du journal scientifique. Pour soumettre un article, ce dernier doit passer par une suite d'étapes :</p> <p>Après avoir soumis un article, on passe à la phase de relecture par trois évaluateurs afin de décider si l'article sera accepté ou refusé. Les articles acceptés peuvent être sans ou avec commentaires, les articles acceptés sans commentaire et avec commentaires mineurs passent à la phase 'Camera Ready'. Les articles acceptés avec commentaires majeurs doivent repasser par la phase de relecture.</p> <p>Si des articles aboutissent à la phase 'Camera Ready', cela signifie que les articles sont prêts à être publiés, ainsi les auteurs doivent soumettre une version finale pour la publication en suivant les directives envoyées via un mail de notification d'acceptation, et doivent aussi respecter l'échéance précisée.</p> <p>Il faut bien évidemment prendre en compte un système d'authentification, afin de procurer aux auteurs la possibilité de consulter l'état d'articles, de modifier les informations concernant leurs profiles ou leurs articles.</p> <p>On note aussi qu'il faut envisager un processus de paiement des frais de publication pour le futur proche du projet, où le journal deviendrait payant.</p>

Portée	<ul style="list-style-type: none"> • <i>Objectifs généraux de l'application à développer :</i> <ul style="list-style-type: none"> ○ Gestion de soumission d'articles scientifiques ○ Gestion de relecture d'articles scientifiques ○ Gestion de validation d'articles scientifiques ○ Implémentation d'un système d'authentification / inscription ○ Prévion d'un système de paiement pour les frais de publication d'articles • <i>Livrable :</i> <ul style="list-style-type: none"> ○ Solution en Java permettant de répondre aux objectifs cités précédemment.
---------------	---

Répartition des tâches

Notre équipe se compose de trois personnes, chacune aura des tâches spécifiques à réaliser.

Les tâches ont été réparties comme représenté dans le tableau suivant :

Anass Abouzakaria	<ul style="list-style-type: none"> • Définition du cadre du projet • Identification des besoins non fonctionnels • Diagramme de classe • Définition du prototype illustrant l'interaction attendu avec le logiciel • Présentation des outils de technologie • Réalisation et démo du logiciel • Diagramme de classe
Etienne Donneger	<ul style="list-style-type: none"> • Définition du glossaire • Identification du modèle de domaine • Identification des risques • Définition du prototype illustrant l'interaction attendu avec le logiciel • Présentation des outils de technologie • Diagramme de séquence • Diagramme de cas d'utilisation
Jean-Baptiste Edouard	<ul style="list-style-type: none"> • Identification des besoins fonctionnels • Diagramme de classe • Justification de la conception • Justification des choix des outils • Présentation des outils de technologie • Diagramme de composants • Diagramme objet-relationnel

Priorisation des cas d'utilisations

Échelle de mesure utilisée : (Élevée > Normale > Faible)

Cas d'utilisation	Priorité
UC-01 : S'authentifier	Élevée
UC-02 : Soumettre un article	Élevée
UC-03 : Affecter un article à un comité de lecture	Élevée
UC-04 : Relire un article	Élevée
UC-05 : Accepter ou rejeter un article	Élevée
UC-06 : Qualifier un commentaire (majeur ou mineur)	Élevée
UC-07 : Envoyer un article à la publication	Élevée
UC-08 : Affecter une nouvelle échéance	Normale
UC-09 : Passer l'article en phase "Camera Ready"	Élevée
UC-10 : Envoyer un mail de changement d'état	Élevée
UC-11 : Écrire un commentaire	Élevée
UC-12 : Consulter les commentaires	Normale
UC-13 : Soumettre une nouvelle version	Élevée
UC-14 : Créer un compte	Élevée
UC-15 : Attribuer une note	Élevée
UC-16 : Consulter l'état de l'article	Normale
UC-17 : Modifier le profil	Faible
UC-18 : Modifier les informations d'un article	Élevée
UC-19 : Payer les frais de publication en ligne	Futur
UC-20 : Payer les frais de publication par virement interac	Futur

Définition du domaine

Glossaire

Article: présenté par le/les auteur(s) pour soumission au journal.

Auteur: auteur d'un article.

Catégorie: caractérise un article pour une soumission.

Comité: groupe de trois évaluateurs chargés de valider (ou invalider) la publication d'un article.

Commentaire: créé par un évaluateur à propos d'un article soumis.

Échéance: précise une date limite pour une soumission.

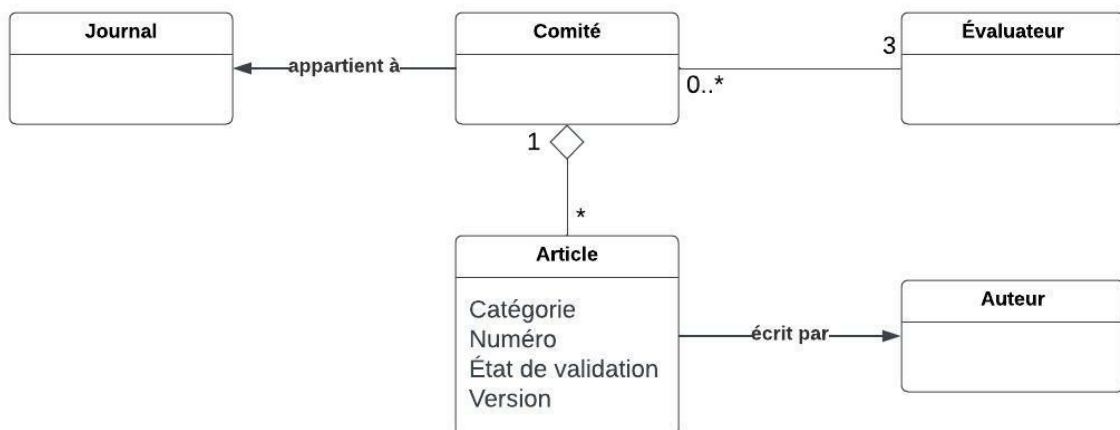
État de validation: caractérise l'état d'un article dans le processus de soumission (refusé avec commentaires, accepté sans commentaires, accepté avec commentaires mineurs ou accepté avec commentaires majeurs).

Évaluateur: chargé d'évaluer une soumission et de modifier son état en plus d'écrire éventuellement des commentaires.

Numéro: le numéro du journal pour lequel les articles sont soumis.

Version: caractérise la version de l'article (soumis, final, etc.).

