

Description textuelles

Use Case 1 : Planification des Examens

Acteurs

- **Professeur** (acteur initiateur)
- **Administrateur** (acteur validateur)
- **Système de planification (Microservice)** (acteur technique)

Préconditions

- Le professeur est authentifié et dispose des droits de planification.
- Le calendrier universitaire est publié et à jour.

Postconditions

- Le planning est enregistré dans la base (MongoDB).
- Des notifications sont envoyées aux étudiants concernés.

Flux Principal Détaillé

Responsable	Système	Serveur
1. Initiation de la planification: Le professeur démarre la procédure	1.1 Afficher l'interface dédiée à la planification avec les options de session (normale ou rattrapage) et de matière 1.2 Permettre la sélection d'une matière (ex. « INF-302 »)	1.1.1 Vérifier que le professeur est authentifié et dispose des droits de planification 1.1.2 Charger le calendrier universitaire (vérifier qu'il est publié et à jour)
2. Génération de la liste des étudiants: (Processus automatisé)	2.1 Identifier le type de session sélectionné 2.2 Appliquer la règle de filtrage : – Si session normale, récupérer tous les étudiants inscrits – Si rattrapage, ne retenir que ceux avec note < 10	2.1.1 Exécuter une requête sur la base d'étudiants (MongoDB ou autre référentiel) 2.1.2 Filtrer les résultats en fonction du critère de note pour la session de rattrapage
3. Attribution des ressources: Le professeur intervient pour le choix de la salle	3.1 Afficher la liste des salles disponibles selon la capacité requise 3.2 Permettre la sélection manuelle d'une salle	3.1.1 Appeler l'API Calendrier pour vérifier la disponibilité des salles en temps réel 3.1.2 Récupérer l'état de

		réserve et transmettre l'information au système
4. Soumission pour validation: Le professeur confirme le planning	4.1 Présenter un récapitulatif complet du planning pour vérification 4.2 Proposer l'envoi de la demande de validation à l'administrateur	4.1.1 Transmettre le planning en file d'attente (via, par exemple, RabbitMQ) 4.1.2 Enregistrer temporairement le planning en attente de validation
5. Validation administrative: L'administrateur prend la décision finale	5.1 Afficher les détails du planning pour validation (ancienne et nouvelle information si applicable) 5.2 Permettre à l'administrateur de sélectionner « Approuver » ou « Refuser »	5.1.1 En cas d'approbation, enregistrer définitivement le planning dans MongoDB 5.1.2 Déclencher l'envoi des notifications aux étudiants concernés

Flux Alternatifs et Exceptions

Exception	Système	Serveur
A. Conflit de salle	A.1 : Afficher le message d'indisponibilité (ex. « La salle S-120 est réservée le 15/12/2024 à 14h »). A.2 : Proposer la salle alternative (ex. S-121).	A.1.1 : Vérifier l'indisponibilité via l'API Calendrier. A.1.2 : Consulter la base pour confirmer la réservation. A.2.1 : Vérifier la disponibilité de S-121. A.2.2 : Enregistrer l'événement et générer une alerte.
B. Refus administratif	B.1 : Afficher le message de refus avec le motif (ex. « Budget insuffisant pour 2 surveillants »). B.2 : Indiquer l'annulation de la planification.	B.1.1 : Annuler l'enregistrement du planning et consigner l'événement dans le log d'audit. B.1.2 : Mettre à jour le statut de la demande dans la base.
C. Données incohérentes ou techniques	C.1 : Afficher un message d'erreur adapté (ex. « Quota d'examens dépassé » ou « Données corrompues »). C.2 : Proposer de réessayer ou contacter le support.	C.1.1 : Déclencher un rollback automatique. C.1.2 : Envoyer une alerte aux développeurs via le système de monitoring. C.2.1 : Vérifier l'intégrité des données. C.2.2 : Consigner l'incident dans le journal de bord.

