

RÉPUBLIQUE DU CAMEROUN
REPUBLIC OF CAMEROON
Peace – Work – Fatherland

UNIVERSITÉ DE DSCHANG
UNIVERSITY OF DSCHANG
Scholae Thesaurus Dschangensis Ibi Cordum

BP 96, Dschang (Cameroun) – Tél./Fax (237) 233 45 13 81
Website : <http://www.univ-dschang.org>
E-mail : udsrectorat@univ-dschang.org



INSTITUT UNIVERSITAIRE DE TECHNOLOGIE
FOTSO VICTOR DE BANDJOUN
FOTSO VICTOR UNIVERSITY
INSTITUTE OF TECHNOLOGY

Département de Génie Télécommunication et
réseau

Department of Computer Engineering
BP 134, Bandjoun – Tél./Fax (237) 99 31 61 30 / 70 64 23 92
Website : <http://www.univ-dschang.org/iutfv/>
E-mail : iut.fotsovictor@univ-dschang.org

DEPARTEMENT DE GENIE INFORMATIQUE

PROJET UML

APPLICATION DE GENERATION D'EXERCICES(GenEx)

Licence3

GROUPE13 :

Rédigée par :

| Noms | MATRICULE |
|------------------------|------------------|
| DJAMPOU Djordan | CM-UDS-25IUT0491 |
| TOUKAM FONKOU FRANKLIN | CM-UDS-23IUT0215 |

Sous l'encadrement :

Dr FOSTING Bertrang

Année académique :2025-2026

Table des matières

| | |
|--|----|
| Résumé Exécutif | 3 |
| Objectifs clés | 3 |
| Bénéfices attendus | 3 |
| 1. Contexte et Enjeux | 4 |
| 1.1 Contexte du projet | 4 |
| 1.2 Problématiques identifiées | 4 |
| 1.3 Opportunités | 4 |
| 2. Vision du Produit | 5 |
| 2.1 Vision stratégique | 5 |
| 2.2 Proposition de valeur | 5 |
| 2.3 Différenciation | 5 |
| 3. Périmètre du Projet | 7 |
| 3.1 Fonctionnalités incluses | 7 |
| Module 1 : Gestion des utilisateurs | 7 |
| Module 2 : Génération d'exercices | 7 |
| Module 3 : Traduction de documents | 7 |
| 3.2 Fonctionnalités exclues (hors périmètre) | 8 |
| 4. Acteurs et Parties Prenantes | 9 |
| 4.1 Utilisateurs finaux | 9 |
| 4.2 Parties prenantes techniques | 9 |
| 5. Exigences Fonctionnelles | 10 |
| 5.1 Authentification et Gestion des Utilisateurs | 10 |
| 5.2 Gestion des Documents Sources | 10 |
| 5.3 Génération d'Exercices | 11 |
| 5.4 Traduction de Documents | 12 |
| 6. Exigences Non Fonctionnelles | 14 |
| 6.1 Performance | 14 |
| 6.2 Sécurité | 14 |
| 6.3 Disponibilité et Fiabilité | 14 |
| 6.4 Utilisabilité | 14 |
| 7. Architecture Technique | 15 |
| 7.1 Architecture globale | 15 |
| 7.2 Stack Technique | 15 |
| 7.3 Architecture Backend | 15 |
| 7.4 Base de Données | 16 |
| 8. Workflows Principaux | 17 |
| 8.1 Workflow : Génération d'exercices | 17 |
| 8.2 Workflow : Traduction de PDF | 17 |
| 9.1 UC-01 : Créer un compte utilisateur | 18 |
| 9.2 UC-02 : Générer des exercices | 19 |