Modèle du domaine



Besoins fonctionnels

Cas d'utilisations

UC-01 : S'authentifier

- **Acteur:** Utilisateur
- **Description:** Les utilisateurs s'authentifient dans le système.
- **Préconditions :**
- **Postconditions:**
 - L'utilisateur est authentifié.
- **Flux :**
 - **Flux principal**
 - i. L'utilisateur se rend sur le site.
 - ii. L'utilisateur entre son identifiant et son mot de passe.
 - iii. Le site valide les identifiants et envoie l'utilisateur sur sa page d'accueil.
 - **Flux alternatif**
 - i. L'utilisateur se rend sur le site.
 - ii. L'utilisateur se trompe en entrant son identifiant et son mot de passe.
 - iii. Le site affiche un message d'erreur invitant l'utilisateur à réessayer ou à suivre la procédure de récupération de mot de passe.

UC-02 : Soumettre un article

- **Acteur:** Auteur
- **Description:** Les auteurs d'un article soumettent un fichier contenant l'article de recherche qu'ils cherchent à faire publier dans le journal.
- **Préconditions :**
 - Être authentifié.
 - Avoir rempli les champs obligatoires de la soumission.
- **Postconditions:**
 - L'article est soumis au journal en attente d'attribution à un comité de relecture.
- **Flux :**
 - **Flux principal**
 - i. L'auteur remplit tous les champs du formulaire de soumission.
 - ii. L'auteur clique sur soumettre.
 - iii. L'article est soumis au journal en attente d'attribution.
 - **Flux alternatif**
 - i. L'auteur oublie ou fait une erreur dans un des champs du formulaire de soumission.
 - ii. L'auteur clique sur soumettre.
 - iii. Un message d'erreur apparaît invitant l'auteur à corriger son erreur.

UC-03 : Affecter un article à un comité de lecture

- **Acteur:** Système
- **Description:** Un article soumis est affecté à un comité de lecture contenant 3 évaluateurs.
- **Préconditions :**
 - L'article a été soumis par un utilisateur valide.
- **Postconditions:**
 - Un groupe de 3 relecteurs est constitué et affecté à l'article.
- **Flux :**
 - **Flux principal**
 - **Flux alternatif**

UC-04 : Relire un article

- **Acteur:** Évaluateur
- **Description:** Les évaluateurs consultent indépendamment le fichier déposé pour lire l'article et procéder à leur évaluation.
- **Préconditions :**
 - L'évaluateur est authentifié.
 - L'article a été déposé dans un format lisible (pdf, docx etc ...).
- **Postconditions:**
 - L'évaluateur a ouvert l'article et peut le lire.
- **Flux :**
 - **Flux principal**
 - i. L'évaluateur se rend dans la liste des articles qui lui sont attribués pour relecture.
 - ii. Il clique sur un article et accède à l'interface de relecture.
 - iii. Il peut alors ouvrir le fichier et lire la proposition des auteurs.
 - **Flux alternatif**
 - i. L'évaluateur se rend dans la liste des articles qui lui sont attribués pour relecture.
 - ii. Il clique sur un article et accède à l'interface de relecture.
 - iii. L'article n'est pas dans un format lisible ou est corrompu. Le relecteur le signale en attente d'une nouvelle version lisible.

UC-05 : Accepter ou rejeter un article

- **Acteur:** Système
- **Description:** Après l'attribution des notes de chaque relecteur du comité, elles sont sommées. Si la somme est supérieure à 4 alors l'article est accepté sinon il est rejeté.
- **Préconditions :**
 - Les trois membres du comité de lecture ont attribué une note à l'article.
- **Postconditions:**
 - L'article est soit accepté, soit refusé.
 - La phase de relecture prend fin.
- **Flux :**
 - **Flux principal**
 - i. Les membres du comité de relecture ajoutent leur note un à un.
 - ii. Le cumul des notes est supérieur à 4.
 - iii. L'article est passé en statut accepté.
 - **Flux alternatif**
 - i. Les membres du comité de relecture ajoutent leur note un à un.
 - ii. Le cumul des notes est inférieur à 4.
 - iii. L'article est passé en statut refusé.

UC-06 : Qualifier un commentaire (majeur ou mineur)

- **Acteur:** Évaluateur
- **Description:** Après rédaction de son commentaire, l'évaluateur décide si celui-ci est majeur ou mineur.
- **Préconditions :**
 - L'évaluateur est authentifié.
 - L'évaluateur a rédigé son commentaire.
- **Postconditions:**
 - Le commentaire est considéré comme majeur ou mineur.
- **Flux :**
 - **Flux principal**
 - i. L'évaluateur finit d'écrire son commentaire.
 - ii. Avant de le soumettre, il doit sélectionner une option (majeur ou mineur).
 - iii. Il sélectionne l'option majeure.
 - iv. Le commentaire est enregistré comme commentaire majeur.
 - **Flux alternatif**
 - i. L'évaluateur finit d'écrire son commentaire.
 - ii. Avant de le soumettre, il doit sélectionner une option (majeur ou mineur).
 - iii. Il sélectionne l'option mineure.
 - iv. Le commentaire est enregistré comme commentaire mineur.

UC-07 : Envoyer un article à la publication

- **Acteur:** Administrateur
- **Description:** L'administrateur peut décider d'envoyer un article dans l'état "Camera-ready" à la publication s'il remplit les conditions fixées par le journal.
- **Préconditions :**
 - L'administrateur est authentifié.
 - L'article est en phase caméra ready et les auteurs ont soumis une version finale.
- **Postconditions:**
 - L'article est considéré comme à paraître et sortira dans la prochaine édition du journal.
- **Flux :**
 - **Flux principal**
 - i. L'administrateur consulte une soumission d'article en Camera Ready.
 - ii. Il vérifie que les consignes de publication ont été suivies.
 - iii. Il clique sur envoyer à la publication.
 - **Flux alternatif**
 - i. L'administrateur consulte une soumission d'article en Camera Ready.
 - ii. Il vérifie que les consignes de publication ont été suivies.
 - iii. Il repère une erreur dans le suivi des instructions.
 - iv. Il notifie les auteurs afin qu'ils corrigent l'erreur.

UC-08 : Affecter une nouvelle échéance

- **Acteur:** Système
- **Description:** Le système affecte une nouvelle date limite pour la soumission d'une version de l'article.
- **Préconditions :**
 - L'article est dans un état qui nécessite une action avant une date limite (Camera Ready par exemple).
- **Postconditions:**
 - L'article a désormais une date limite pour réaliser la prochaine action.
- **Flux :**
 - **Flux principal**
 - i. L'article passe dans un état qui nécessite une date limite.
 - ii. Le système consulte les délais prévus par le processus métier (exemple: 1 mois dans le cas du dépôt d'une version finale).
 - iii. Le système attribue la date limite correspondante.