Use Case 2 : Saisie des Notes

Acteurs

- **Professeur** (acteur principal)
- **Système de gestion des notes (API REST)** (acteur technique)

Préconditions

- Le professeur est authentifié.
- La session d'examen est clôturée (statut « Terminée »).

Postconditions

- Les notes sont stockées dans Elasticsearch pour une recherche optimisée.
- Une notification est envoyée aux étudiants pour informer de la mise à jour.

Flux Principal Détaillé

Responsable	Système	Serveur
1. Accès à la saisie des notes: Le professeur sélectionne la matière et la session (normale ou rattrapage)	1.1 Afficher l'interface de saisie dédiée avec listes déroulantes pour matière et session	1.1.1 Vérifier l'authentification du professeur et prendre en compte que la session d'examen est clôturée
2. Affichage du tableau de saisie: (Lancement automatique)	2.1 Afficher un tableau comportant au moins : – Une colonne « Note » (champ numérique) – Une colonne « Statut » indiquant la validation ou absence de saisie	2.1.1 Charger les données des étudiants concernés 2.1.2 Initialiser le modèle de saisie des notes en provenance de la base
3. Saisie et validation des notes: Le professeur saisit les notes pour chaque étudiant	3.1 Valider en temps réel que chaque note est comprise entre 0 et 203. 2 Afficher immédiatement un message d'erreur en cas d'anomalie	3.1.1 Envoyer la saisie via WebSocket pour sauvegarde automatique toutes les 30 secondes 3.1.2 Exécuter la logique de validation côté serveur
4. Finalisation et transmission: Après saisie, le professeur finalise la procédure	4.1 Compiler l'ensemble des notes saisies et présenter un récapitulatif 4.2 Envoyer une demande de validation à l'administrateur	4.1.1 Transférer les données vers le système de gestion des notes (via API REST) 4.1.2 Enregistrer les notes dans Elasticsearch pour une recherche

		optimisée
--	--	-----------

Flux Alternatifs et Exceptions

Exception	Système	Serveur
A. Note invalide	A.1 : Afficher immédiatement le message d'erreur (« La note doit être un nombre entre 0 et 20 »). A.2 : Surligner le champ concerné.	A.1.1 : Enregistrer l'erreur dans le log. A.1.2 : Bloquer la sauvegarde automatique jusqu'à correction de la note.
B. Sauvegarde manuelle	B.1 : Afficher une confirmation de sauvegarde manuelle (ex. « Données sauvegardées à 14:30:22 »). B.2 : Actualiser l'interface pour refléter l'état sauvegardé.	B.1.1 : Recevoir la commande de sauvegarde manuelle. B.1.2 : Enregistrer l'état actuel des notes dans la base.
C. Session expirée ou conflit	C.1 : Afficher le message d'erreur (ex. « La saisie est clôturée depuis le 30/11/2024 ») ou signaler un conflit (si note déjà validée). C.2 : Proposer une procédure de reprise le cas échéant.	C.1.1 : Déclencher la procédure de rollback. C.1.2 : Enregistrer la fin de session dans le log. C.2.1 : Alerter l'administrateur du conflit. C.2.2 : Consigner l'incident pour audit.

Use Case 3 : Modification des Notes

Acteurs

- **Professeur** (acteur initiateur)
- **Administrateur** (acteur validateur)
- **Système de gestion des notes (API REST)** (acteur technique)

Préconditions

- Le professeur est authentifié.
- Une note existe déjà avec le statut « Provisoire » ou « Validée ».

Postconditions

- La note est mise à jour dans la base avec versionning.

- Une notification est envoyée à l'étudiant pour l'informer de la modification.