| | |
|--|----|
| 9.3 UC-03 : Traduire un document PDF | 20 |
| 10. Contraintes et Risques | 21 |
| 10.1 Contraintes techniques | 21 |
| 10.2 Contraintes réglementaires | 21 |
| 10.3 Analyse des risques | 21 |
| 11. Planning Prévisionnel | 22 |
| 11.1 Phases du projet | 22 |
| 12. Assurance Qualité | 23 |
| 12.1 Stratégie de tests | 23 |
| 12.2 Critères d'acceptation | 23 |
| 12.3 Gestion des bugs | 23 |
| 13. Déploiement et Exploitation | 24 |
| 13.1 Environnements | 24 |
| 13.2 Architecture de déploiement | 24 |
| 13.3 Procédure de déploiement | 24 |
| 13.4 Monitoring et alertes | 24 |
| 15. Glossaire | 26 |
| 16. Annexes | 27 |
| Annexe A : Références API | 27 |
| Annexe B : Schémas de la base de données | 27 |
| Annexe C : Exemples de requêtes API | 27 |
| Conclusion | 28 |
| Prochaines étapes | 28 |

Résumé Exécutif

GenEX-APP est une plateforme web innovante destinée aux enseignants et formateurs, permettant la génération automatique d'exercices pédagogiques à partir de documents sources et la traduction professionnelle de PDF.

Objectifs clés

- Automatiser la création d'exercices pédagogiques variés (QCM, questions ouvertes, etc.)
- Permettre la traduction automatique et professionnelle de documents PDF
- Réduire le temps de préparation des supports de cours de 70%
- Garantir la qualité académique des contenus générés par IA
- Offrir une interface intuitive et moderne

Bénéfices attendus

- Gain de temps significatif pour les enseignants
- Diversification automatique des formats d'exercices
- Accessibilité multilingue des contenus pédagogiques
- Traçabilité et historique des générations
- Exportation professionnelle en PDF

1. Contexte et Enjeux

1.1 Contexte du projet

Dans le contexte éducatif actuel, les enseignants font face à une charge de travail croissante pour la préparation de supports pédagogiques diversifiés. La génération manuelle d'exercices variés et la traduction de documents représentent des tâches chronophages qui limitent le temps consacré à l'accompagnement des apprenants.

L'intelligence artificielle générative offre désormais des opportunités pour automatiser ces tâches répétitives tout en maintenant un haut niveau de qualité académique.

1.2 Problématiques identifiées

- **Temps de préparation élevé** : La création manuelle d'exercices variés nécessite plusieurs heures par fiche
- **Traduction coûteuse** : Les traductions professionnelles de documents sont onéreuses et lentes
- **Manque de diversité** : Difficulté à varier les formats d'exercices par manque de temps
- **Barrière linguistique** : Accès limité aux ressources pédagogiques en langues étrangères

1.3 Opportunités

L'émergence des modèles d'IA générative (Gemini, DeepSeek) permet aujourd'hui de :

- Générer automatiquement des exercices de qualité académique
- Traduire des documents en conservant la mise en forme
- Analyser et extraire du contenu pédagogique depuis des PDF
- Produire des exports professionnels prêts à l'impression

2. Vision du Produit

2.1 Vision stratégique

GenEX-APP aspire à devenir l'outil de référence pour les enseignants souhaitant optimiser leur préparation pédagogique grâce à l'intelligence artificielle, tout en conservant le contrôle total sur la qualité et la pertinence des contenus générés.

2.2 Proposition de valeur

| | |
|----------------------|---|
| Pour | Les enseignants, formateurs et créateurs de contenu pédagogique |
| Qui | Ont besoin de générer rapidement des exercices variés et traduire des documents |
| GenEX-APP est | Une plateforme intelligente de génération de contenu pédagogique |
| Qui permet | De diviser par 3 le temps de préparation tout en diversifiant les formats |

2.3 Différenciation

GenEX-APP se différencie par :

- L'utilisation combinée de plusieurs IA (Gemini + DeepSeek) pour optimiser qualité et coût
- La génération de PDF professionnels avec mise en forme académique

- L'analyse OCR avancée pour extraire du contenu de documents numérisés
- La traduction avec conservation de la mise en page originale
- Un système de crédits flexible et transparent

3. Périmètre du Projet

3.1 Fonctionnalités incluses

Module 1 : Gestion des utilisateurs

- Inscription et authentification par email/mot de passe
- Gestion des profils utilisateurs (enseignant, administrateur)
- Système de crédits et quotas
- Historique des générations et traductions

