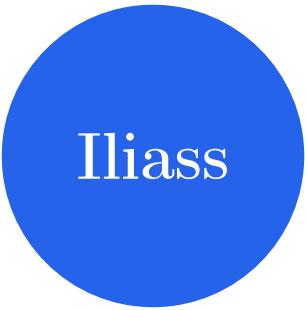


# Guide du Développeur 2

## Demandes & Notifications



Iliass

### Tes responsabilités :

Gestion des demandes (créer, modifier, supprimer)

Workflow de validation (statuts, transitions)

- ⬆️ Upload des pièces jointes
- 🔔 Système de notifications
- ✉️ Envoi d'emails automatiques

## Table des matières

<b>1 Ton rôle dans l'équipe</b>	<b>2</b>
1.1 Fichiers à créer . . . . .	2
<b>2 Comprendre le Workflow</b>	<b>3</b>
2.1 Les 6 statuts possibles . . . . .	3
2.2 Qui peut faire quoi? . . . . .	3
2.3 Les transitions possibles . . . . .	3
<b>3 Tâche 1 : Créer le modèle Demande</b>	<b>4</b>
3.1 Fichier app/models/Demande.php . . . . .	4
3.2 Génération du numéro unique . . . . .	4
3.3 Les jointures SQL . . . . .	4
<b>4 Tâche 2 : Créer le modèle PieceJointe</b>	<b>5</b>
4.1 Fichier app/models/PieceJointe.php . . . . .	5
4.2 Règles de sécurité pour l'upload . . . . .	5
4.3 Comment fonctionne l'upload en PHP . . . . .	5
<b>5 Tâche 3 : Créer le modèle Validation</b>	<b>6</b>
5.1 Fichier app/models/Validation.php . . . . .	6
<b>6 Tâche 4 : Créer le modèle Notification</b>	<b>7</b>
6.1 Fichier app/models/Notification.php . . . . .	7
6.2 Types de notifications . . . . .	7
<b>7 Tâche 5 : Créer le contrôleur Demande</b>	<b>8</b>
7.1 Fichier app/controllers/DemandeController.php . . . . .	8
7.2 Vérifications à faire dans chaque méthode . . . . .	9
<b>8 Tâche 6 : Indicateur d'urgence</b>	<b>10</b>
<b>9 Tâche 7 : Notifications et Emails</b>	<b>11</b>
9.1 Quand envoyer des notifications? . . . . .	11
9.2 Templates d'emails . . . . .	11
<b>10 Tâche 8 : Créer les vues</b>	<b>12</b>
10.1 Liste des demandes (professeur) . . . . .	12
10.2 Formulaire de création . . . . .	12
10.3 Détail d'une demande . . . . .	12
<b>11 Planning suggéré</b>	<b>13</b>
<b>12 Checklist finale</b>	<b>13</b>

# 1 Ton rôle dans l'équipe

Tu gères le **cœur fonctionnel** de l'application : le cycle de vie complet des demandes de changement/annulation de séances.

## À savoir

C'est grâce à ton travail que :

- Les professeurs peuvent créer des demandes
- L'assistante peut valider ou rejeter
- Le directeur peut approuver
- Tout le monde reçoit des notifications

## 1.1 Fichiers à créer

app/models/

Demande.php	→ CRUD des demandes
PieceJointe.php	→ Gestion des fichiers uploadés
Validation.php	→ Historique des validations
Notification.php	→ Gestion des notifications

app/controllers/

DemandeController.php	→ Actions sur les demandes
NotificationController.php	→ Actions sur les notifications

app/views/professeur/demandes/

index.php	→ Liste des demandes du prof
create.php	→ Formulaire de création
show.php	→ Détail d'une demande
edit.php	→ Formulaire de modification

app/views/assistante/

file-attente.php	→ Liste des demandes à traiter
demande-detail.php	→ Page de traitement

app/views/directeur/

demandes-validation.php	→ Demandes à approuver
-------------------------	------------------------

app/views/emails/

demande-soumise.php
demande-validee.php
demande-approuvee.php
demande-rejetee.php

## 2 Comprendre le Workflow

Avant de coder, tu dois **parfaitemenr comprendre** le cycle de vie d'une demande.

### 2.1 Les 6 statuts possibles

Statut	Couleur	Signification
brouillon	Gris	Le prof a commencé mais pas encore soumis
en_attente	Jaune	Soumis, attend l'assistante
validee_assistante	Bleu	L'assistante a validé, attend le directeur
approuvee	Vert	Le directeur a approuvé (terminé)
rejetee	Rouge	Refusée par l'assistante ou le directeur
annulee	Noir	Annulée par le professeur

### 2.2 Qui peut faire quoi ?

Action	Professeur	Assistante	Directeur
Créer une demande			
Modifier une demande	(si brouillon)		
Soumettre			
Annuler			
Valider → Directeur			
Rejeter			
Demander des infos			
Approuver (finale)			

### 2.3 Les transitions possibles

#### 💡 Concept

**Règle importante :** Un statut ne peut passer qu'à certains autres statuts.

- brouillon → en\_attente (soumettre) ou annulee
- en\_attente → validee\_assistante ou rejetee ou annulee
- validee\_assistante → approuvee ou rejetee

Tu dois vérifier ces règles dans ton code !

