

Cahier des Charges pour le Logiciel de Gestion de Conférences Académiques

Filiere : data and software engineering

Preparée par :

Khawla Hamma

Mayssoune Ouikrim

Fatima Zahra Kasmi

2025 /2026

Table des Matières

- 1. Introduction**
- 2. Besoins Clients et Spécifications du Logiciel**
 - 2.1 Besoins Clients
 - 2.2 Spécifications Fonctionnelles
 - 2.3 Spécifications Techniques
 - 2.4 Dictionnaire de Données
- 3. Modèle du Cycle de Vie du Logiciel**
 - 3.1 Présentation du Modèle Incrémental
 - 3.2 Justification du Choix
 - 3.3 Application du Modèle Incrémental
- 4. Planning Prévisionnel**
- 5. Sécurité et Confidentialité**
- 6. Accessibilité**
- 7. Maintenance et Support Post-Déploiement**
- 8. Indicateurs de Performance (KPI)**
- 9. Conception et Documentation via AGL**
 - 9.1 Choix des AGL
 - 9.2 Conception UML
 - 9.3 Conception UX/UI(**DESIGN UX & LES PAGES REELES DU APPLICATION**)
 - 9.4 Conception des Tests Unitaires et Résultat
 - 9.5 Modèle Physique de la Base de Données
- 10.Développement d'une Sous-Partie du Logiciel**
- 11.Analyse des Risques et Mitigation**
- 12.Glossaire**

1. Introduction

Ce cahier des charges définit les besoins des parties prenantes et les spécifications d'un logiciel de gestion de conférences académiques, implémenté comme une application de bureau avec **.NET 6.0**, **WPF**, et **MySQL**. L'objectif est de fournir une solution intuitive, sécurisée et efficace pour permettre aux organisateurs de planifier, gérer et suivre des événements, tout en offrant une expérience fluide aux participants et intervenants.

2. Besoins Clients et Spécifications du Logiciel

Besoins Clients

- **Gestion des Participants** : Interface pour l'inscription avec collecte sécurisée des données (nom, email, affiliation). Suivi en temps réel des inscriptions via un tableau de bord.
- **Gestion des Intervenants** : Enregistrement des intervenants (biographie, sujet), gestion des résumés et supports, planification des créneaux horaires sans conflits.
- **Gestion des Organisateurs** : Comptes avec rôles définis (administrateur, logistique) et outil de gestion des tâches.
- **Gestion des Conférences** : Planification avec agenda interactif et intégration de visioconférences (Zoom, Teams).
- **Gestion Financière** : Gestion des frais d'inscription, suivi des transactions, génération de factures, enregistrement des sponsors.

Spécifications des Exigences

Exigences Fonctionnelles

- Inscription et authentification des utilisateurs (participants, intervenants, organisateurs, administrateurs) avec multi-facteurs (MFA).
- Création, modification et suppression des conférences (titre, date, lieu, lien visioconférence).
- Planification des interventions avec attribution des intervenants, salles et horaires, évitant les conflits.
- Gestion des documents : upload/download des supports (PDF/PPT) par intervenants, accès sécurisé pour participants.
- Envoi de notifications automatisées (email, SMS) pour confirmations, rappels (24h avant session) et mises à jour.
- Gestion des paiements : traitement des frais d'inscription via Stripe, génération de factures PDF.
- Enregistrement des sponsors avec suivi des contributions financières.
- Recherche avancée des conférences avec filtres (date, thème, intervenant).

- Suivi des KPI : taux d'inscription, satisfaction moyenne, revenus totaux.

Exigences Non Fonctionnelles Techniques

- **Performance** : Temps de réponse de l'interface < 500 ms pour 1000 utilisateurs simultanés.
- **Sécurité** : Chiffrement AES-256 des données sensibles, conformité RGPD (consentement, droit à l'oubli), MFA obligatoire.
- **Compatibilité** : Fonctionnement sur Windows 10 ou ultérieur (64 bits).
- **Scalabilité** : Supporte jusqu'à 5000 utilisateurs et 100 conférences par base MySQL.
- **Sauvegarde** : Sauvegardes automatiques quotidiennes de la base MySQL avec restauration en < 1h.
- **Disponibilité** : Disponibilité de l'application à 99,9 % (hors maintenance).

Exigences d'Interface

- Interface moderne avec Material Design, supportant les thèmes clair/sombre.
- Responsive pour différentes résolutions (minimum 1024x768).
- Accessibilité conforme WCAG 2.1 (niveau AA) : contraste élevé, navigation clavier.
- Support multilingue (français, anglais).
- Tableau de bord utilisateur personnalisé par rôle (participant, intervenant, organisateur, administrateur).
- Agenda interactif avec vues mensuelle/hebdomadaire/journalière et filtres (conférence, intervenant, salle).
- Formulaire intuitifs pour inscription, soumission de documents, et paiements.

Exigences de Tests

- Tests unitaires pour l'authentification (MFA), les paiements (simulations Stripe), et les notifications (envoi multi-canal).
- Tests d'intégration pour vérifier les interactions entre modules (inscription, planification, notifications).
- Tests de performance pour valider le temps de réponse (< 500 ms) sous charge (1000 utilisateurs).
- Tests de sécurité : audit OWASP pour détecter les vulnérabilités (injections SQL, XSS).
- Tests d'accessibilité : conformité WCAG 2.1 via outils automatisés (axe, WAVE).
- Tests utilisateurs avec 5 participants pour valider l'ergonomie (inscription en < 2 min).

Spécifications Techniques

- **Frontend** : WPF avec Material Design In XAML pour une interface moderne.
- **Backend** : .NET 6.0 avec Entity Framework Core pour la logique métier.
- **Base de données** : MySQL .
- **Environnement** : Windows 10 ou ultérieur, 500 Mo d'espace disque, 4 Go RAM minimum.
- **Visioconférence** : Intégration des API Zoom/Teams pour les liens de sessions virtuelles.
- **Notifications** : Intégration SendGrid (email) et Twilio (SMS).
- **Paiements** : Intégration Stripe pour les transactions.

- **Graphiques** : LiveCharts2 pour les visualisations de KPI.
- **Architecture** : MVVM (Model-View-ViewModel) pour une séparation claire.