Flux Principal Détaillé

Responsable	Système	Serveur
1. Recherche de la note: Le professeur accède à l'interface de recherche et saisit le nom ou matricule de l'étudiant	1.1 Afficher l'interface de recherche (par nom ou matricule)	1.1.1 Exécuter une recherche dans la base (Elasticsearch ou équivalent) afin de récupérer la note existante
2. Affichage de la note à modifier: Le professeur consulte la note actuelle	2.1 Afficher la note existante ainsi que son statut (Provisoire ou Validée)	2.1.1 Récupérer les données de la note avec historique (versionning) depuis la base
3. Saisie de la nouvelle note: Le professeur saisit la nouvelle valeur	3.1 Recevoir la nouvelle valeur et actualiser l'interface de saisie	3.1.1 Valider le format numérique et préparer la mise à jour dans la base
4. Enregistrement en statut "En Attente": Processus d'enregistrement de la modification	4.1 Mettre à jour l'interface pour indiquer le statut « En attente »	4.1.1 Enregistrer temporairement la nouvelle note avec son statut dans la base (intégrant le versionning)
5. Notification à l'administrateur: L'administrateur doit être informé pour validation	5.1 Afficher le récapitulatif incluant l'ancienne note, la nouvelle note et le justificatif dans l'interface de validation	5.1.1 Transmettre la notification (via API REST ou Kafka) à l'administrateur pour prise de décision

Flux Alternatifs et Exceptions

Exception	Système	Serveur
A. Refus de l'administrateur	A.1 : Afficher le message de refus (ex. « Modification refusée : Hors délai »).	A.1.1 : Restaurer la note initiale dans la base.
	A.2 : Indiquer la restauration de la note initiale.	A.1.2 : Consigner l'incident dans le log d'audit.
B. Violation de session ou droits	B.1 : Afficher le message d'erreur signalant une modification non autorisée.	B.1.1 : Empêcher la modification et générer une alerte de sécurité.
	B.2 : Bloquer l'opération de modification dans l'interface.	B.1.2 : Enregistrer la tentative d'accès non autorisé dans le log de sécurité.

Use Case 4 : Demande de Relevé de Notes (Étudiant)

Acteurs

- **Étudiant** (acteur initiateur)
- **Administrateur** (acteur valideur)
- **Système de génération de PDF (Microservice)** (acteur technique)

Préconditions

- L'étudiant est authentifié.
- Les notes sont en statut « Validée » (approuvées par l'administrateur).

Postconditions

- Un lien sécurisé (JWT) est généré, valable 48 heures.
- Le relevé de notes est généré au format PDF et stocké sur AWS S

Flux Principal Détaillé

Responsable	Système	Serveur
1. Remplissage du formulaire: L'étudiant accède à l'interface de demande et complète le formulaire en choisissant la session et le format (PDF ou Excel)	1.1 Afficher le formulaire de demande avec champs de sélection de session et de format	1.1.1 Vérifier l'authentification de l'étudiant et charger les informations liées à la session
2. Envoi de la demande: L'étudiant soumet sa demande	2.1 Transmettre la demande de relevé via l'interface (ex. via RabbitMQ)	2.1.1 Injecter la demande dans la file d'attente destinée à l'administrateur 2.1.2 Notifier l'administrateur d'un nouveau ticket
3. Génération du lien sécurisé: Processus automatisé de génération du lien	3.1 Générer dynamiquement un lien sécurisé (ex. avec un token JWT) valable 48 heures	3.1.1 Créer le token JWT et l'associer à la demande 3.1.2 Enregistrer le lien dans le système de suivi des demandes
4. Envoi du relevé: Le relevé est communiqué à l'étudiant	4.1 Compiler le relevé au format PDF et joindre le lien sécurisé à l'email	4.1.1 Envoyer l'email via Amazon SES 4.1.2 Stocker le document PDF sur AWS S3 pour consultation

Flux Alternatifs et Exceptions

Exception	Système	Serveur
A. Refus administratif	A.1 : Afficher le message explicatif (ex. « Notes non validées »).	A.1.1 : Annuler le processus de génération du relevé.
	A.2 : Indiquer l'annulation du processus de génération du relevé.	A.1.2 : Consigner l'événement de refus dans les logs.
B. Lien expiré ou tentative de fraude	B.1 : Afficher le message (« Ce lien a expiré. Soumettez une nouvelle demande. »).	B.1.1 : Déclencher le verrouillage du compte en cas de fraude.
	B.2 : Signaler l'anomalie à l'utilisateur via l'interface.	B.1.2 : Notifier l'administrateur et enregistrer l'incident dans le log de supervision.

