المندوبية السامية للتخطيط
HAUT-COMMISSARIAT AU PLAN

INSEA
Institut National de Statistique
Et d'Economie Appliquée

Rapport du Cahier des Charges : Logiciel de Gestion de Conférences Académiques

Preparée par :

Khawla Hamma

Mayssoune Ouikrim

Fatima Zahra Kasmi

1. Résumé Exécutif

Le **Cahier des Charges** définit les spécifications pour une application de bureau développée avec **.NET 6.0**, **WPF**, et **MYSQL**, visant à gérer des conférences académiques. L'objectif est de fournir une solution intuitive, sécurisée et performante pour les organisateurs, participants, intervenants, et administrateurs, avec des fonctionnalités telles que l'inscription, la planification, les paiements, et les notifications. Le projet adopte un modèle incrémental, avec une durée estimée de **12 à 19 semaines** (incluant une marge de 10 %). Ce rapport résume les besoins, le plan de développement, les risques, et propose des recommandations pour garantir le succès du projet.

2. Objectifs du Projet

- **Objectif principal** : Développer une application de bureau pour planifier, gérer, et suivre des conférences académiques, en présentiel ou à distance.
- **Public cible** : Organisateurs, participants, intervenants, administrateurs.
- **Fonctionnalités clés** :
    - Inscription/authentification sécurisée avec MFA.
    - Gestion des conférences (CRUD, agenda interactif, visioconférences).
    - Gestion des documents (upload/download de supports).
    - Paiements via Stripe et gestion des sponsors.
    - Notifications automatisées (email/SMS via SendGrid/Twilio).
    - Suivi des KPI (inscriptions, satisfaction, revenus).
- **Exigences non fonctionnelles** :
    - Temps de réponse < 500 ms pour 1000 utilisateurs.
    - Sécurité (AES-256, RGPD, MFA).
    - Accessibilité (WCAG 2.1, niveau AA).
    - Compatibilité Windows 10+ (64 bits).
    - Scalabilité jusqu'à 5000 utilisateurs et 100 conférences.

3. Analyse des Besoins Clients

Les besoins clients, identifiés dans le cahier, sont structurés par rôle :

- **Participants** : Inscription facile, accès aux supports, agenda personnalisé.
- **Intervenants** : Soumission de documents, gestion de profil, planification des interventions.
- **Organisateurs** : Création de conférences, gestion financière, envoi de notifications.
- **Administrateurs** : Supervision des utilisateurs, configuration système, suivi des performances.

Ces besoins sont traduits en **exigences fonctionnelles**, **non fonctionnelles techniques**, **d'interface**, et **de tests**, garantissant une couverture complète des attentes.

4. Spécifications Techniques

- **Frontend** : WPF avec Material Design In XAML pour une interface moderne, multilingue (français, anglais), et accessible.
- **Backend** : .NET 6.0 avec Entity Framework Core pour la logique métier, suivant le modèle MVVM.
- **Base de données** : MYSQL
- **Intégrations** :
    - **Visioconférence** : API Zoom/Teams.
    - **Paiements** : Stripe.
    - **Notifications** : SendGrid (email), Twilio (SMS).
    - **Graphiques** : LiveCharts2 pour les KPI.
- **Environnement** : Windows 10+ (64 bits), 500 Mo d'espace disque, 4 Go RAM minimum.

Le dictionnaire de données définit 7 tables principales (Utilisateur, Conference, Intervention, Salle, Sponsor, Paiement, Notification), avec des types adaptés à MYSQL

5. Plan de Développement

Le développement suit un **modèle incrémental**, avec 5 incréments livrant progressivement les fonctionnalités. Chaque incrément inclut les étapes : Spécification, Conception, Programmation, Validation et Vérification (V&V), et Livraison.

5.1 Incréments

1. **Base de Données** (2-3 semaines)
    - Objectif : Créer une base MYSQL.
    - Livrables : Schéma documenté, base intégrée avec migrations Entity Framework Core.
    - Outils : Draw.io, DBeaver, Eclipse, xUnit.
2. **Interface Utilisateur** (3-4 semaines)
    - Objectif : Développer une interface WPF responsive.
    - Livrables : Interface fonctionnelle avec vues XAML et ViewModels.
    - Outils : Figma, Eclipse, Accessibility Insights.
3. **Gestion des Conférences** (2-3 semaines)
    - Objectif : Implémenter les fonctionnalités CRUD et l'agenda.
    - Livrables : Module intégré avec contrôles WPF personnalisés.
    - Outils : Lunacy, Enterprise Architect, Eclipse, BenchmarkDotNet.
4. **Paiements et Sponsoring** (3-4 semaines)
    - Objectif : Gérer les transactions via Stripe et les sponsors.
    - Livrables : Système intégré avec génération de factures (PdfSharp).
    - Outils : Lunacy, Enterprise Architect, Eclipse, OWASP ZAP.
5. **Notifications** (2-3 semaines)
    - Objectif : Automatiser les communications (email/SMS).
    - Livrables : Système intégré avec scheduler Quartz.NET.
    - Outils : Lunacy, Draw.io, Eclipse, SendGrid/Twilio logs.