UC-09 : Passer l'article en phase "Camera Ready"

- **Acteur:** Système
- **Description:** Le système marque l'article comme "Camera Ready" s'il est accepté sans commentaires ou avec commentaires mineurs.
- **Préconditions :**
 - L'article est accepté.
 - L'article n'a que des commentaires mineurs ou aucun commentaire.
- **Postconditions:**
 - L'article est dans la phase Camera Ready.
 - Les instructions de publications sont communiquées aux auteurs.
- **Flux :**
 - **Flux principal**
 - i. L'article vient d'être accepté et n'a aucun commentaire ou simplement des commentaires mineurs.
 - ii. L'état de l'article change en Camera Ready.
 - iii. Les instructions de publications (gabarit, autorisation de publications, etc.) sont envoyées.

UC-10 : Envoyer un mail de changement d'état

- **Acteur:** Client mail
- **Description:** Le système envoie un mail aux auteurs les informant du changement d'état de leur article et contenant les instructions correspondant à la suite des événements.
- **Préconditions :**
 - L'article vient de changer d'état.
 - L'adresse mail renseignée dans le profil de l'auteur responsable est valide.
- **Postconditions:**
 - Un mail notifiant du changement d'état et des instructions à suivre parvient à l'adresse mail indiquée.
- **Flux :**
 - **Flux principal**
 - i. Un changement d'état de l'article survient.
 - ii. Le mail de notification est généré avec les instructions correspondantes.
 - iii. Le mail est envoyé à l'adresse mail de contact pour l'article.

UC-11 : Écrire un commentaire

- **Acteur:** Évaluateur
- **Description:** L'évaluateur ajoute un commentaire à l'article.
- **Préconditions :**
 - L'évaluateur est authentifié.
- **Postconditions:**
 - Le commentaire de l'auteur est enregistré en relation avec l'article.
- **Flux :**
 - **Flux principal**
 - i. L'évaluateur a fini de relire l'article.
 - ii. Il clique dans le champ de commentaire.
 - iii. Après avoir rédigé son commentaire, il clique sur enregistrer.

UC-12 : Consulter les commentaires

- **Acteur:** Auteur
- **Description:** Les auteurs consultent les commentaires laissés sur une version de l'article par les évaluateurs.
- **Préconditions :**
 - L'auteur est authentifié.
- **Postconditions:**
 - L'auteur peut lire les commentaires des évaluateurs.
- **Flux :**
 - **Flux principal**
 - i. L'auteur consulte ses soumissions et clique sur l'une d'entre elles qui a été relue.
 - ii. Les commentaires sont affichés en dessous des informations de l'article et l'auteur peut en prendre connaissance.

UC-13 : Soumettre une nouvelle version

- **Acteur:** Auteur
- **Description:** Les auteurs soumettent une nouvelle version de l'article tenant compte des commentaires des évaluateurs pour relecture par le comité.
- **Préconditions :**
 - L'auteur est authentifié.
 - L'auteur est invité à soumettre une nouvelle version.
- **Postconditions:**
 - Une nouvelle version est disponible pour le comité de relecture. La soumission passe en état de relecture.
- **Flux :**
 - **Flux principal**
 - i. L'auteur se rend sur la page de la soumission.
 - ii. Il ajoute un nouveau fichier contenant la nouvelle version de l'article.
 - iii. Il clique sur soumettre.
 - iv. La nouvelle version est envoyée.

UC-14 : Créer un compte

- **Acteur:** Utilisateur
- **Description:** Les utilisateurs ne disposant pas d'un compte en créent un.
- **Préconditions :**
 - L'auteur ne dispose pas déjà d'un compte.
- **Postconditions:**
 - L'auteur dispose maintenant d'un compte actif sur lequel il peut se connecter.
- **Flux :**
 - **Flux principal**
 - i. L'utilisateur se rend sur le site du journal.
 - ii. Il clique sur créer un compte.
 - iii. Il remplit le formulaire de création de comptes avec des informations valides.
 - iv. Il soumet le compte.
 - v. Le compte est créé et l'utilisateur peut s'y connecter.
 - **Flux alternatif**
 - i. L'utilisateur se rend sur le site du journal.
 - ii. Il clique sur créer un compte.
 - iii. Il remplit le formulaire de création de comptes avec des informations valides.
 - iv. Le système détecte que les informations sont erronées ou correspondent à un compte existant.
 - v. Le système invite l'utilisateur à corriger son erreur ou à se connecter au compte existant.

UC-15 : Attribuer une note

- **Acteur:** Évaluateur
- **Description:** Un évaluateur attribue une note entre 0 et 3 à un article.
- **Préconditions :**
 - L'évaluateur est authentifié.
 - L'évaluateur n'a pas encore attribué de note à cet article.
- **Postconditions:**
 - La note est enregistrée.
- **Flux :**
 - **Flux principal**
 - i. L'auteur lit l'article selon **UC-04**.
 - ii. Il va ensuite entrer la note voulue dans le champ prévu du formulaire de relecture.
 - iii. Il soumet la note en cliquant sur soumettre.
 - **Flux alternatif**
 - i. L'auteur lit l'article selon **UC-04**.
 - ii. Il va ensuite entrer la note voulue dans le champ prévu du formulaire de relecture.
 - iii. La note entrée est d'une valeur invalide ou dans un mauvais format.
 - iv. Le relecteur est prévenu de son erreur et invité à la corriger.

UC-16 : Consulter l'état de l'article

- **Acteur:** Auteur
- **Description:** Les auteurs peuvent se connecter et consulter l'état de leur article.
- **Préconditions :**
 - L'auteur est authentifié.
- **Postconditions:**
 - L'auteur est informé de l'état de l'article.
- **Flux :**
 - **Flux principal**
 - **Flux alternatif**

UC-17 : Modifier le profil

- **Acteur:** Utilisateur
- **Description:** Les utilisateurs peuvent modifier les informations de leur profil.
- **Préconditions :**
 - L'utilisateur est authentifié.
- **Postconditions:**
 - Les informations de l'utilisateur sont mises à jour.
- **Flux :**
 - **Flux principal**
 - i. L'utilisateur se rend sur son profil.
 - ii. Il clique sur modifier.
 - iii. Il accède au formulaire de modification de ses informations.
 - iv. Il remplit le formulaire avec les nouvelles informations et le soumet.
 - v. Le profil est mis à jour avec les nouvelles informations.
 - **Flux alternatif**
 - i. L'utilisateur se rend sur son profil.
 - ii. Il clique sur modifier.
 - iii. Il accède au formulaire de modification de ses informations.
 - iv. Il remplit le formulaire avec les nouvelles informations, mais commet une erreur sur le format des informations.
 - v. L'utilisateur est prévenu par un message de son erreur et est invité à la corriger avant soumission.