### 3 Tâche 1 : Créer le modèle Demande

#### 3.1 Fichier app/models/Demande.php

##### ☰ Tâche

**Objectif :** Gérer toutes les opérations sur la table `demandes`.

**Méthodes à créer :**

- `generateNumero()` : Génère un numéro unique (ex : DEM-2025-00042)
- `create($data)` : Créer une nouvelle demande
- `findById($id)` : Trouver une demande avec tous ses détails (jointures)
- `findByProfesseur($profId)` : Lister les demandes d'un professeur
- `findEnAttente()` : Lister les demandes en attente (pour l'assistante)
- `findValideesAssistante()` : Demandes validées (pour le directeur)
- `update($id, $data)` : Modifier une demande
- `updateStatut($id, $statut)` : Changer le statut
- `soumettre($id)` : Passer de brouillon à en\_attente
- `delete($id)` : Supprimer (seulement si brouillon)
- `countByStatut($profId)` : Compter par statut (pour le dashboard)

#### 3.2 Génération du numéro unique

##### ⓘ À savoir

Le numéro de demande suit le format : **DEM-ANNÉE-NUMÉRO**

Exemple : DEM-2025-00042

**Comment faire :**

1. Récupérer le dernier numéro de l'année en cours
2. Extraire le numéro et l'incrémenter
3. Formater avec des zéros devant (sprintf)

#### 3.3 Les jointures SQL

Quand tu récupères une demande, tu as besoin d'informations de plusieurs tables. Utilise des **JOIN** pour récupérer en une seule requête :

- Le nom du professeur (table `utilisateurs`)
- Les infos de la séance (table `seances`)
- Le nom de la matière (table `matieres`)
- Le nom de la salle (table `salles`)

**Concept à rechercher :** "SQL JOIN multiple tables"

## 4 Tâche 2 : Créer le modèle PieceJointe

### 4.1 Fichier app/models/PieceJointe.php

#### ☰ Tâche

**Objectif :** Gérer l'upload sécurisé des fichiers (certificats, justificatifs).

**Méthodes à créer :**

- `upload($file, $demandeId)` : Uploader un fichier
- `findByDemande($demandeId)` : Lister les pièces jointes d'une demande
- `delete($id)` : Supprimer une pièce jointe

### 4.2 Règles de sécurité pour l'upload

#### ⚠ Important

**TRÈS IMPORTANT - Sécurité upload :**

1. **Vérifier le type MIME réel** avec `finfo` (pas l'extension !)
2. **Types autorisés** : PDF, JPG, PNG uniquement
3. **Taille max** : 5 Mo
4. **Renommer le fichier** avec un nom unique (unqid)
5. **Stocker hors du dossier public** (dans `storage/uploads/`)

### 4.3 Comment fonctionne l'upload en PHP

#### 💡 Concept

Quand un utilisateur envoie un fichier via un formulaire :

1. PHP stocke le fichier temporairement
2. Les infos sont dans `$_FILES['nom_du_champ']`
3. Tu dois le déplacer avec `move_uploaded_file()`
4. Puis enregistrer les infos en base de données

**Concept à rechercher** : "PHP file upload security"

## 5 Tâche 3 : Créer le modèle Validation

### 5.1 Fichier app/models/Validation.php

#### ☰ Tâche

**Objectif :** Garder un historique de toutes les actions sur une demande.

Chaque fois que quelqu'un valide, rejette ou approuve, on enregistre :

- Qui a fait l'action (validateur\_id)
- Quelle action (validation, rejet, approbation)
- Son commentaire éventuel
- La date/heure

**Méthodes à créer :**

- `create($demandeId, $validateurId, $action, $commentaire)`
- `findByDemande($demandeId)` : Historique complet
- `getLastValidation($demandeId)` : Dernière action

#### ⓘ À savoir

Cet historique permet d'afficher une **timeline** dans le détail de la demande, montrant toutes les étapes par lesquelles elle est passée.

## 6 Tâche 4 : Créer le modèle Notification

### 6.1 Fichier app/models/Notification.php

#### ☰ Tâche

**Objectif :** Gérer les notifications in-app (l'icône cloche dans le header).

**Méthodes à créer :**

- `create($userId, $type, $titre, $message, $demandeId)`
- `findByUtilisateur($userId)` : Toutes les notifications
- `getUnread($userId)` : Notifications non lues
- `countUnread($userId)` : Compter les non lues (pour le badge)
- `markAsRead($id)` : Marquer comme lue
- `markAllAsRead($userId)` : Tout marquer comme lu

### 6.2 Types de notifications

- `nouvelle_demande` : Nouvelle demande à traiter (pour assistante)
- `validation` : Demande validée par l'assistante (pour prof)
- `approbation` : Demande approuvée (pour prof)
- `rejet` : Demande rejetée (pour prof)
- `commentaire` : Nouveau commentaire

## 7 Tâche 5 : Créer le contrôleur Demande

### 7.1 Fichier app/controllers/DemandeController.php

C'est le fichier le plus important. Il contient toute la logique métier.