2.4 Dictionnaire de Données

| Table | Champ | Type | Description |
|--------------|---------------------|--------------|---|
| Utilisateur | id | INT (PK) | Identifiant unique |
| | nom | VARCHAR(255) | Nom complet |
| | email | VARCHAR(255) | Email unique |
| | role | ENUM | Rôle (organisateur, participant, intervenant, administrateur) |
| | motDePasse | VARCHAR(255) | Mot de passe hashé (SHA-256) |
| Conference | id | INT (PK) | Identifiant unique |
| | titre | VARCHAR(255) | Titre de la conférence |
| | date | DATETIME | Date et heure |
| | organisateurId | INT (FK) | Référence à l'organisateur |
| | visioconferenceLien | VARCHAR(255) | Lien Zoom/Teams |
| Intervenant | id | INT (PK) | Identifiant unique |
| | conferenceId | INT (FK) | Référence à la conférence |
| | intervenantId | INT (FK) | Référence à l'intervenant |
| | horaire | DATETIME | Date et heure de l'intervention |
| | salleId | INT (FK) | Référence à la salle |
| | supportPresentation | VARCHAR(255) | Lien vers le support |
| Salle | id | INT (PK) | Identifiant unique |
| | nom | VARCHAR(255) | Nom de la salle |
| | capacite | INT | Capacité maximale |
| | accessibilite | BOOLEAN | Accessibilité PMR |
| Sponsor | id | INT (PK) | Identifiant unique |
| | nom | VARCHAR(255) | Nom du sponsor |
| | montant | DECIMAL | Montant du sponsoring |
| | conferenceId | INT (FK) | Référence à la conférence |
| Paieement | id | INT (PK) | Identifiant unique |
| | utilisateurId | INT (FK) | Référence à l'utilisateur |
| | montant | DECIMAL | Montant payé |
| | statut | ENUM | Statut (en attente, complété, annulé) |
| Notification | dateTransaction | DATETIME | Date de la transaction |
| | id | INT (PK) | Identifiant unique |
| | utilisateurId | INT (FK) | Référence au destinataire |
| | type | ENUM | Type (email, SMS, push) |

| Table | Champ | Type | Description |
|-------|-----------|----------|----------------------------|
| | contenu | TEXT | Contenu de la notification |
| | dateEnvoi | DATETIME | Date d'envoi |

3. 3. Modèle du Cycle de Vie du Logiciel

3.1 Présentation du Modèle Incrémental

Le développement adoptera un modèle incrémental, où chaque itération (incrément) ajoute des fonctionnalités testées et validées à la version précédente. Ce modèle permet une progression modulaire, avec des livraisons intermédiaires pour évaluer les résultats et ajuster les priorités.

3.2 Justification du Choix

- **Livraison rapide** : Fournir des versions utilisables dès les premiers incréments.
- **Flexibilité** : Adapter les fonctionnalités aux retours des utilisateurs.
- **Réduction des risques** : Identifier et corriger les problèmes via des tests progressifs.

3.3 Application du Modèle Incrémental

Chaque incrément suit un cycle structuré en cinq étapes : **Spécification, Conception, Programmation, Validation et Vérification (V&V), et Livraison**. Voici les détails pour chaque incrément.

Incréments du Cycle de Vie du Logiciel

Incrément 1 : Base de Données

- **Objectif** : Créer une base relationnelle robuste avec MySQL pour stocker les données du système (utilisateurs, conférences, interventions, etc.).
- **Étapes** :
 1. **Spécification**
 - **Activités** : Analyse des besoins fonctionnels (gestion des utilisateurs, conférences, paiements) pour identifier les entités principales (Utilisateur, Conference, Intervention, etc.) et leurs relations (1-n, n-n). Rédaction d'un document précisant les champs requis, leurs types, et les contraintes (unicité, non-nullité).
 - **Outils** : Tableaux Excel pour lister les entités, réunions avec les parties prenantes via Zoom.
 - **Durée** : 3 jours.
 - **Résultat** : Document d'analyse des entités et relations.
 2. **Conception**
 - **Activités** : Création d'un modèle conceptuel de données (MCD) avec diagrammes UML (diagrammes de classes) via Draw.io. Définition du schéma relationnel MySQL : tables, clés primaires/étrangères, index

(ex. : index sur email pour recherches rapides). Planification des contraintes d'intégrité (ex. : NOT NULL sur titre dans Conference).

- **Outils** : Draw.io, DBeaver (pour visualiser MySQL).
- **Durée** : 5 jours.
- **Résultat** : Schéma relationnel documenté.

3. Programmation

- **Activités** : Écriture des scripts SQL pour créer les tables (ex. : `CREATE TABLE Conference (...)`), implémenter les relations (clés étrangères), et ajouter les contraintes. Configuration initiale de la base MySQL avec un jeu de données fictif (10 utilisateurs, 5 conférences) via Entity Framework Core migrations.
- **Outils** : eclipse , Entity Framework Core, MySQL.
- **Durée** : 5 jours.
- **Résultat** : Base de données fonctionnelle.

4. Validation et Vérification (V&V)

- **Activités** : Tests unitaires pour vérifier l'intégrité (insertion sans doublons sur email), la cohérence des relations (suppression en cascade), et la performance (temps de réponse < 1s pour 1000 enregistrements). Utilisation de requêtes SQL pour simuler des scénarios (ex. : `SELECT * FROM Conference WHERE date > CURRENT_TIMESTAMP`).
- **Outils** : DBeaver, xUnit pour tests unitaires.
- **Critères** : 100 % des contraintes respectées, zéro erreur d'intégrité.
- **Durée** : 3 jours.
- **Résultat** : Rapport de tests validé.

5. Livraison

- **Activités** : Intégration de la base MySQL dans le projet .NET. Rédaction d'une documentation technique (structure des tables, relations, accès via EF Core). Formation rapide des développeurs sur les migrations et requêtes MySQL
- **Outils** : eclipse
- **Durée** : 2 jours.
- **Résultat** : Base intégrée avec documentation.
- **Livrables** : Schéma documenté, base de données MySQL .