Module 2 : Génération d'exercices

- Upload de documents sources (PDF, DOCX, images)
- Extraction OCR intelligente du texte
- Configuration des types d'exercices (QCM, Vrai/Faux, Questions ouvertes, etc.)
- Génération via IA avec paramétrage (nombre, difficulté, langue)
- Édition et prévisualisation des exercices
- Export PDF avec version questions et version corrigé
- Génération de QR code pour accès mobile

Module 3 : Traduction de documents

- Upload de PDF à traduire
- Détection automatique de la langue source
- Choix de la langue cible (français, anglais, espagnol, etc.)
- Traitement asynchrone avec suivi de progression
- Conservation de la mise en page originale
- Export du PDF traduit
- Historique des traductions

3.2 Fonctionnalités exclues (hors périmètre)

- Édition collaborative en temps réel
- Application mobile native (iOS/Android)
- Intégration avec plateformes LMS (Moodle, Canvas) - prévu phase 2
- Reconnaissance vocale pour génération d'exercices
- Gamification avancée (badges, classements)

4. Acteurs et Parties Prenantes

4.1 Utilisateurs finaux

| Acteur | Rôle | Besoins principaux |
|-------------------|--|------------------------------|
| Enseignant | Utilisateur principal créant des exercices | Rapidité, qualité, diversité |
| Etudiant | Utilisateur potentiel | Tableau de bord |

4.2 Parties prenantes techniques

- **Équipe de développement** : Conception, développement et maintenance de la plateforme
- **Équipe DevOps** : Déploiement, monitoring et infrastructure cloud
- **Équipe QA** : Tests, validation de la qualité des contenus générés
- **Support utilisateurs** : Assistance technique et accompagnement des enseignants

5. Exigences Fonctionnelles

Cette section détaille les exigences fonctionnelles organisées par module. Chaque exigence est identifiée par un code unique (RF-XXX) pour la traçabilité.

5.1 Authentification et Gestion des Utilisateurs

| ID | Exigence | Priorité |
|---------------|---|----------|
| RF-001 | Le système doit permettre l'inscription avec email et mot de passe | Haute |
| RF-002 | Le système doit valider la force du mot de passe (8 caractères minimum) | Haute |
| RF-003 | Le système doit générer un token JWT à la connexion | Haute |
| RF-004 | Le système doit permettre la déconnexion et blacklister le token | Moyenne |
| RF-005 | Le système doit gérer les rôles (Enseignant, Administrateur) | Haute |

5.2 Gestion des Documents Sources

| ID | Exigence | Priorité |
|---------------|------------------------------|----------|
| RF-010 | Le système doit accepter les | Haute |

| | | |
|---------------|---|----------------|
| | formats PDF, DOCX, images (JPG, PNG) | |
| RF-011 | Le système doit extraire le texte via OCR (Tesseract) | Haute |
| RF-012 | Le système doit nettoyer le texte extrait (IA) | Moyenne |
| RF-013 | Le système doit stocker les documents de manière sécurisée | Haute |
| RF-014 | Le système doit permettre la suppression de documents | Moyenne |

5.3 Génération d'Exercices

| ID | Exigence | Priorité |
|---------------|------------------------------|-----------------|
| RF-020 | Le système doit permettre | Haute |

| | | |
|---------------|---|----------------|
| | de configurer le type d'exercices (QCM, ouvert, etc.) | |
| RF-021 | Le système doit permettre de spécifier le nombre d'exercices | Haute |
| RF-022 | Le système doit générer les exercices via Gemini ou DeepSeek | Haute |
| RF-023 | Le système doit générer un PDF de questions et un PDF de corrigés | Haute |
| RF-024 | Le système doit gérer les formules mathématiques LaTeX | Moyenne |
| RF-025 | Le système doit générer un QR code pour accès mobile | Basse |

5.4 Traduction de Documents

| ID | Exigence | Priorité |
|---------------|--------------------------|-----------------|
| RF-030 | Le système doit détecter | Haute |

| | | |
|--------------------|--|----------------|
| | automatiquement la langue source | |
| RF- 031 | Le système doit supporter au moins 10 langues cibles | Moyenne |
| RF- 032 | Le système doit traiter les traductions en arrière-plan | Haute |
| RF- 033 | Le système doit afficher la progression de la traduction | Moyenne |
| RF- 034 | Le système doit conserver la mise en page du PDF original | Haute |