#### ☰ Tâche

##### Méthodes pour le PROFESSEUR :

- `index()` : Afficher la liste des demandes du prof
- `create()` : Afficher le formulaire de création
- `store()` : Enregistrer la nouvelle demande
- `show($id)` : Afficher le détail d'une demande
- `edit($id)` : Afficher le formulaire de modification
- `update($id)` : Enregistrer les modifications
- `delete($id)` : Supprimer une demande (si brouillon)
- `submit($id)` : Soumettre un brouillon
- `cancel($id)` : Annuler une demande

#### ☰ Tâche

##### Méthodes pour l'ASSISTANTE :

- `fileAttente()` : Afficher la file d'attente
- `valider($id)` : Valider et transmettre au directeur
- `rejeter($id)` : Rejeter avec motif obligatoire
- `demandeInfos($id)` : Renvoyer au prof pour compléments

#### ☰ Tâche

##### Méthodes pour le DIRECTEUR :

- `listeValidation()` : Demandes en attente d'approbation
- `approuver($id)` : Approuver définitivement
- `rejeter($id)` : Rejeter avec motif

## 7.2 Vérifications à faire dans chaque méthode

### ⚠ Important

Pour CHAQUE méthode, tu dois vérifier :

1. L'utilisateur est-il connecté ?
2. A-t-il le bon rôle pour cette action ?
3. Le token CSRF est-il valide ? (pour les POST)
4. La demande existe-t-elle ?
5. La demande est-elle dans le bon statut pour cette action ?
6. L'utilisateur a-t-il le droit sur cette demande spécifique ?

## 8 Tâche 6 : Indicateur d'urgence

Dans la file d'attente de l'assistante, tu dois afficher un indicateur d'urgence basé sur la date de la séance :

### À savoir

#### Comment calculer :

1. Récupérer la date de la séance
2. Calculer la différence avec la date actuelle
3. Convertir en jours
4. Assigner la couleur selon le tableau

## 9 Tâche 7 : Notifications et Emails

### 9.1 Quand envoyer des notifications ?

Événement	Notifier qui ?	Email ?
Demande soumise	Assistante(s)	Oui
Demande validée par assistante	Professeur + Directeur	Oui
Demande approuvée	Professeur	Oui
Demande rejetée	Professeur	Oui
Demande d'infos complémentaires	Professeur	Oui

### 9.2 Templates d'emails

Crée des fichiers PHP dans `app/views/emails/` pour chaque type d'email. Chaque template doit contenir :

- Le sujet de l'email
- Le corps du message (HTML ou texte)
- Les variables dynamiques (nom du prof, numéro de demande, etc.)

#### **i** À savoir

**Conseil :** Pour l'envoi d'emails, Dev 1 aura créé une classe `Mailer.php`. Tu l'utilises simplement en appelant sa méthode `send()`.

## 10 Tâche 8 : Créer les vues

### 10.1 Liste des demandes (professeur)

La page `index.php` doit afficher :

- Widgets avec le compteur par statut (X en attente, Y approuvées, etc.)
- Tableau avec : Numéro, Type, Date séance, Matière, Statut, Date création
- Filtres par statut
- Bouton "Nouvelle demande"
- Actions : Voir, Modifier (si brouillon), Supprimer (si brouillon)

### 10.2 Formulaire de création

Le formulaire doit contenir :

- Champ caché CSRF
- Select : Type de demande (Changement / Annulation)
- Select : Séance concernée (liste des cours du prof)
- Si changement : Date picker pour la nouvelle date
- Si changement : Heure début et fin
- Si changement : Select nouvelle salle
- Textarea : Motif (minimum 20 caractères)
- Input file : Pièces jointes (multiple, optionnel)
- Bouton "Enregistrer brouillon"
- Bouton "Soumettre"

### 10.3 Détail d'une demande

Cette page doit afficher :

- Toutes les informations de la demande
- Le statut actuel avec sa couleur
- La timeline des validations (historique)
- Les pièces jointes téléchargeables
- Les boutons d'action selon le rôle et le statut

## 11 Planning suggéré

---

### Tâches

---

Modèle Demande.php (CRUD de base)  
Modèle PieceJointe.php (upload sécurisé)  
Modèles Validation.php et Notification.php  
DemandeController (actions professeur)  
DemandeController (actions assistante)  
DemandeController (actions directeur)  
Vues du professeur  
Vues assistante et directeur  
Notifications et emails  
Tests et debug

---

## 12 Checklist finale

Le professeur peut créer une demande  
Le professeur peut modifier/supprimer un brouillon  
Le professeur peut soumettre une demande  
L'upload de fichiers fonctionne et est sécurisé  
L'assistante voit la file d'attente avec urgences  
L'assistante peut valider ou rejeter  
Le directeur peut approuver ou rejeter  
Les transitions de statut respectent les règles  
Les notifications sont créées à chaque étape  
Les emails sont envoyés  
L'historique des validations est affiché

---

**Bon courage Iliass ! 🔑**

Tu gères le cœur de l'application.