Cahier des Charges - Système de Gestion des Notes

1. Vue d'ensemble du projet

Titre: Système de Gestion des Notes (Notes Management System)
Client: Institution ICT
Développement: Django 4.2.0
Theme UI: AdminLTE 4
Date de démarrage: 2024
État actuel: Phase 2 (En développement - Tableau des notes complété)

---

2. Objectifs du projet

Objectif principal

Créer une plateforme de gestion centralisée des notes étudiantes permettant aux administrateurs et instructeurs de:

- Consulter les notes par département, filière et niveau
- Éditer les notes (CC, TP, SN)
- Importer/exporter les données via Excel
- Gérer les utilisateurs et permissions

Objectifs secondaires

- Simplifier l'administration des données avec une interface intuitive
- Assurer la sécurité des données via authentification et permissions
- Fournir des statistiques et rapports sur les performances étudiantes
- Faciliter la maintenance avec une documentation complète

---

3. Exigences fonctionnelles

3.1 Authentification et Gestion des Utilisateurs

- [x] Système de login/logout
- [x] Gestion des sessions
- [x] Authentification django.contrib.auth
- [x] Distinction rôles: Administrateur, Instructeur, Étudiant
- [ ] Authentification LDAP (bonus)
- [ ] Réinitialisation de mot de passe

3.2 Tableau de Bord (Home)

- [x] Affichage des statistiques globales
  - Nombre total d'étudiants
  - Nombre total de filières

    - Nombre de cours/UE
    - Moyenne générale
- [x] Design minimal avec tuiles (stat boxes)
- [x] Actions rapides (liens vers tableau des notes, admin)
- [x] Notes récentes affichées
- [ ] Graphiques de performance (bonus)
- [ ] Alertes pour notes faibles

3.3 Tableau des Notes

Filtrage et Navigation

- [x] Filtme en cascade: Département → Filière → Niveau
- [x] Appels API asynchrones pour filtrage
- [x] Pagination des résultats
- [x] Chargement dynamique des options de filtre
- [ ] Recherche par nom d'étudiant
- [ ] Filtrage par UE spécifique
- [ ] Tri des colonnes (bonus)

Édition des Notes

- [x] Modal d'édition pour CC/TP/SN par étudiant/UE
- [x] Validation des saisies (nombres décimaux, plages 0-20)
- [x] Permissions basées sur rôles (superuser/staff peut éditer)
- [x] Affichage distinction cellules éditables vs lecture seule
- [ ] Validation côté serveur des limites de notes
- [ ] Historique des modifications

Import/Export

- [x] Export des notes en Excel (.xlsx)
- [x] Sélection des UE avant export
- [x] Import des notes depuis Excel
- [x] Modal pour sélectionner UE avant import
- [x] Boutons désactivés tant que niveau non sélectionné
- [x] Gestion des erreurs lors de l'import
- [ ] Export en PDF avec mise en forme
- [ ] Import par CSV
- [ ] Validation du fichier avant import

3.4 Interface Utilisateur (AdminLTE 4)

- [x] Template de base avec navbar sticky
- [x] Navigation principale (Home, Tableau Notes, Admin)
- [x] Affichage du username et bouton Déconnexion séparé
- [x] Footer fixe en bas
- [x] Responsive design (mobile/tablet/desktop)

- [x] CDN avec vérification SRI (SHA384)
- [x] Badge styling pour tous les variants Bootstrap
- [x] Thème cohérent AdminLTE
- [ ] Mode sombre (Dark mode)
- [ ] Personnalisation des couleurs

3.5 Administration Django

- [x] Interface admin Django intégrée
- [x] Gestion des Départements
- [x] Gestion des Filières
- [x] Gestion des Niveaux
- [x] Gestion des UE (Unités d'Enseignement)
- [x] Gestion des Étudiants
- [x] Gestion des Notes
- [ ] Gestion en masse (bulk actions)
- [ ] Import depuis CSV via admin