6. Exigences Non Fonctionnelles

6.1 Performance

RNF-001: Temps de génération d'exercices : < 30 secondes pour 10 exercices

RNF-002: Temps de traduction : < 2 minutes pour un document de 20 pages

RNF-003: Temps de réponse API : < 500ms pour 95% des requêtes

RNF-004: Support de 100 utilisateurs simultanés minimum

6.2 Sécurité

RNF-010: Authentification JWT avec expiration à 24h

RNF-011: Hashage des mots de passe avec bcrypt

RNF-012: HTTPS obligatoire en production

RNF-013: Validation et sanitization de tous les inputs

RNF-014: Protection CORS configurée

6.3 Disponibilité et Fiabilité

RNF-020: Disponibilité : 99,5% (downtime maximum 3,65 heures/mois)

RNF-021: Sauvegarde automatique de la base de données quotidienne

RNF-022: Gestion des erreurs avec logs détaillés

RNF-023: Reprise sur erreur pour les tâches asynchrones

6.4 Utilisabilité

RNF-030: Interface responsive (desktop, tablette, mobile)

RNF-031: Temps d'apprentissage : < 15 minutes pour les fonctions de base

RNF-032: Messages d'erreur clairs et en français

RNF-033: Feedback visuel pour toutes les actions longues

7. Architecture Technique

7.1 Architecture globale

GenEX-APP adopte une architecture client-serveur moderne avec séparation Frontend / Backend et intégration de services IA externes.

[Architecture globale du système]

7.2 Stack Technique

| | |
|------------------------------|---|
| Backend | FastAPI (Python 3.10+), Uvicorn/Gunicorn |
| Base de données | MySQL 8.0+ avec SQLAlchemy ORM |
| IA / Machine Learning | Google Gemini API, DeepSeek API |
| OCR | Tesseract 4.0+ avec PyTesseract |
| Génération PDF | WeasyPrint, ReportLab |
| Traduction | Deep Translator (Google Translate) |

7.3 Architecture Backend

Le backend suit une architecture en couches :

- **API Layer (FastAPI):** Gestion des endpoints REST, validation des données
- **Service Layer:** Logique métier, orchestration des services IA
- **Data Access Layer:** ORM SQLAlchemy, CRUD operations
- **External Services:** Gemini, DeepSeek, Tesseract, Deep Translator