Incrément 2 : Interface Utilisateur

- **Objectif** : Développer une interface utilisateur moderne et responsive avec WPF pour permettre aux utilisateurs d'interagir avec le système.
- **Étapes** :
 1. **Spécification**
 - **Activités** : Identification des besoins utilisateurs (navigation simple, formulaires d'inscription, agenda visuel). Rédaction d'un cahier des charges UX précisant les vues (Accueil, Inscription, Agenda) et les interactions (boutons, champs). Recueil des retours via un atelier avec 5 utilisateurs potentiels.
 - **Outils** : Google Docs, ateliers Zoom.
 - **Durée** : 5 jours.
 - **Résultat** : Spécifications UX/UI détaillées.
 2. **Conception**

- **Activités** : Création de wireframes et maquettes haute-fidélité avec Figma, utilisant Material Design In XAML. Définition d'un design system (couleurs, typographie, contrôles). Planification de la responsivité pour différentes résolutions (minimum 1024x768). Validation des maquettes par un focus group.
 - **Outils** : Figma, Material Design In XAML Toolkit.
 - **Durée** : 7 jours.
 - **Résultat** : Maquettes validées.
3. **Programmation**
- **Activités** : Développement en WPF avec XAML pour les vues (ex. : `MainWindow.xaml`, `InscriptionView.xaml`). Implémentation des ViewModels avec CommunityToolkit.Mvvm pour la logique MVVM. Intégration de données fictives via un service mock avant connexion à la base MySQL
 - **Outils** : Visual Studio 2022, WPF, CommunityToolkit.Mvvm.
 - **Durée** : 7 jours.
 - **Résultat** : Interface fonctionnelle en local.
4. **Validation et Vérification (V&V)**
- **Activités** : Tests sur différentes résolutions Windows (1024x768 à 4K). Tests d'ergonomie avec 3 utilisateurs (temps pour inscription < 2 min). Vérification de l'accessibilité (WCAG 2.1, navigation clavier). Utilisation d'outils comme Accessibility Insights pour un score > 90.
 - **Outils** : Accessibility Insights, xUnit pour tests UI.
 - **Critères** : 100 % compatibilité, score UX > 85 %.
 - **Durée** : 5 jours.
 - **Résultat** : Rapport de tests et corrections.
5. **Livraison**
- **Activités** : Intégration de l'interface dans le projet .NET. Rédaction d'un guide utilisateur pour la navigation. Présentation aux parties prenantes pour validation finale.
 - **Outils** : Eclipse, Markdown.
 - **Durée** : 3 jours.
 - **Résultat** : Interface intégrée et documentée.
- **Livrables** : Interface WPF fonctionnelle.

Incrément 3 : Gestion des Conférences

- **Objectif** : Permettre la création et la gestion des événements (conférences, sessions).
- **Étapes** :
 1. **Spécification**
 - **Activités** : Définition des fonctionnalités CRUD pour les conférences (ajouter, lire, modifier, supprimer) et de l'agenda interactif. Rédaction des cas d'utilisation (ex. : "Organisateur crée une conférence"). Identification des contraintes (ex. : pas de chevauchement d'horaires).
 - **Outils** : Google Docs, architect entreprise.
 - **Durée** : 3 jours.
 - **Résultat** : Document fonctionnel.
 2. **Conception**
 - **Activités** : Conception de la logique métier avec des services .NET (ex. : `ConferenceService`). Design de l'agenda avec un contrôle WPF

personnalisé (inspiré de FullCalendar). Schéma des interactions avec la base MySQL via Entity Framework Core (requêtes pour création/modification).

- **Outils** : lunacy, Eclipse.
- **Durée** : 5 jours.
- **Résultat** : Schéma de services et design agenda.

3. Programmation

- **Activités** : Implémentation backend avec .NET 6.0 pour la logique métier. Connexion à la base MySQL via Entity Framework Core. Développement frontend de l'agenda en WPF avec contrôles personnalisés. Tests manuels en local.
- **Outils** : Eclipse, Entity Framework Core, WPF.
- **Durée** : 5 jours.
- **Résultat** : Module fonctionnel en local.

4. Validation et Vérification (V&V)

- **Activités** : Tests fonctionnels (création réussie d'une conférence, modification sans erreur). Tests de performance (temps de réponse < 500 ms). Simulation de 50 conférences pour vérifier l'agenda. Correction des bugs identifiés.
- **Outils** : xUnit, BenchmarkDotNet pour performance.
- **Critères** : 100 % des cas d'utilisation validés.
- **Durée** : 3 jours.
- **Résultat** : Rapport de tests.

5. Livraison

- **Activités** : Intégration du module dans le projet .NET. Documentation des fonctionnalités (CRUD, agenda) avec Markdown. Formation des organisateurs sur l'utilisation du module.
- **Outils** : Eclipse, Markdown.
- **Durée** : 2 jours.
- **Résultat** : Module intégré et documenté.

- **Livrables** : Module de gestion des conférences.

Incrément 4 : Paiements et Sponsoring

- **Objectif** : Gérer les transactions financières (frais d'inscription, sponsors).
- **Étapes** :

1. Spécification

- **Activités** : Analyse des besoins (paiement des frais, enregistrement des sponsors). Choix de Stripe comme passerelle (sécurité, facilité d'intégration). Définition des flux (paiement → confirmation → facture).
- **Outils** : Google Docs, réunions Zoom.
- **Durée** : 5 jours.
- **Résultat** : Spécifications paiement/sponsoring.

2. Conception

- **Activités** : Conception du workflow de paiement (formulaire WPF → Stripe → retour). Design des formulaires sponsors (nom, montant) en XAML. Schéma des entités associées (Paiement, Sponsor) pour MySQL via Entity Framework Core.
- **Outils** : lunacy, architect Entreprise.