UC-18 : Modifier les informations d'un article

- **Acteur:** Auteur
- **Description:** Les auteurs peuvent modifier des informations associées à un article à l'exception du nom du premier auteur.
- **Préconditions :**
 - L'auteur doit être authentifié.
 - L'auteur doit être celui qui a créé cette soumission.
- **Postconditions:**
 - Les informations de l'article sont mises à jour.
- **Flux :**
 - **Flux principal**
 - i. L'auteur se rend sur la page de sa soumission et clique sur modifier.
 - ii. Le système lui présente alors le formulaire de modification des informations de l'article.
 - iii. L'auteur change les informations nécessaires et soumet le formulaire.
 - iv. Après vérification les modifications sont acceptées et la soumission est mise à jour avec les nouvelles informations.
 - **Flux alternatif**
 - i. L'auteur se rend sur la page de sa soumission et clique sur modifier.
 - ii. Le système lui présente alors le formulaire de modification des informations de l'article.
 - iii. L'auteur change les informations nécessaires et soumet le formulaire.
 - iv. Si l'auteur a commis une erreur dans le format des informations, le système le lui indique et l'invite à la corriger avant soumissions.

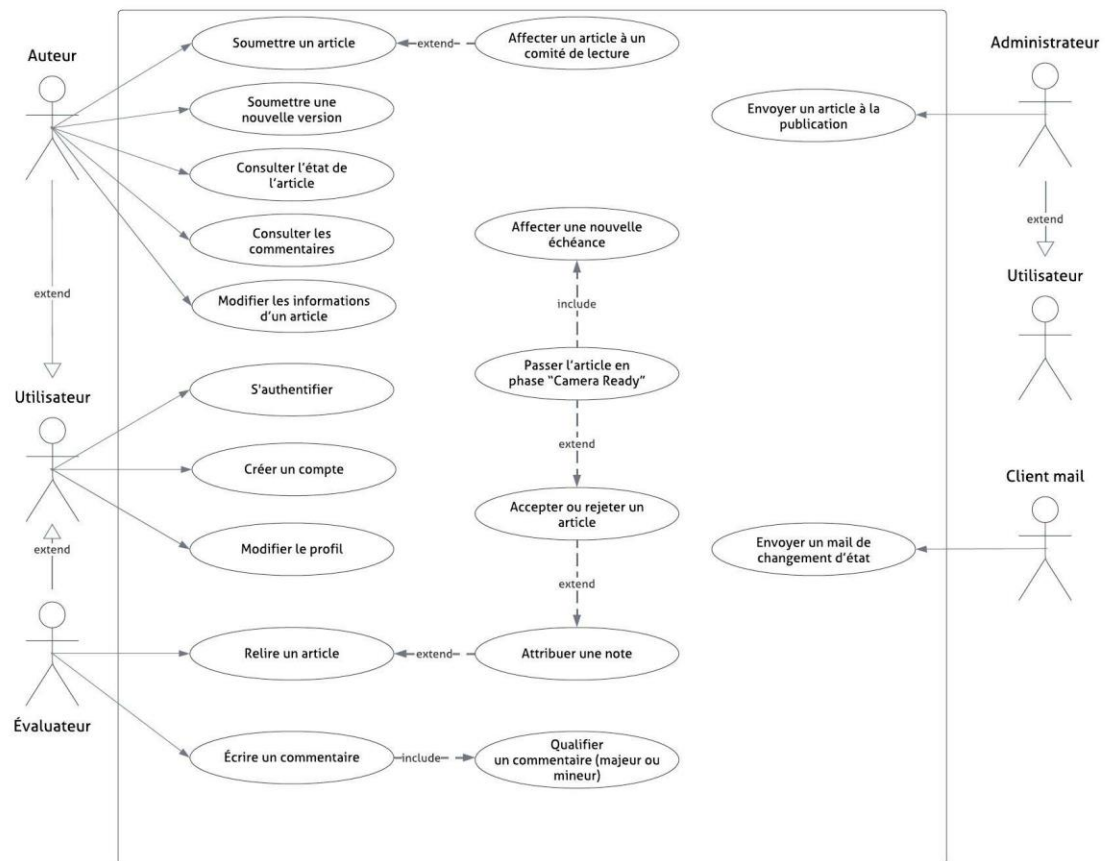
(à prévoir) UC-19 : Payer les frais de publication en ligne

- **Acteur:** Auteur
- **Description:** Les auteurs paient les frais de publication en ligne après acceptation de leur article
- **Préconditions :**
 - Être authentifié.
 - L'article a été accepté et est prêt à continuer vers la publication.
- **Postconditions:**
- **Flux :**
 - **Flux principal**
 - i. L'auteur est dirigé vers une interface de paiement sécurisé.
 - ii. Il entre ses informations de paiement.
 - iii. Le service vérifie la validité du paiement.
 - iv. Une facture est émise et communiquée à l'auteur.
 - **Flux alternatif**
 - i. L'auteur est dirigé vers une interface de paiement sécurisé.
 - ii. Il entre ses informations de paiement.
 - iii. Le service vérifie la validité du paiement.
 - iv. Le paiement est rejeté ou les informations entrées sont invalides.