5.2 Planning

- **Durée totale** : 12-17 semaines (13-19 avec marge de 10 %).

- **Rôles** : Analyste, Architecte, Designer UX/UI, Développeur, Testeur, Administrateur système.
- **Dépendances** :
    - Incrément 2 dépend de l'Incrément 1 (base de données).
    - Incréments 3-5 dépendent de l'Incrément 2 (interface).

5.3 Outils AGL

- **UML** : Enterprise Architect pour les diagrammes (cas d'utilisation, classes, séquences, états).
- **UX/UI** : Lunacy pour les maquettes.
- **IDE** : Eclipse pour le développement.

6. Conception et Documentation

6.1 UML

- **Cas d'utilisation** : Couvre les interactions des acteurs (Utilisateur, Conférencier, Administrateur) avec des fonctionnalités comme consulter, gérer, et soumettre.
- **Classes** : 8 classes principales (Utilisateur, Conference, Intervention, etc.) avec relations 1-n et 1-1.
- **Séquences** : Scénarios pour inscription, planification, et paiements.
- **États** : Cycles de vie pour Conference (Créée → Archivée) et Intervention (Proposée → Présentée).

6.2 UX/UI

- **Pages clés** : Accueil, dashboards (par rôle), agenda interactif, interfaces de paiement et notifications.
- **Prototypes** : Maquettes Lunacy validées par tests utilisateurs.
- **Design** : Material Design, thèmes clair/sombre, conformité WCAG 2.1.

6.3 Tests

- **Unitaires** : Authentification (MFA), paiements (Stripe), notifications (multi-canal).
- **Intégration** : Interactions entre modules.
- **Performance** : Temps de réponse < 500 ms sous charge.
- **Sécurité** : Audit OWASP (injections SQL, XSS).
- **Accessibilité** : Tests avec axe/WAVE.
- **Utilisateurs** : Validation ergonomique avec 5 participants.

7. Sécurité et Conformité

- **Sécurité** : Chiffrement AES-256, MFA obligatoire, audit OWASP.
- **RGPD** : Consentement explicite, droit à l'oubli, sauvegardes quotidiennes.
- **Accessibilité** : WCAG 2.1 (niveau AA), contraste élevé, navigation clavier.

8. Maintenance et Support

- **Maintenance** : Mises à jour trimestrielles, correction de bugs.
- **Support** : Assistance par email, SLA de 48h.

9. Risques et Mitigation

Risque	Probabilité	Impact	Mitigation
Défaillance visioconférence	Moyenne	Élevé	Tester API Zoom/Teams, lien manuel en secours.
Retard développement	Moyenne	Modéré	Marge de 10 %, priorisation des incréments.
Fuite de données	Faible	Critique	Chiffrement AES-256, audit RGPD, tests OWASP.
Non-conformité accessibilité	Faible	Modéré	Tests automatisés (axe/WAVE), validation WCAG.

10. Indicateurs de Performance (KPI)

- **Taux d'inscription** : % de places remplies.
- **Satisfaction moyenne** : Échelle 0-5 via enquêtes.
- **Temps de réponse** : < 500 ms pour l'interface.

11. Recommandations

1. **Priorisation des incréments** : Commencer par la base de données et l'interface pour obtenir un prototype fonctionnel rapidement, facilitant les retours utilisateurs.
2. **Tests rigoureux** :
    - Effectuer des tests de charge précoces pour valider la scalabilité MYSQL(5000 utilisateurs).
    - Renforcer les tests de sécurité avant l'intégration de Stripe.
3. **Formation** : Former les organisateurs dès l'Incrément 3 pour anticiper l'adoption.
4. **Accessibilité** : Intégrer un expert WCAG dès la conception UX pour minimiser les ajustements.
5. **Documentation** : Maintenir une documentation évolutive (Markdown) pour chaque incrément, facilitant la maintenance.

12. Conclusion

Le **Cahier des Charges** fournit une feuille de route claire pour développer une application de gestion de conférences robuste et utilisateur-centrée. Le modèle incrémental, combiné à une architecture .NET/WPF/MYSQL, garantit une livraison progressive et adaptable. En suivant les recommandations et en gérant proactivement les risques, le projet a de fortes chances d'atteindre ses objectifs dans les délais impartis (13-19 semaines). Une attention particulière aux tests (sécurité, accessibilité) et à la formation des utilisateurs sera essentielle pour assurer une adoption réussie.