3.6 API REST

- [x] Endpoint `/api/filieres/` - Filières par département
- [x] Endpoint `/api/niveaux/` - Niveaux par filière
- [x] Endpoint `/api/notes/` - Notes filtrées
- [x] Endpoint `/api/notes/import/` - Import Excel
- [x] Endpoint `/api/notes/export/` - Export Excel
- [ ] Documentation API complète (DRF docs)
- [ ] Authentification API (tokens)
- [ ] Pagination API standardisée

---

4. Exigences non-fonctionnelles

4.1 Performance

- [x] Pages charge < 2 secondes (avec CDN)
- [ ] Images optimisées (WebP)
- [ ] Cache des filtres (Redis, bonus)
- [ ] Lazy loading des données
- [ ] Compression gzip activée

4.2 Sécurité

- [x] Protection CSRF
- [x] Validation des entrées utilisateur
- [x] Permissions basées rôles (Django permissions)
- [x] Session timeout (optionnel)
- [ ] Chiffrement des mots de passe (bcrypt)
- [ ] Audit logs pour modifications sensibles
- [ ] HTTPS obligatoire

4.3 Maintenabilité

- [x] Code documenté avec commentaires
- [x] .gitignore approprié
- [x] requirements.txt avec dépendances
- [x] README.md avec instructions
- [x] Tests unitaires (Django TestCase)
- [ ] Code coverage > 80%
- [ ] Documentation API Swagger/OpenAPI

4.4 Compatibilité

- [x] Django 4.2.0
- [x] Python 3.8+
- [x] Windows / Linux / macOS
- [x] Firefox, Chrome, Safari, Edge modernes
- [ ] Accessibilité WCAG 2.1 AA

---

5. Architecture Technique

5.1 Stack Technologique

```
FRONTEND
├── HTML5
├── Bootstrap 5.3.0 (CDN)
├── AdminLTE 4 (CDN)
├── jQuery 3.6.0 (CDN)
├── Font Awesome 6.4.0 (CDN)
└── Vanilla JavaScript (ES6)

BACKEND
├── Django 4.2.0
├── Django ORM
├── SQLite (dev), PostgreSQL (prod)
└── Python 3.8+

LIBRARIES
├── openpyxl (Excel I/O)
├── reportlab (PDF generation)
├── Pillow (Image processing)
└── pytest-django (Testing)

DEPLOYMENT
├── Git (Version Control)
├── GitHub (Repository)
└── TBD (Hosting: Heroku, PythonAnywhere, etc.)
```

5.2 Structure des Fichiers

```
backend/
├── manage.py
├── db.sqlite3
├── requirements.txt
├── .gitignore
├── README.md
├── CAHIER_DES_CHARGES.md
├── PLAN_IMPLEMENTATION.md
├── backend/
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   └── asgi.py
└── notes/
    ├── models.py
    ├── views.py
    ├── urls.py
    ├── forms.py (TBD)
    ├── serializers.py (TBD)
    ├── tests.py (✓ 12/12 passing)
    ├── admin.py
    ├── templates/notes/
    │   ├── base_adminlte.html (Template master)
    │   ├── home.html (Ancienne version)
    │   ├── home_adminlte.html (✓ Dashboard)
    │   ├── tableau_notes_adminlte.html (✓ Tableau)
    │   └── ... (autres pages)
    ├── static/css/
    │   └── custom.css (TBD)
    ├── static/js/
    │   └── grades.js (✓ Contrôleur tableau)
    └── migrations/
        ├── 0001_initial.py
        └── 0002_ue_note.py
```

5.3 Modèles de Données

```
Département
├── id (PK)
├── nom
└── code

Filière
├── id (PK)
├── nom
├── code
└── departement_id (FK)

Niveau
├── id (PK)
```

```
├── numero (1, 2, 3, etc.)
└── filiere_id (FK)

UE (Unité d'Enseignement)
├── id (PK)
├── code
├── nom
├── credits
└── filiere_id (FK)

Étudiant
├── id (PK)
├── user_id (FK)
├── matricule
├── nom
├── prenom
└── filiere_id (FK)

Note
├── id (PK)
├── etudiant_id (FK)
├── ue_id (FK)
├── cc (Contrôle Continu, 0-20)
├── tp (Travaux Pratiques, 0-20)
├── sn (Synthèse, 0-20)
└── date_creation
```