(à prévoir) UC-20 : Payer les frais de publication par virement interac

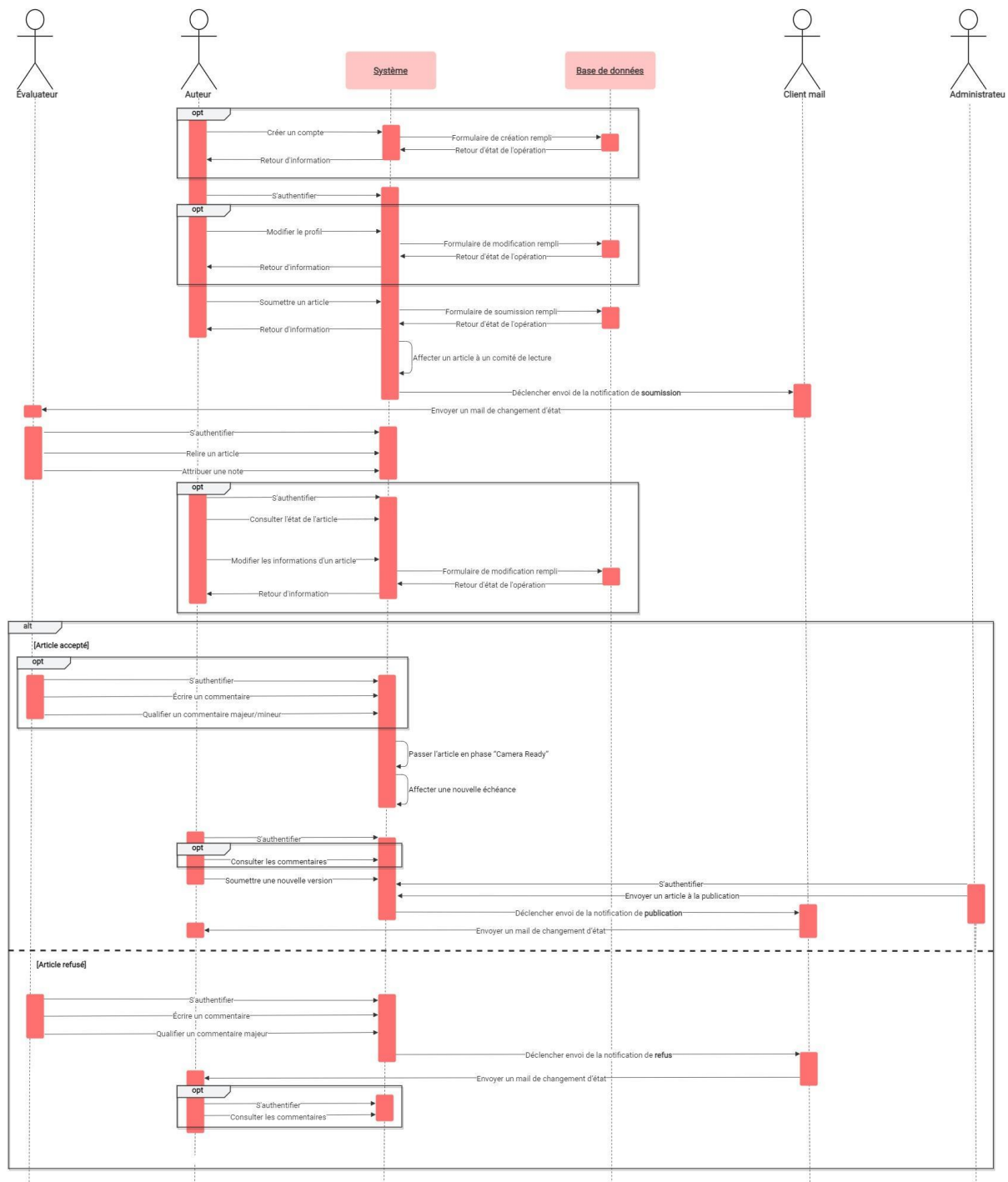
- **Acteur:** Auteur
- **Description:** Les auteurs paient les frais de publication par virement interac après acceptation de leur article.
- **Préconditions :**
- **Postconditions:**
- **Flux :**
 - **Flux principal**
 - **Flux alternatif**

Diagramme de cas d'utilisation



<https://lucid.app/publicSegments/view/c5553d69-8f93-437a-a6c7-87355f153f47>

Diagramme de séquence



[Lien vers le diagramme de séquence](#)

Besoins non fonctionnels

- Facilité d'utilisation.
- Les interfaces utilisateurs doivent être intuitives.
- Le temps de réponse ne doit pas dépasser 200ms.
- Sécurité du système.
- La fiabilité doit être assurée.

- Disponibilité 24/7.
- Facilité de maintenance.
- Facilité de déploiement.
- Facilité de modification des règles pour le système de points.
- Facilité d'évolution pour le système de paiement.

Risques

Deux catégories de risques ont été identifiées pour ce projet: les risques liés à la **conception** et les risques liés au **fonctionnement** du système.

Conception

Identifiant	Risque
Risque #1	Conflit entre membres de l'équipe.
Risque #2	Manque d'implication de la part d'un membre de l'équipe.
Risque #3	Non-respect des délais.
Risque #4	Inaccessibilité des plateformes de partage, travail et collaboration.
Risque #5	Les modifications apportées ne sont pas en accord avec les nouveaux besoins du client.

Fonctionnement

Identifiant	Risque
Risque #1	Un évaluateur est indisponible lors de la phase de relecture (refus, oubli, etc.).
Risque #2	Un article est soumis dans plusieurs catégories différentes.
Risque #3	Un même article est soumis plusieurs fois pour des numéros différents.
Risque #4	Un même article est soumis plusieurs fois pour une soumission.
Risque #5	Un changement de date pour la soumission ou pour la révision par les évaluateurs.
Risque #6	Un désaccord survient entre un auteur et un évaluateur.
Risque #7	Le service d'authentification est inaccessible.
Risque #8	Le compte d'un auteur est compromis.
Risque #9	Le service de paiement est inaccessible ou compromis.