- **Durée** : 7 jours.
- **Résultat** : Workflow et formulaires conçus.
- 3. **Programmation**
 - **Activités** : Intégration de l'API Stripe pour les paiements dans .NET. Développement des services pour gérer les sponsors. Ajout de la logique de facturation (PDF généré via PdfSharp). Tests manuels en local.
 - **Outils** : Visual Studio 2022, Stripe SDK, PdfSharp.
 - **Durée** : 7 jours.
 - **Résultat** : Système fonctionnel en local.
- 4. **Validation et Vérification (V&V)**
 - **Activités** : Tests de sécurité (aucune donnée sensible exposée, HTTPS pour Stripe). Tests fonctionnels (paiement réussi, facture générée). Simulation de 20 transactions pour vérifier la robustesse.
 - **Outils** : xUnit, OWASP ZAP (sécurité).
 - **Critères** : 100 % des paiements réussis, zéro faille critique.
 - **Durée** : 5 jours.
 - **Résultat** : Rapport de tests sécurité/fonctionnalité.
- 5. **Livraison**
 - **Activités** : Intégration du système dans le projet .NET. Documentation des processus de paiement et sponsoring. Formation des organisateurs sur la gestion financière.
 - **Outils** : Eclipse, Markdown.
 - **Durée** : 3 jours.
 - **Résultat** : Système intégré et documenté.
- **Livrables** : Système de paiement et gestion des sponsors.

Incrément 5 : Notifications

- **Objectif** : Automatiser les communications avec les utilisateurs (email, SMS).
- **Étapes** :
 1. **Spécification**
 - **Activités** : Définition des types de notifications (confirmation, rappel) et des déclencheurs (inscription, 24h avant événement). Analyse des besoins de personnalisation (nom, date). Choix de SendGrid (email) et Twilio (SMS).
 - **Outils** : Google Docs, tableaux décisionnels.
 - **Durée** : 3 jours.
 - **Résultat** : Spécifications des notifications.
 2. **Conception**
 - **Activités** : Conception des templates (HTML pour email, texte pour SMS). Définition des services d'envoi dans .NET. Planification des déclencheurs via un scheduler (ex. : Quartz.NET).
 - **Outils** : lunacy (templates), Draw.io.
 - **Durée** : 5 jours.
 - **Résultat** : Templates et schéma d'envoi.
 3. **Programmation**
 - **Activités** : Intégration de SendGrid et Twilio via leurs SDK dans .NET. Développement des services pour gérer les notifications.

Implémentation d'un scheduler avec Quartz.NET pour les rappels automatiques.

- **Outils** : Visual Studio 2022, SendGrid SDK, Twilio SDK, Quartz.NET.
- **Durée** : 5 jours.
- **Résultat** : Système fonctionnel en local.

4. Validation et Vérification (V&V)

- **Activités** : Tests d'envoi (email reçu en < 5s, SMS en < 10s).
Vérification de la personnalisation (nom correct dans le message).
Simulation de 100 notifications pour tester la scalabilité.
- **Outils** : xUnit, logs SendGrid/Twilio.
- **Critères** : 98 % de délivrabilité, zéro erreur de format.
- **Durée** : 3 jours.
- **Résultat** : Rapport de tests.

5. Livraison

- **Activités** : Intégration du système dans le projet .NET avec scheduler actif. Documentation des templates et déclencheurs. Formation des organisateurs sur la personnalisation des notifications.
- **Outils** : Eclipse, Markdown.
- **Durée** : 2 jours.
- **Résultat** : Système intégré et documenté.

- **Livrables** : Notifications automatisées opérationnelles.

4. Planning Prévisionnel

Le projet est estimé entre **12 et 17 semaines**, avec une marge de 10 % (13-19 semaines).
Chaque incrément détaille la durée par étape et les rôles responsables.

| Incrément | Étape | Durée | Responsable | Tâches principales | Livrables | Dépendances |
|---------------------------------|----------------------------|---------------------|------------------------|---|--------------------|-------------|
| 1. Base de Données | Spécification | 3 jours | Analyste | Analyse des entités et besoins | Document d'analyse | Aucune |
| | Conception | 5 jours | Architecte BD | Modélisation UML, schéma MySQL | Schéma relationnel | - |
| | Programmation | 5 jours | Développeur BD | Création des tables et contraintes | Base de données | - |
| | Validation et Vérification | 3 jours | Testeur | Tests d'intégrité et cohérence | Rapport de tests | - |
| | Livraison | 2 jours | Administrateur système | Déploiement et documentation | Base déployée, doc | - |
| | Total | 2-3 semaines | | | | |
| 2. Interface Utilisateur | Spécification | 5 jours | Analyste/Designer UX | Identification des besoins utilisateurs | Spécifications UX | Incrément 1 |
| | Conception | 7 jours | Designer UI | Maquettage avec Figma | Maquettes Figma | - |
| | Programmation | 7 jours | Développeur Frontend | Développement React.js | Code frontend | - |
| | Validation et Vérification | 5 jours | Testeur | Tests responsive et ergonomie | Rapport de tests | - |

| Incrément | Étape | Durée | Responsable | Tâches principales | Livrables | Dépendances |
|--------------------------------|----------------------------|---------------------|----------------------|--------------------------------------|------------------------------|-------------|
| | Livraison | 3 jours | Développeur Frontend | Déploiement de l'interface | Interface déployée | - |
| | Total | 3-4 semaines | | | | |
| 3. Gestion Conférences | Spécification | 3 jours | Analyste | Définition des fonctionnalités CRUD | Document fonctionnel | Incrément 2 |
| | Conception | 5 jours | Architecte logiciel | Conception API et agenda | Schéma API, design agenda | - |
| | Programmation | 5 jours | Développeur Backend | Implémentation Node.js | Code backend | - |
| | Validation et Vérification | 3 jours | Testeur | Tests fonctionnels et performance | Rapport de tests | - |
| | Livraison | 2 jours | Développeur Backend | Déploiement du module | Module déployé | - |
| | Total | 2-3 semaines | | | | |
| 4. Paiements/Sponsoring | Spécification | 5 jours | Analyste | Analyse des besoins de paiement | Spécifications paiement | Incrément 3 |
| | Conception | 7 jours | Architecte logiciel | Workflow paiement et formulaires | Design workflow | - |
| | Programmation | 7 jours | Développeur Backend | Intégration Stripe, gestion sponsors | Code paiement | - |
| | Validation et Vérification | 5 jours | Testeur Sécurité | Tests de sécurité et fiabilité | Rapport de tests | - |
| | Livraison | 3 jours | Développeur Backend | Déploiement du système | Système déployé | - |
| | Total | 3-4 semaines | | | | |
| 5. Notifications | Spécification | 3 jours | Analyste | Définition des notifications | Spécifications notifications | Incrément 3 |
| | Conception | 5 jours | Architecte logiciel | Templates et système d'envoi | Design templates | - |
| | Programmation | 5 jours | Développeur Backend | Intégration SendGrid/Twilio | Code notifications | - |
| | Validation et Vérification | 3 jours | Testeur | Tests d'envoi et personnalisation | Rapport de tests | - |
| | Livraison | 2 jours | Développeur Backend | Déploiement des notifications | Système déployé | - |
| | Total | 2-3 semaines | | | | |