Use Case 5 : Demande de Liste d'Étudiants Valides (Professeur)

Acteurs

- **Professeur** (acteur initiateur)
- **Administrateur** (acteur secondaire)
- **Système d'analyse des données (Spark)** (acteur technique)

Préconditions

- Le professeur est authentifié.
- Le formulaire est correctement rempli avec le niveau (ex. L3) et l'UE (ex. « UE-INFO »).

Postconditions

- Une liste d'étudiants est générée au format CSV ou Excel (maximum 500 lignes).
- Un lien sécurisé (valable 24 heures) est envoyé par email.

Flux Principal Détaillé

Responsable	Système	Serveur
1. Définition des filtres: Le professeur se connecte à l'interface de requête et définit ses critères (ex. niveau L3, UE «UE-INFO», statut, session)	1.1 Afficher l'interface de sélection des filtres avec les options disponibles	1.1.1 Vérifier les droits d'accès et préparer les paramètres pour la requête sur Spark (Hadoop)
2. Exécution de la requête:	2.1 Déclencher la requête	2.1.1 Exécuter le job sur Hadoop

(Lancement automatique après validation)	via Spark pour extraire la liste des étudiants	2.1.2 Retourner et formater le résultat pour affichage
3. Validation du résultat: Le système contrôle la taille du résultat	3.1 Vérifier que le nombre de lignes est inférieur à 500Si >500, afficher « Affinez vos critères. »	3.1.1 Compter le nombre de lignes retournées 3.1.2 Si nécessaire, interrompre le processus et consigner l'erreur
4. Génération et envoi du lien: Finalisation de la procédure par le système	4.1 Générer le fichier (CSV ou Excel), créer un lien sécurisé (valable 24 heures) et afficher l'interface d'envoi	4.1.1 Stocker temporairement le fichier sur Google Cloud Storage 4.1.2 Générer le lien via AWS CloudFront et envoyer l'email

Flux Alternatifs et Exceptions

Exception	Système	Serveur
A. Aucun résultat	A.1 : Afficher le message « Aucun étudiant trouvé pour ces filtres. ». A.2 : Terminer la procédure de requête sans génération de fichier.	A.1.1 : Ne pas générer de fichier. A.1.2 : Consigner l'absence de résultat dans le log d'audit.
B. Requête trop lourde	B.1 : Afficher le message d'erreur (« La requête dépasse les limites système. Contactez l'administrateur. »). B.2 : Inviter à affiner les critères.	B.1.1 : Interrompre le processus de génération du fichier. B.1.2 : Enregistrer l'erreur dans le log et notifier le support technique.

Use Case 6 : Validation Administrative

Acteurs

- **Administrateur** (acteur principal)
- **Système de workflow (Camunda)** (acteur technique)

Préconditions

- Une demande est en attente dans la file Kafka (par exemple, « MODIFICATION_NOTE_123 »).

Postconditions

- La demande est marquée « Approuvée » ou « Rejetée » dans Camund

- Une notification est envoyée via WebSocket au demandeur.