Conception

Diagramme de classe

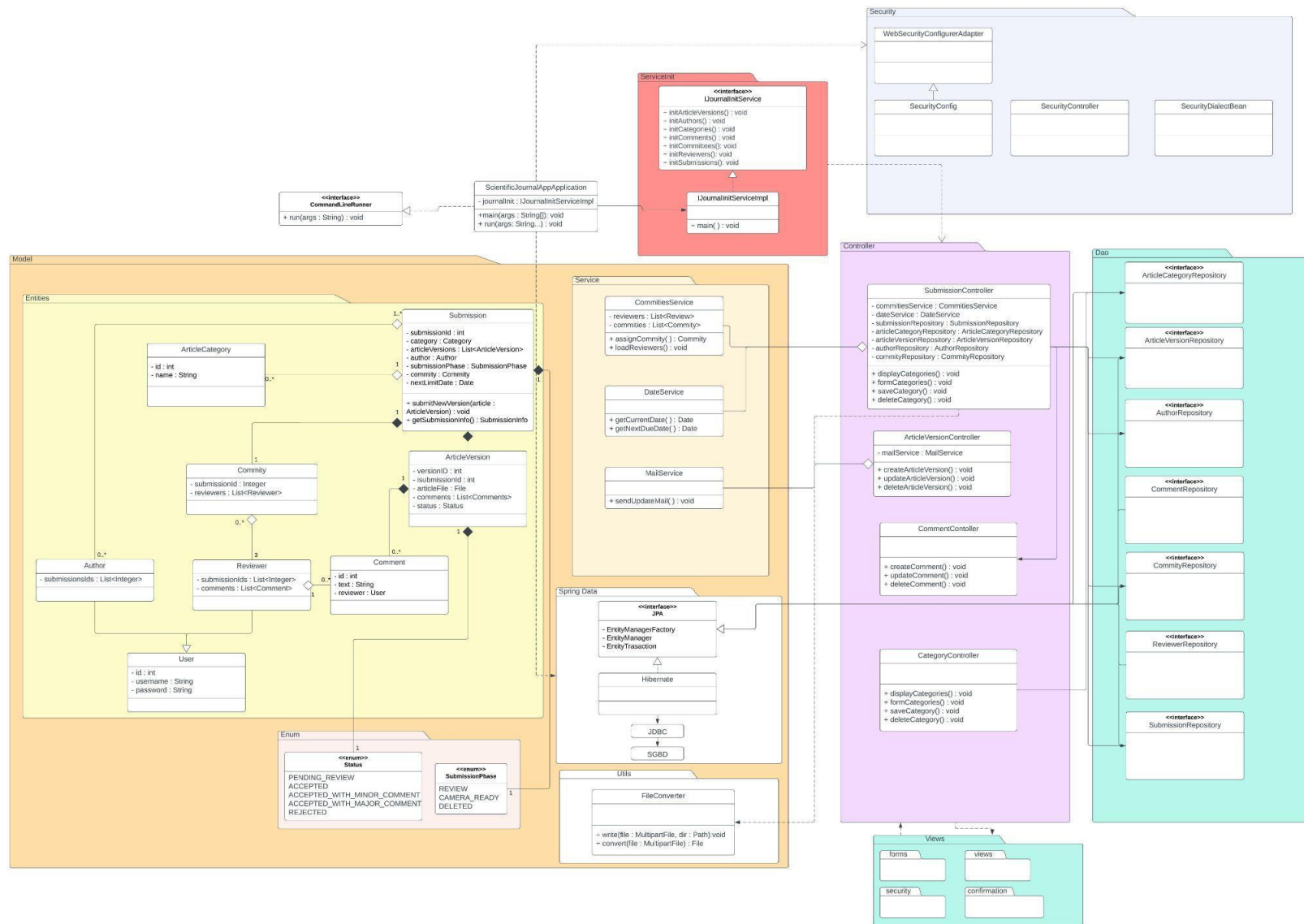
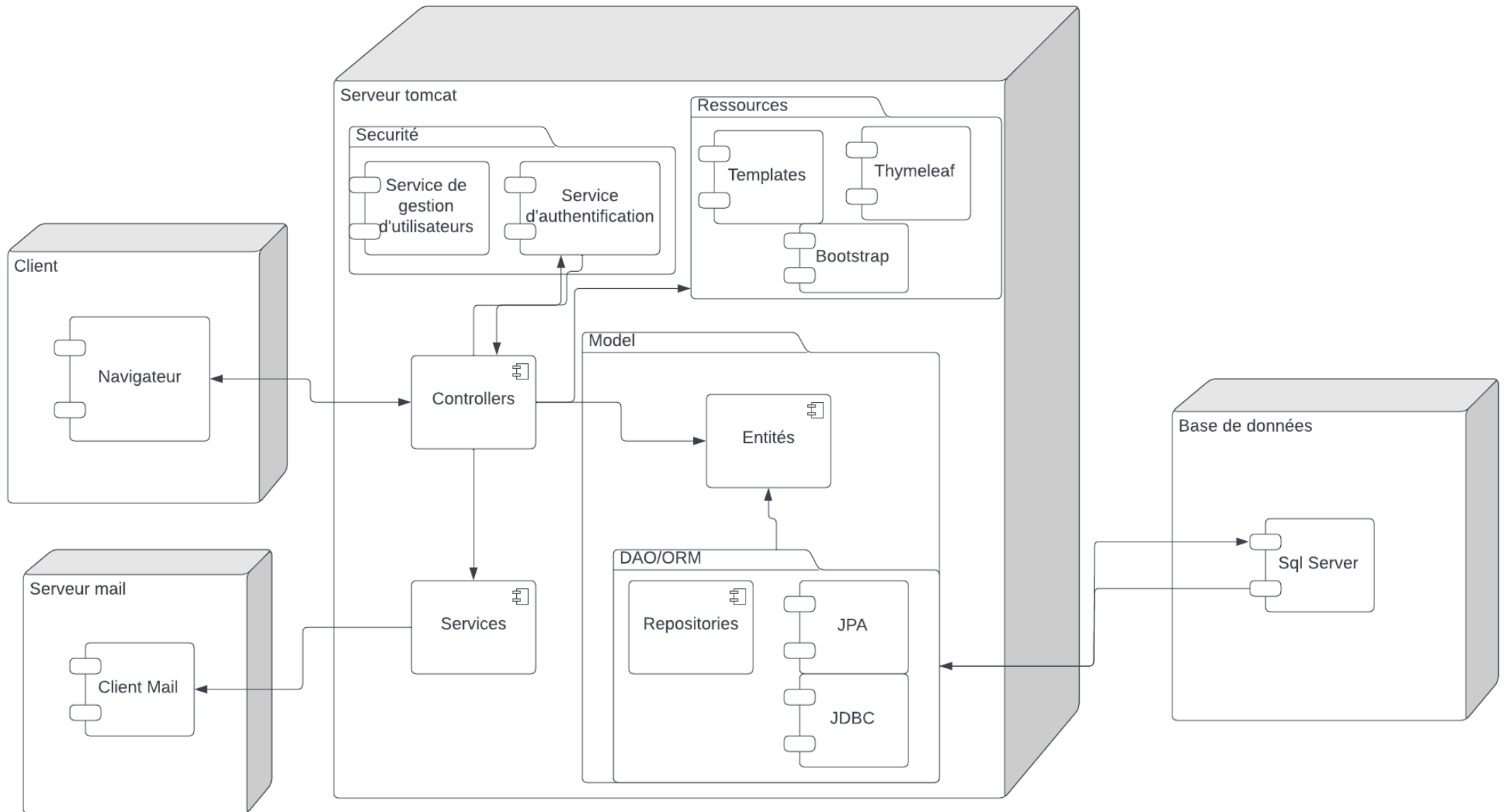
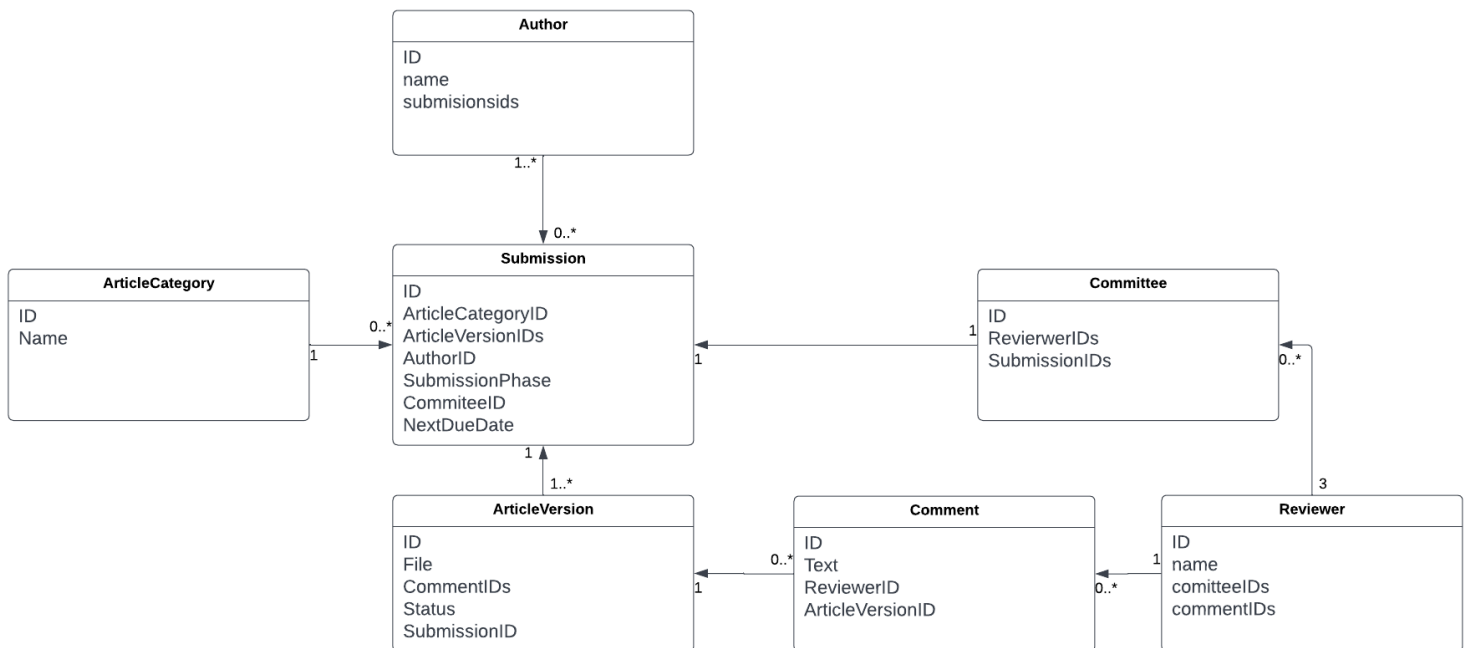
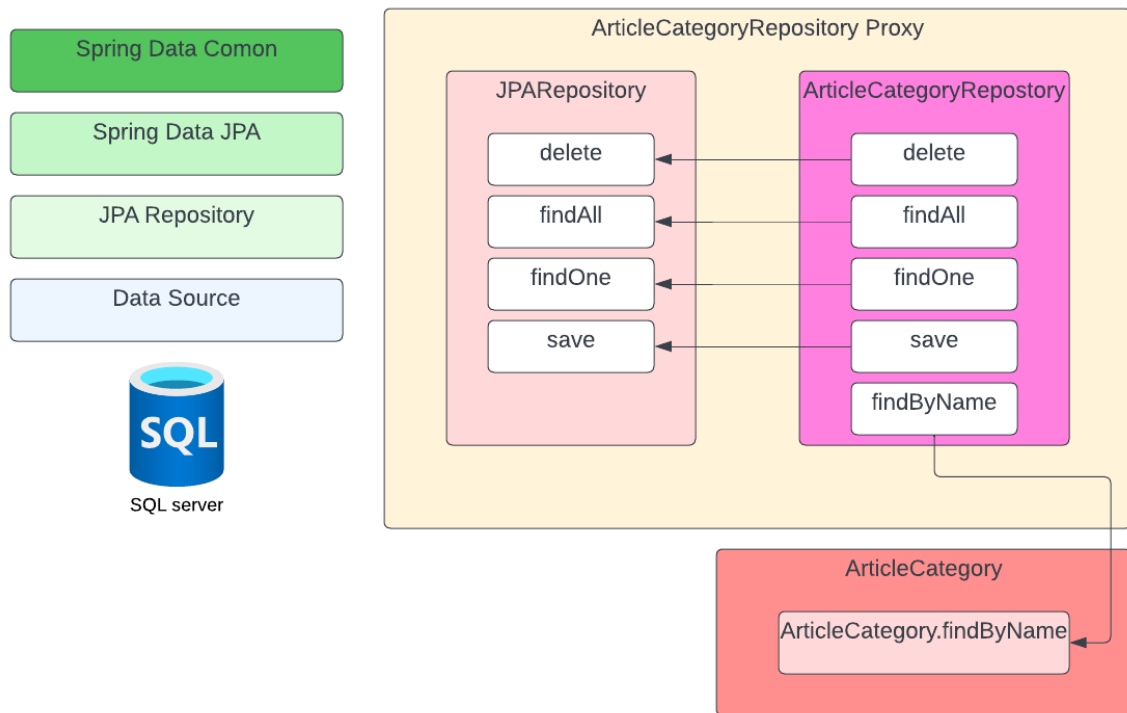