Durée Totale Estimée

- **Minimum** : $2 + 3 + 2 + 3 + 2 = 12$ semaines.
- **Maximum** : $3 + 4 + 3 + 4 + 3 = 17$ semaines.
- **Avec marge de 10 %** : 13-19 semaines.

Notes sur les Rôles

- **Analyste** : Définit les besoins et spécifications.
- **Architecte BD/Logiciel** : Conçoit les structures techniques.
- **Designer UX/UI** : Crée les maquettes et interfaces.
- **Développeur Backend/Frontend** : Programme les fonctionnalités.
- **Testeur** : Valide et vérifie la qualité.
- **Administrateur système** : Gère le déploiement.

5. Sécurité et Confidentialité

- Conformité RGPD : Consentement explicite, droit à l'oubli.
- Chiffrement AES-256 pour les données sensibles.
- Authentification MFA obligatoire.
- Audit de sécurité avant déploiement.

6. Accessibilité

- Respect des normes WCAG 2.1 (niveau AA).
- Options de contraste élevé et navigation clavier.

7. Maintenance et Support Post-Déploiement

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

8. Indicateurs de Performance (KPI)

- Taux d'inscription (% de places remplies).
- Satisfaction moyenne (échelle 0-5 via enquêtes).
- Temps de réponse serveur (< 500 ms).

9. Conception et Documentation via AGL

9.1 Choix des AGL

- **UML** : Architect Entreprise.

- **UX/UI** : lunacy(prototypage rapide).
- **IDE** : Eclipse.

9.2 Conception UML

9.2 Conception UML

9.2.1 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation illustre les interactions entre les acteurs et le système, en mettant en évidence les fonctionnalités principales.

Diagramme de cas d'utilisation complet - Système de Gestion de Conférences

Voici l'explication détaillée du diagramme de cas d'utilisation complet :

Acteurs

1. Utilisateur

- Représente les visiteurs standard du système
- Accède principalement aux fonctionnalités de consultation
- Constitue la base des autres types d'utilisateurs (par héritage)

2. Conférencier

- Hérite des capacités de l'Utilisateur (relation de généralisation)
- Possède des fonctionnalités supplémentaires liées à la présentation de conférences
- Représente un rôle spécialisé dans le système

3. Administrateur

- Dispose des droits de gestion sur l'ensemble du système
- Supervise les conférences, conférenciers et documents
- Contrôle le fonctionnement global de la plateforme

Cas d'utilisation principaux

Pour l'Utilisateur

- 1. Consulter les conférences**
 - Fonctionnalité de base permettant de voir toutes les conférences
 - S'étend vers "Filtrer les conférences" (relation <<extend>>)
 - Peut être étendue par "Rechercher une conférence"
- 2. Rechercher une conférence**

- Extension de "Consulter les conférences"
- Permet une recherche ciblée selon différents critères
- 3. **Consulter les documents**
 - Accès aux documents publics du système
 - S'étend vers "Télécharger les documents" (relation <<extend>>)
- 4. **Consulter les statistiques**
 - Visualisation des données statistiques sur les conférences
- 5. **Contacteur l'équipe**
 - Fonctionnalité de communication avec les responsables

Pour le Conférencier

1. **S'inscrire comme conférencier**
 - Processus d'inscription spécifique
 - S'étend vers "Soumettre des documents" et "Gérer son profil"

Pour l'Administrateur

1. **Gérer les conférences**
 - Administration complète des événements
 - Inclut obligatoirement "Gérer les conférenciers" (relation <<include>>)
 - S'étend vers "Planifier les sessions"
2. **Gérer les conférenciers**
 - Supervision des intervenants
 - S'étend vers "Valider les inscriptions"
3. **Gérer les documents**
 - Administration des ressources documentaires
 - S'étend vers "Gérer les droits d'accès"

Relations spécifiques

Relations d'extension (<<extend>>)

Ces relations indiquent des fonctionnalités optionnelles qui étendent un cas d'utilisation de base :

- "Filtrer les conférences" étend "Consulter les conférences"
- "Télécharger les documents" étend "Consulter les documents"
- "Soumettre des documents" et "Gérer son profil" étendent "S'inscrire comme conférencier"
- "Planifier les sessions" étend "Gérer les conférences"
- "Valider les inscriptions" étend "Gérer les conférenciers"
- "Gérer les droits d'accès" étend "Gérer les documents"