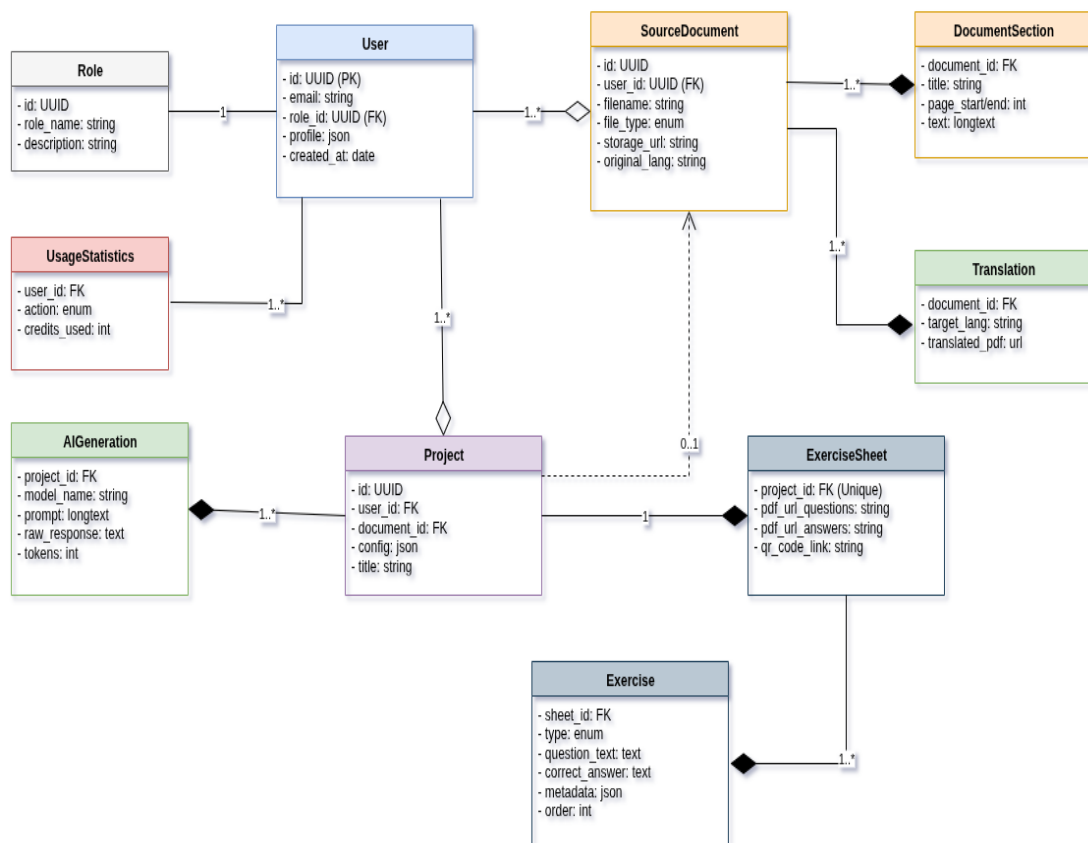
[Architecture en couches du Backend]

7.4 Base de Données

Principales tables :

- users : Gestion des utilisateurs et crédits
- roles : Rôles et permissions
- source_documents : Documents uploadés
- projects : Projets de génération d'exercices
- exercise_sheets : Feuilles d'exercices générées
- exercises : Exercices individuels
- translations : Historique des traductions
- usage_statistics : Statistiques d'utilisation

[Modèle Conceptuel de Données (classUML)]



8. Workflows Principaux

8.1 Workflow : Génération d'exercices

1. L'enseignant se connecte à la plateforme
2. Upload d'un document source (PDF, DOCX, image)
3. Le système extrait le texte via OCR
4. L'IA (Gemini) nettoie et structure le texte extrait
5. L'enseignant configure : types d'exercices, nombre, difficulté
6. Le système crée un projet et lance la génération asynchrone
7. L'IA (Gemini/DeepSeek) génère les exercices au format JSON
8. Le système parse et valide les exercices
9. Génération de 2 PDF : version questions + version corrigé
10. L'enseignant télécharge les PDF et/ou accède via QR code

[Diagramme de séquence - Génération d'exercices]