Flux Principal Détaillé

Responsable	Système	Serveur
1. Réception de la notification: L'administrateur reçoit une alerte via Slack	1.1 Afficher dans l'interface les détails de la demande en attente	1.1.1 Récupérer la demande depuis la file Kafka 1.1.2 Transmettre la notification via Slack
2. Examen de la demande: L'administrateur consulte les informations (ancienne note, nouvelle note, justificatif)	2.1 Afficher les informations détaillées de la demande dans l'interface de validation	2.1.1 Récupérer et agréger les données depuis Camunda et la base de versionning
3. Prise de décision: L'administrateur approuve ou refuse via l'interface	3.1 Enregistrer la décision (clique sur « Approuver » ou « Refuser »)	3.1.1 Mettre à jour le statut de la demande dans Camunda 3.1.2 Consigner (et, en cas de refus, enregistrer le motif)
4. Mise à jour et notification: Les conséquences de la décision sont diffusées	4.1 Afficher le résultat de la validation dans l'interface (approuvé ou rejeté)	4.1.1 Envoyer une notification via WebSocket au demandeur 4.1.2 Actualiser les logs de workflow

Flux Alternatifs et Exceptions

Exception	Système	Serveur
A. Données incomplètes	A.1 : Afficher le message « Le justificatif est manquant. Requête renvoyée au professeur. ». A.2 : Exiger une correction avant poursuite.	A.1.1 : Annuler la mise à jour de la demande. A.1.2 : Renvoyer la demande avec les informations manquantes et consigner l'événement dans le log de workflow.
B. Violation de droits	B.1 : Afficher un message d'erreur indiquant la violation de droits. B.2 : Bloquer l'opération de modification dans l'interface utilisateur.	B.1.1 : Empêcher la modification et générer une alerte de sécurité. B.1.2 : Enregistrer la tentative d'accès non autorisé dans le log de sécurité.

Use Case 7 : Exportation/Impression de Données

Acteurs

- **Professeur / Étudiant** (acteur initiateur)
- **Système de génération de documents (Microservice)** (acteur technique)

Préconditions

- Les données sont validées (statut « Approuvé »).
- Le format d'exportation est choisi (PDF ou Excel).

Postconditions

- Un fichier est généré avec des métadonnées (ex. « Généré le 15/12/2024 par Prof. X »).
- Le fichier est stocké temporairement (par exemple sur Google Cloud Storage) et accessible via un lien.

Flux Principal Détaillé

Responsable	Système	Serveur
1. Sélection des données à exporter: L'utilisateur (professeur ou étudiant) choisit, via l'interface, les données à exporter	1.1 Afficher l'interface d'exportation avec options de filtrage ou sélection manuelle	1.1.1 Charger les données depuis la base (pour des données validées, statut « Approuvé »)
2. Choix du format d'exportation: L'utilisateur indique le format souhaité (PDF ou Excel)	2.1 Valider la sélection du format et afficher un aperçu si nécessaire	2.1.1 Si PDF, appliquer un template LaTeX ; si Excel, préparer un export brut avec filtres
3. Génération asynchrone du fichier: Une fois les choix validés, le système lance la génération	3.1 Démarrer le processus de génération asynchrone (ex. via celery et Redis)	3.1.1 Convertir les données dans le format choisi 3.1.2 Gérer le rendu du document et le suivi du processus
4. Mise à disposition du fichier: Une fois généré, le document est rendu accessible	4.1 Afficher ou envoyer un lien sécurisé pour le téléchargement ou lancer l'impression	4.1.1 Stocker temporairement le fichier sur Google Cloud Storage 4.1.2 Générer un lien sécurisé (avec validité définie) 4.1.3 Transmettre le document par email ou via un service d'impression

Flux Alternatifs et Exceptions

Exception	Système	Serveur
A. Format non supporté	A.1 : Afficher le message « Choisissez PDF ou Excel. ». A.2 : Demander une nouvelle sélection de format.	A.1.1 : Interrompre le processus de génération du document. A.1.2 : Consigner l'erreur dans le log pour suivi et analyse ultérieure.
B. Échec de la génération	B.1 : Afficher le message d'erreur (« Erreur technique. Réessayez ou contactez support@univ.fr. »). B.2 : Proposer de relancer la requête de génération.	B.1.1 : Annuler le job de génération du document. B.1.2 : Notifier le support technique via le système de monitoring pour une intervention rapide.