Diagramme de composants



Modèle de persistance objet-relationnel



Justification de la conception

Notre conception se base sur le framework Java Spring et plus précisément Spring Boot. Cela nous offre une architecture classique **Model-View-Controller** avec toute la puissance des outils du framework.

Concernant notre modèle et notre gestion des données, nous nous reposons sur l'**Object Relational Mapping** du framework Spring issu du composant Spring Data. Tout d'abord du point de vue de l'architecture, nous proposons d'utiliser SQL Server en raison de l'expérience de l'équipe.

D'autre part le Modèle implémente le framework Spring Data pour utiliser la puissance de l'ORM. Deux packages soutiennent cette implémentation.

Le package « *entités* » contient les classes et l'architecture des objets stockés en base de données. Nous avons fait plusieurs choix de conception à ce niveau:

- Tout d'abord la classe *User* se décline en 2 implémentations *Author* et *Reviewer*. L'idée est que dans la communauté scientifique, un auteur peut tout à fait être relecteur pour d'autres articles donc quand un utilisateur s'authentifie ou qu'on va chercher à instancier un utilisateur pour lui appliquer de la logique, il sera instancié soit en tant qu'auteur soit en tant que relecteur en fonction du contexte.
- Nous avons ensuite fait un choix de représentation pour la notion d'article. La classe *Submission* représente toute la session de soumission avec les versions successives de l'article et un historique des actions effectuées. La classe *ArticleVersion*, elle, représente une version soumise de l'article avec les commentaires qui sont propres à cette version. Cette distinction nous permet de garder une trace de toutes les étapes du processus de soumission dans un objet qui contient les éléments communs de ces étapes.
- Les différentes entités contiennent leurs attributs respectifs qui correspondent aux informations qui les caractérisent en base de données. De plus, afin de créer les relations nécessaires, nous utilisons le *pattern décorateur* par l'intermédiaire de Spring. On peut ainsi caractériser des relations « One to One », « One to Many », etc. Cela reflète les relations entre les tables présentes dans la base. Cette abstraction des données nous donne une grande flexibilité dans l'ajout de relations et de caractérisation dans les données. Cela permet aussi une grande lisibilité dans la manipulation des données, on a une correspondance immédiate entre le code et la base de données.