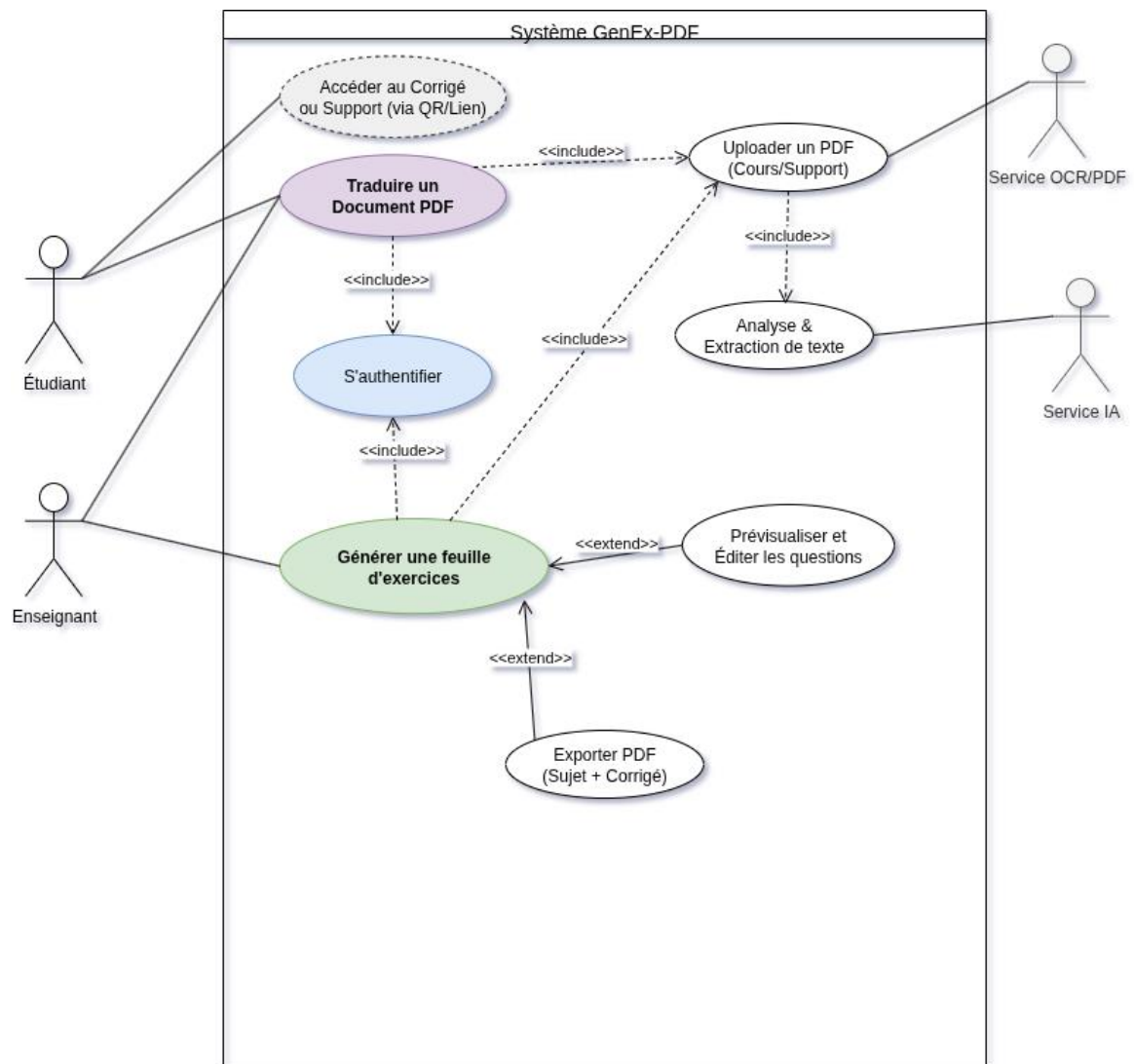
8.2 Workflow : Traduction de PDF

1. L'utilisateur upload un PDF à traduire
2. Le système extrait le texte de chaque page (PyPDF2)
3. Détection automatique de la langue source
4. L'utilisateur sélectionne la langue cible
5. Traduction asynchrone page par page (Deep Translator)
6. Reconstruction du PDF avec le texte traduit (WeasyPrint)
7. L'utilisateur est notifié de la fin de traduction
8. Téléchargement du PDF traduit

[Diagramme d'activité - Traduction PDF]

9. Cas d'Utilisation Détaillés

[Diagramme de cas d'utilisation UML]



9.1 UC-01 : Créer un compte utilisateur

Acteur principal : Enseignant

Préconditions : Aucune

Postconditions : Un compte utilisateur est créé avec 10 crédits gratuits

Scénario nominal :

1. L'utilisateur accède à la page d'inscription

2. Saisit son email et un mot de passe (8 caractères minimum)
3. Accepte les conditions d'utilisation
4. Le système valide les données
5. Un compte est créé avec le rôle 'Enseignant'
6. L'utilisateur est redirigé vers le tableau de bord

Scénarios alternatifs :

- Email déjà utilisé → Message d'erreur
- Mot de passe trop faible → Message de validation

9.2 UC-02 : Générer des exercices

Acteur principal : Enseignant authentifié

Préconditions : Avoir au moins 1 crédit disponible

Postconditions : Des exercices sont générés et téléchargeables en PDF

Scénario nominal :