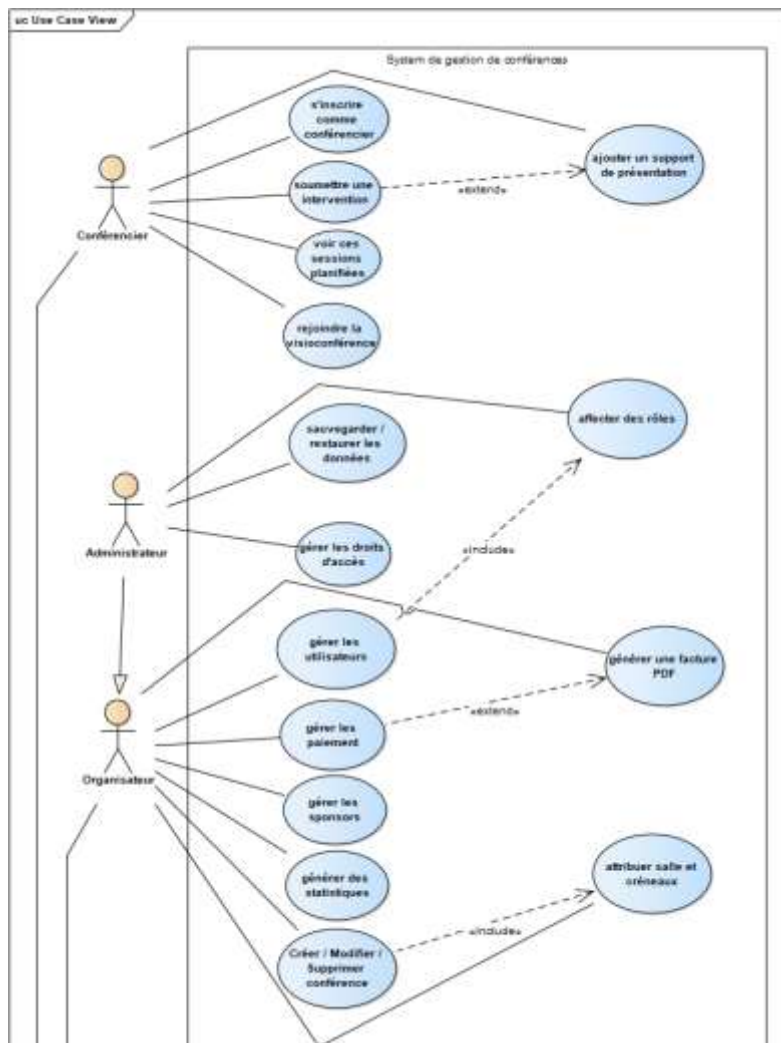
Relation d'inclusion (<<include>>)

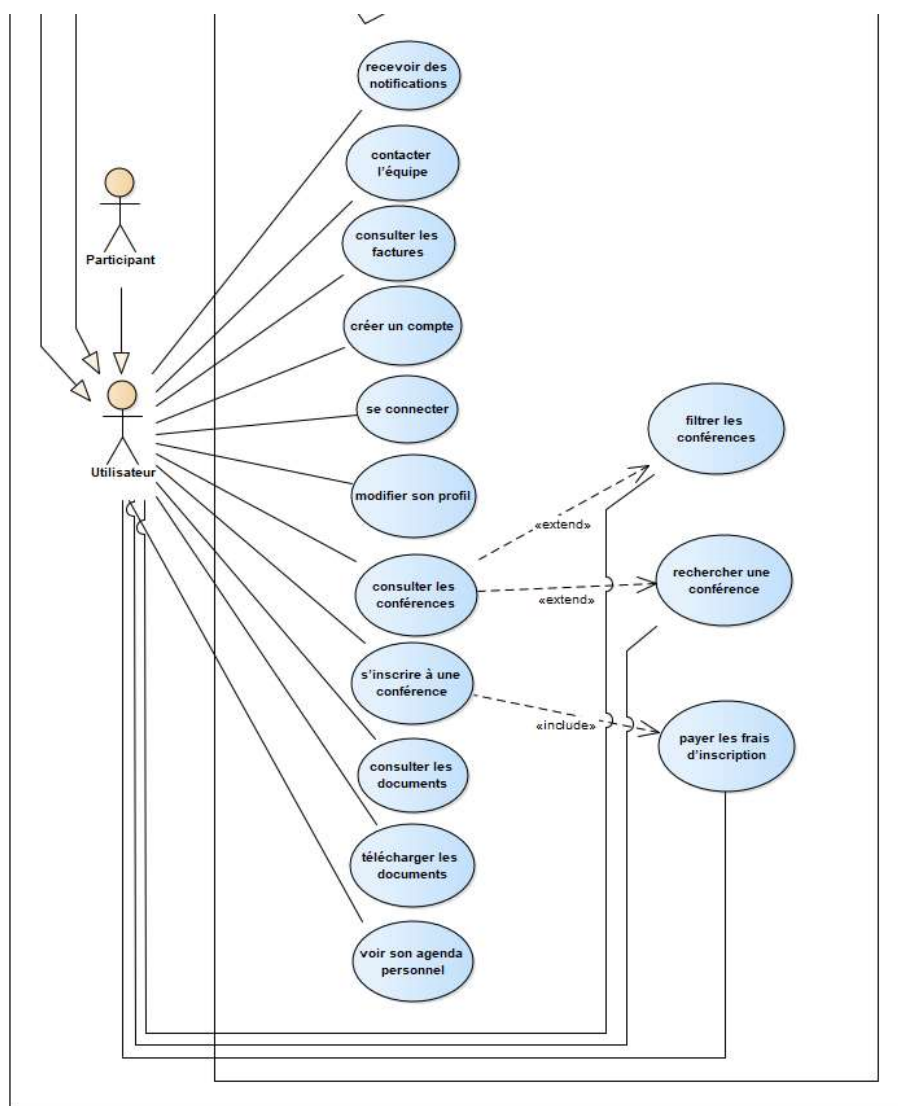
Cette relation indique une fonctionnalité obligatoirement incluse dans une autre :

- "Gérer les conférences" inclut "Gérer les conférenciers", car la gestion des événements nécessite la gestion des intervenants

Relation d'héritage (généralisation)

- Le Conférencier hérite des fonctionnalités de l'Utilisateur, comme indiqué par la flèche de généralisation et la note explicative





9.2.2 Diagramme de Classes

Le diagramme de classes décrit la structure statique du système, avec les entités principales et leurs relations.