Le package « *dao* » contient les abstractions des répertoires qui sont responsables d'exécuter les requêtes à la base de données. Cette couche permet de détacher notre code de l'exécution réelle des requêtes. Les détails des requêtes et l'adaptation à chaque service de base de données sont gérés par le framework sous la forme d'une boîte noire. Cela renforce les bonnes pratiques et ajoute de la flexibilité à notre code.

Le package « *enum* » contient deux énumérations qui représentent le statut de l'article (s'il a été accepté ou non) et la phase actuelle de la session de soumission (relecture ou « Camera Ready »). Cela sert à contraindre les états disponibles tout en permettant l'extension.

Le deuxième grande partie de notre architecture est le package « *controller* ». Les contrôleurs de notre application tirent eux aussi parti de la puissance des décorateurs dans le but d'implémenter le *routing* vers les bonnes méthodes des contrôleurs. Cette couche de l'application lie tous les différents éléments du modèle et de la vue.

Pour la « *View* », le package ressources contient les pages, styles, formulaires, images, etc. nécessaires à l'interface de l'application. Ce package tire pleinement parti du concept de "template" supporté par Thymeleaf (inclus dans Spring) qui permet d'injecter du code dans du html afin de compiler et générer la page html avec les bons éléments au moment de la requête.

Enfin, fonctionnant en parallèle de notre architecture principale se trouvent des services. Ces services servent à isoler des tâches bien spécifiques qui sont périphériques aux scénarios principaux de l'application. La sécurité tout d'abord qui contient à la fois la gestion de l'authentification et la gestion des utilisateurs. Parmi les autres services on peut trouver la gestion des notifications et des mails.

Outils et technologies

Outils de collaboration

- **Google Docs**: un logiciel de traitement de texte basé sur le Web.
- **LucidChart**: une plateforme de collaboration en ligne, basée sur le cloud, permettant la création de diagrammes et la visualisation de données, et autres schémas conceptuels.

Outils de conception

- **PlantUML**: un logiciel libre permettant de générer des diagrammes depuis un langage textuel simple.
- **Figma**: un éditeur de graphiques vectoriels et un outil de prototypage.

Outils de développement

- **Java**: un langage de programmation orienté objet.
- **Github**: un service web d'hébergement et de gestion de développement de logiciels.
- **Spring**: Un cadre basé sur Java Enterprise Edition (J2EE), permet de faciliter la configuration du projet, surtout la phase du développement et la maintenance.

Justification du choix des outils

Google Docs et **LucidChart** sont des outils de choix pour la rédaction, la création et le partage de documents et diagrammes permettant à chacun de travailler efficacement, à distance, et à son rythme. L'intégration du plugin LucidChart aux documents Google est aussi un atout majeur.

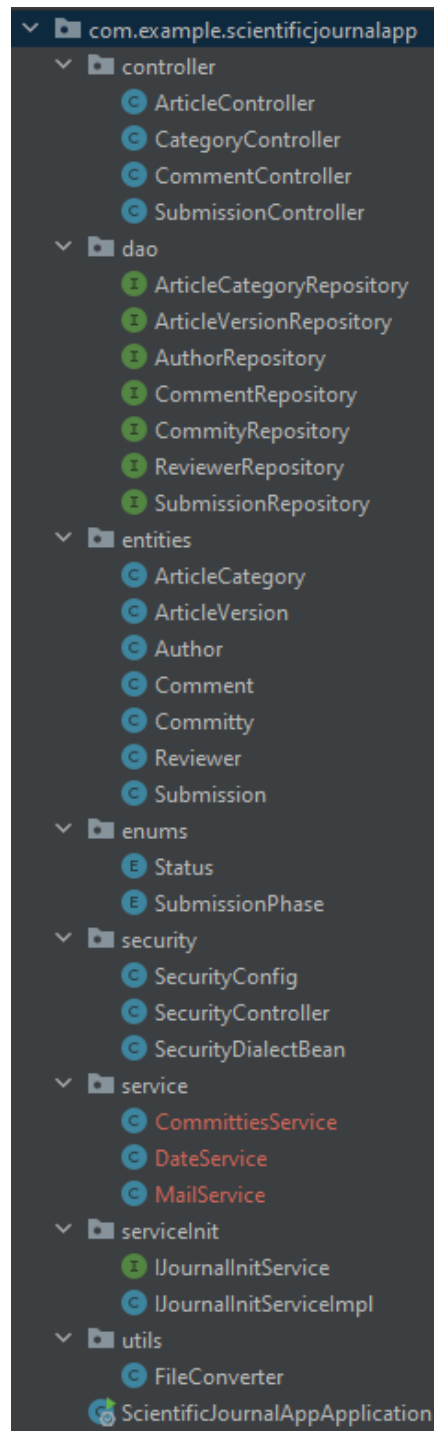
Pour la conception, **PlantUML** permet de générer rapidement des diagrammes UML dans une syntaxe proche de ce qui sera réalisé lors du développement. **Figma** est une référence dans le domaine du prototypage permettant de créer des écrans et leurs interactions de manière entièrement visuelle. C'est aussi une plateforme qui encourage le travail collaboratif en étant plusieurs à pouvoir éditer le même document et fournir des annotations sur les esquisses développées.

Java est le langage du cours et populaire pour la réalisation d'applications en orienté objet. **Github** est aussi la référence en termes d'hébergement de code source à l'aide de son logiciel de versioning **Git**.


Enfin, **Spring** a été utilisé essentiellement pour pouvoir bénéficier de l'IoC (Inversion of Control offerte par l'architecture de Spring), de gestion interne des couches MVC par le module Spring MVC, du mapping objet relationnel (ORM) offert par le module Spring Data, et finalement de la couche sécurité procurée par Spring Security.

Réalisation

Structure de l'application (packages)



Quelques captures d'écran des interfaces de l'application :



Authentification

Email

Password

SE CONNECTER


Scientific journal management

HomeCategoriesArticles SubmittedEdit ProfileLogout

Article Categories List	
Id	Name
1	Brief Report
2	Case Report
3	Communication
4	Conference Report
5	Editorial
6	Essay
7	Review
8	Protocol

Scientific journal management

HomeCategoriesArticles SubmittedEdit ProfileLogout

Article Submissions List							
Add Submission +							
Id	Article Category Name	Version Id of Article	Author Lastname	Submission Phase	Id of committy	Date	
1	Case Report	1	Baptiste	REVIEW	1	2022-12-19 20:37:38.199	
2	Case Report	2	Donneger	REVIEW	2	2022-12-19 20:37:38.208	