7. L'enseignant upload un document (cours, chapitre)
8. Le système extrait le contenu
9. L'enseignant configure : 5 QCM + 3 questions ouvertes
10. Validation et lancement de la génération
11. Affichage de la progression (traitement asynchrone)
12. Génération réussie : 2 PDF disponibles (questions + corrigés)
13. Déduction d'1 crédit du compte

Scénarios alternatifs :

- Crédits insuffisants → Blocage et proposition d'achat
- Erreur IA → Retry automatique puis notification d'échec
- Document vide → Message d'erreur explicite

9.3 UC-03 : Traduire un document PDF

Acteur principal : Enseignant authentifié

Préconditions : Avoir un PDF à traduire

Postconditions : Un PDF traduit est disponible au téléchargement

Scénario nominal :

14. L'utilisateur upload le PDF source
15. Sélectionne la langue cible (auto-détection de la source)
16. Lance la traduction
17. Suivi de la progression (barre de progression)
18. Téléchargement du PDF traduit
19. Le PDF original et traduit restent accessibles dans l'historique

Scénarios alternatifs :

- PDF scanné avec OCR difficile → Message d'avertissement qualité
- Langue non supportée → Liste des langues disponibles

| Risque | Probabilité | Impact | Mitigation |
|----------------------|-------------|----------|-----------------------------------|
| API IA indisponible | Faible | Critique | Système de fallback + cache |
| Qualité insuffisante | Moyenne | Élevé | Validation humaine + feedback |
| Surcharge serveur | Moyenne | Moyen | Auto-scaling + rate limiting |
| Faible de sécurité | Faible | Critique | Audits réguliers + tests sécurité |
| Coût API dépassé | Moyenne | Élevé | Monitoring + alertes budgétaires |

10. Contraintes et Risques

10.1 Contraintes techniques

- **Dépendance aux API tierces** : Les services Gemini et DeepSeek doivent être disponibles et performants
- **Limitation des API gratuites** : Quotas d'appels API limitent le nombre de générations
- **Qualité de l'OCR** : La qualité d'extraction dépend de la qualité du PDF source
- **Taille des fichiers** : Upload limité à 50 Mo par fichier pour éviter les timeouts

10.2 Contraintes réglementaires

- **RGPD** : Consentement utilisateur, droit à l'oubli, portabilité des données
- **Propriété intellectuelle** : Les contenus générés appartiennent à l'utilisateur
- **Hébergement** : Serveurs conformes à la réglementation européenne

10.3 Analyse des risques

11. Planning Prévisionnel

Le développement suivra une approche agile en sprints de 2 semaines. Durée totale estimée : 4 mois.

11.1 Phases du projet

| Phase | Durée | Livrables clés |
|--------------------------|------------|--|
| Phase 1: Inception | 2 semaines | Cahier des charges, maquettes, architecture |
| Phase 2: Élaboration | 4 semaines | Backend API, base de données, authentification |
| Phase 3: Construction | 6 semaines | Modules de génération et traduction, tests |
| Phase 4: Transition | 2 semaines | Déploiement, documentation, formation |
| Maintenance | Continue | Corrections bugs, nouvelles fonctionnalités |

12. Assurance Qualité

12.1 Stratégie de tests

- **Tests unitaires:** Couverture $> 80\%$ du code backend (pytest)
- **Tests d'intégration:** Validation des workflows complets (pytest + fixtures)
- **Tests de performance:** Load testing avec 100+ utilisateurs simultanés (Locust)
- **Tests de sécurité:** Scan des vulnérabilités (OWASP ZAP)
- **Tests utilisateurs:** Tests d'acceptation avec enseignants (Sessions guidées)

12.2 Critères d'acceptation

- Tous les tests unitaires passent (100%)
- Couverture de code backend $\geq 80\%$
- Temps de génération $< 30s$ (95% des cas)
- Aucune faille de sécurité critique (audit OWASP)
- Validation utilisateur : 8/10 de satisfaction minimum

12.3 Gestion des bugs

Classification des bugs par sévérité :