---

6. État actuel vs État attendu

Feature	État	% Complet	Notes
Authentication	✅ Complété	100%	Login/logout fonctionnel
Home Dashboard	✅ Complété	100%	Tuiles stat + actions rapides
Tableau Notes	✅ Complété	100%	Filtres, édition, import/export
Filtres Cascadants	✅ Complété	100%	API async fonctionnelle
API REST	⚠ Basique	70%	Endpoints principaux OK, docs TBD
Admin Django	✅ Complété	100%	CRUD complet sur tous modèles
Tests	✅ Complété	100%	12/12 tests passing
Git & Docs	✅ Complété	100%	README + requirements + gitignore
Graphiques/Stats Avancées	✗ TBD	0%	À implémenter
Recherche Avancée	✗ TBD	0%	À implémenter
Historique Modifications	✗ TBD	0%	À implémenter
Mode Sombre	✗ TBD	0%	À implémenter

Feature	État	% Complet	Notes
Export PDF	❌ TBD	0%	À implémenter
Authentification LDAP	❌ TBD	0%	Bonus futur

7. Critères d'acceptation

Phase 1: Core Features (✅ COMPLÉTÉE)

- [x] Authentification fonctionnelle
- [x] Dashboard avec stats
- [x] Tableau des notes complètement opérationnel
- [x] Filtres en cascade département → filière → niveau
- [x] Édition des notes avec modal
- [x] Import/Export Excel
- [x] Tests passants (12/12)
- [x] Git initialisé avec commits
- [x] Documentation README

Phase 2: Amélioration UX (⚠ EN COURS)

- [ ] Graphiques PyChart.js ou Chart.js sur tableau notes
- [ ] Recherche par nom d'étudiant
- [ ] Filtres supplémentaires (UE spécifique, plage de notes)
- [ ] Export PDF avec mise en forme
- [ ] Valider que les notes sont dans plages valides
- [ ] Message de confirmation avant import destructif

Phase 3: Avancé (📋 PLANIFIÉE)

- [ ] Historique des modifications avec audit logs
- [ ] Mode sombre (CSS variables)
- [ ] Statistiques par département/filière
- [ ] Authentification LDAP
- [ ] Dashboard instructeur personnel
- [ ] Benchmark de performance

Phase 4: Production (🚀 FUTURO)

- [ ] Déploiement sur serveur (PythonAnywhere, Heroku)
- [ ] Configuration HTTPS/SSL
- [ ] Sauvegarde automatique (backup)
- [ ] Monitoring et alertes
- [ ] Migration vers PostgreSQL

8. Risques et Mitigations

Risque	Probabilité	Impact	Mitigation
Perte de données	Basse	Critique	Backups quotidiens, git history
Accès non autorisé	Basse	Critique	Permissions Django, CSRF, HTTPS
Performance dégradée	Moyenne	Moyenne	Cache, pagination, indexation DB
Incompatibilité navigateur	Basse	Basse	Tests cross-browser, fallbacks
Données corrompues à l'import	Moyenne	Moyenne	Validation, transaction rollback
Délai de livraison	Basse	Moyen	Planification itérative, sprints

---

9. Ressources

Équipe

- **Développeur**: T4zor
- **QA/Testeur**: À assigner
- **Designer UI/UX**: À assigner (AdminLTE pré-fait)

Outils

- Visual Studio Code
- Git/GitHub
- Django Development Server
- SQLite / PostgreSQL
- AdminLTE 4 Theme

Documentation utile

- Django 4.2 Official Docs
- AdminLTE 4 Docs
- Bootstrap 5 Docs
- openpyxl Docs

---

10. Validation et Sign-off

Rôle	Nom	Date	Signature
Développeur	T4zor	11/02/2026	✓
Chef de Projet	TBD		
Client	Institution ICT		

---

document généré: 11/02/2026
Prochaine révision: À planifier
Version: 1.0