- **Critique** : Blocage complet du système → Correction immédiate
- **Majeur** : Fonctionnalité principale inutilisable → Correction sous 48h
- **Mineur** : Gêne sans blocage → Correction prochaine release
- **Cosmétique** : Problème visuel mineur → Backlog

13. Déploiement et Exploitation

13.1 Environnements

- **Développement** : Local : machines des développeurs → pip install + uvicorn
- **Staging** : Serveur de pré-production → Docker + Gunicorn
- **Production** : Serveur cloud (AWS/GCP/Azure) → Docker + Kubernetes + Nginx

13.2 Architecture de déploiement

- Load Balancer (Nginx) : Répartition de charge
- Application Servers : 2+ instances FastAPI (Gunicorn + Uvicorn workers)
- Base de données : MySQL répliquée (master-slave)
- Storage : S3 ou équivalent pour les fichiers (PDF, documents)
- Cache : Redis pour les sessions et rate limiting

13.3 Procédure de déploiement

20. Vérification des tests (CI/CD pipeline)
21. Build de l'image Docker
22. Déploiement sur environnement staging
23. Tests de smoke (santé des endpoints)
24. Validation par le QA
25. Déploiement progressif en production (blue-green)
26. Monitoring post-déploiement (24h)

13.4 Monitoring et alertes

- Uptime et disponibilité du service
- Temps de réponse des API endpoints
- Nombre de requêtes par minute

- Taux d'erreur (4xx, 5xx)
- Usage CPU et mémoire
- Santé de la base de données (connexions, temps de requête)
- Coûts des API tierces (Gemini, DeepSeek)

15. Glossaire

API : Application Programming Interface - Interface de programmation

IA : Intelligence Artificielle

JWT : JSON Web Token - Token d'authentification sécurisé

LLM : Large Language Model - Modèle de langage de grande taille

OCR : Optical Character Recognition - Reconnaissance optique de caractères

ORM : Object-Relational Mapping - Mappage objet-relationnel

PDF : Portable Document Format - Format de document portable

QCM : Questions à Choix Multiples

REST : Representational State Transfer - Architecture d'API

RGPD : Règlement Général sur la Protection des Données

UX/UI : User Experience / User Interface - Expérience et interface utilisateur

16. Annexes

Annexe A : Références API

- **Google Gemini API:** <https://ai.google.dev/gemini-api/docs>
- **DeepSeek API:** <https://platform.deepseek.com/docs>
- **FastAPI Documentation:** <https://fastapi.tiangolo.com/>
- **Tesseract OCR:** <https://github.com/tesseract-ocr/tesseract>

Annexe B : Schémas de la base de données

[INSÉRER : Schéma détaillé des tables avec types de données]

Annexe C : Exemples de requêtes API

Exemple d'authentification :

```
POST /api/genex/auth/login
{
  "email": "prof@example.com",
  "password": "SecurePass123"
}
```

Exemple de génération d'exercices :

```
POST /api/genex/create/projects
{
  "document_id": "uuid",
  "title": "Exercices Chapitre 3",
  "config": {"exercises": {"total": 10, "types": [...]}}
}
```

Conclusion

GenEX-APP représente une solution innovante pour moderniser la préparation pédagogique en tirant parti des technologies d'intelligence artificielle. Le projet répond à un besoin réel exprimé par les enseignants et s'inscrit dans une démarche d'optimisation du temps et de diversification des approches pédagogiques.

Les technologies choisies (FastAPI, MySQL, Gemini/DeepSeek, Tesseract, WeasyPrint, ReactJS, Tailwindcss) offrent un équilibre entre performance, fiabilité et coûts maîtrisés. L'architecture modulaire permet une évolution progressive avec l'ajout de nouvelles fonctionnalités.

Prochaines étapes

27. Validation du cahier des charges avec les parties prenantes
28. Conception détaillée des maquettes d'interface
29. Mise en place de l'environnement de développement
30. Sprint 1 : Développement du backend (authentification + base de données)
31. Tests utilisateurs alpha avec un groupe d'enseignants pilotes

Ce cahier des charges constitue le document de référence pour le développement de GenEX-APP. Il évoluera selon les retours utilisateurs et les contraintes techniques identifiées durant les phases de construction.