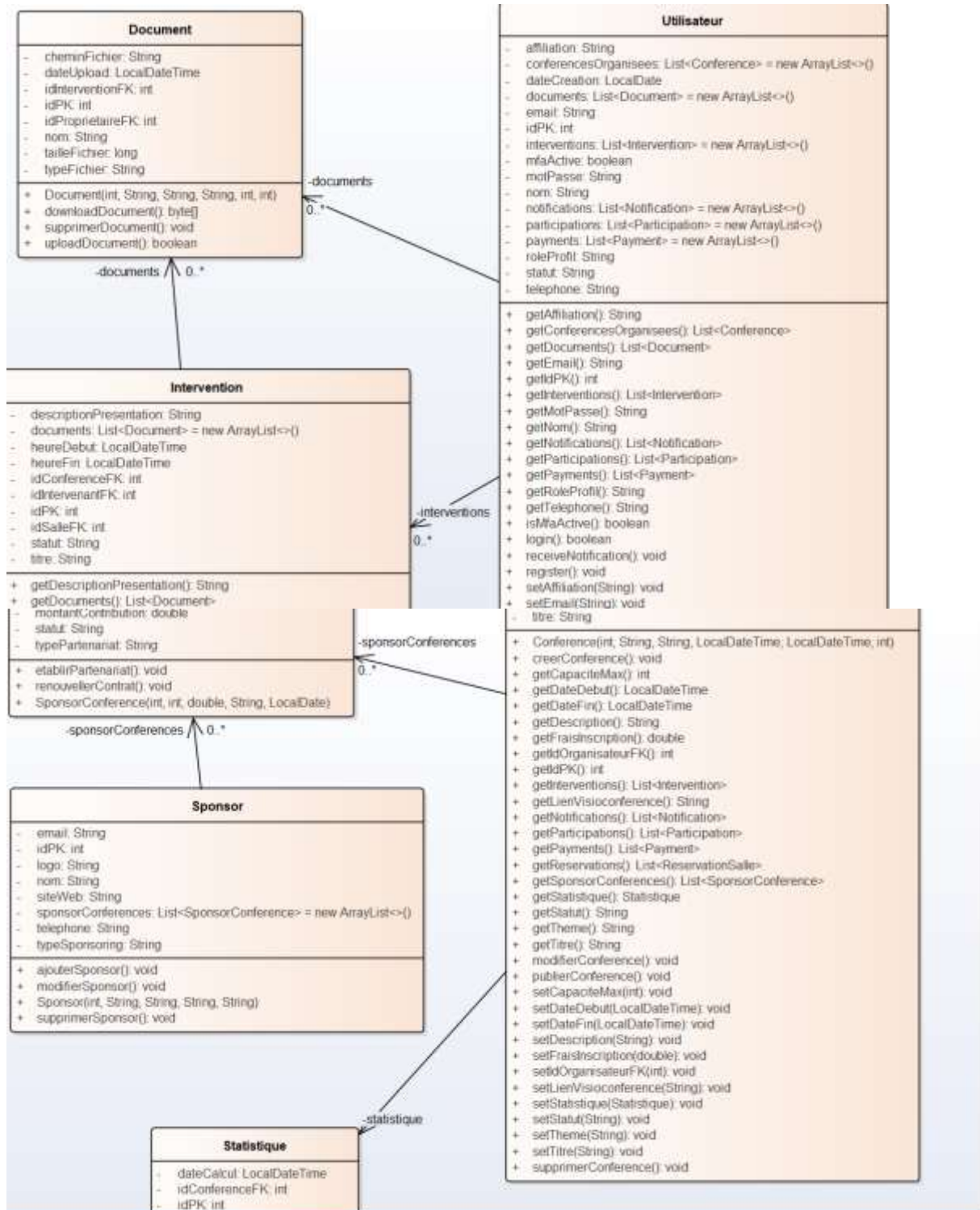
Classes Principales

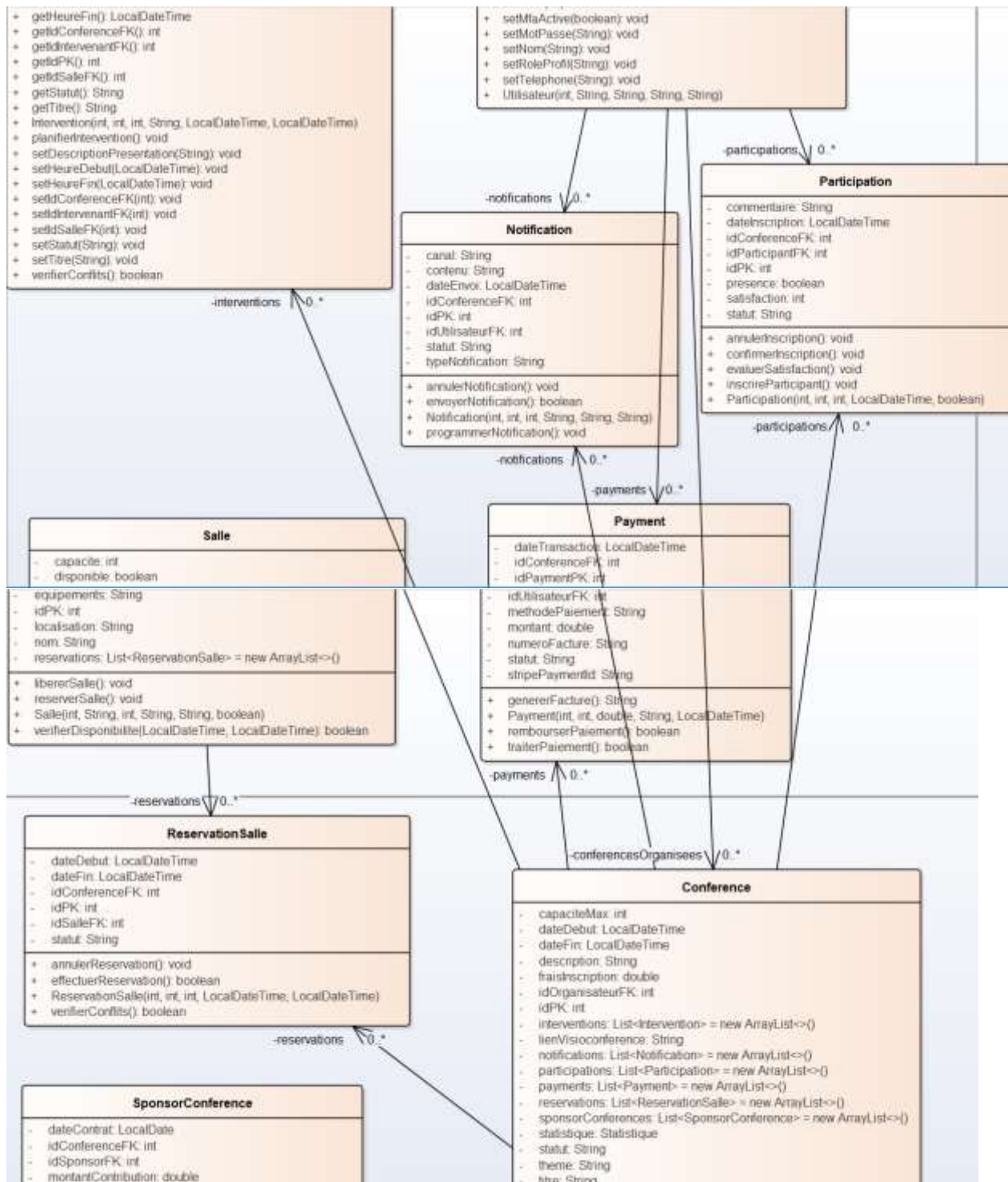
1. Utilisateur

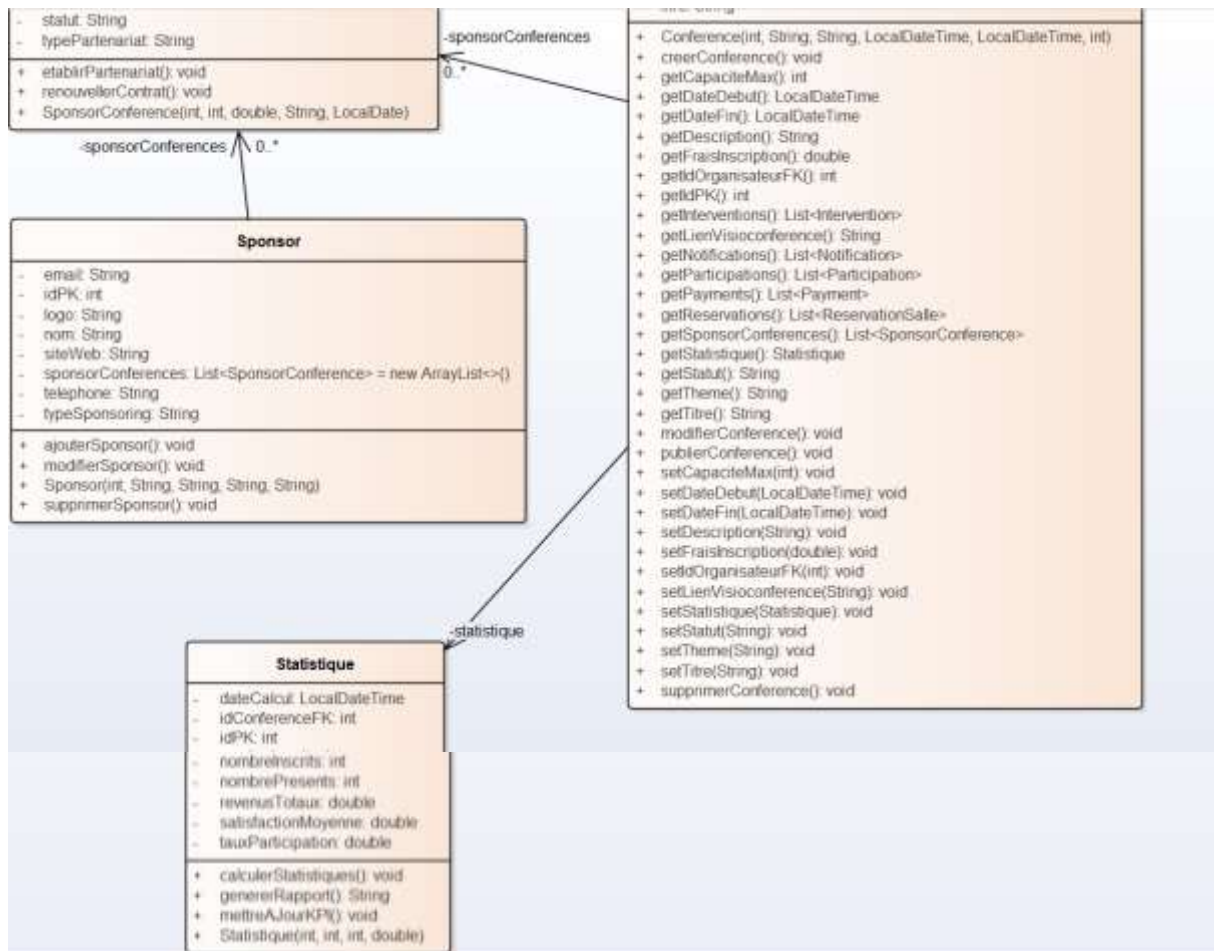
- **Attributs** : id: INT (PK), nom: VARCHAR(255), email: VARCHAR(255) (unique), role: ENUM (participant, intervenant, organisateur, administrateur), motDePasse: VARCHAR(255) (hashé).
- 2. **Conference**
 - **Attributs** : id: INT (PK), titre: VARCHAR(255), date: DATETIME, organisateurId: INT (FK), visioconferenceLien: VARCHAR(255).
- 3. **Intervention**
 - **Attributs** : id: INT (PK), conferenceId: INT (FK), intervenantId: INT (FK), horaire: DATETIME, salleId: INT (FK), supportPresentation: VARCHAR(255).
- 4. **Salle**
 - **Attributs** : id: INT (PK), nom: VARCHAR(255), capacite: INT, accessibilite: BOOLEAN.
- 5. **Sponsor**
 - **Attributs** : id: INT (PK), nom: VARCHAR(255), montant: DECIMAL, conferenceId: INT (FK).
- 6. **Paielement**
 - **Attributs** : id: INT (PK), utilisateurId: INT (FK), montant: DECIMAL, statut: ENUM (en attente, complété, annulé), dateTransaction: DATETIME.
- 7. **Notification**
 - **Attributs** : id: INT (PK), utilisateurId: INT (FK), type: ENUM (email, SMS, push), contenu: TEXT, dateEnvoi: DATETIME.
- 8. **Statistique** (ajoutée pour suivi KPI)
 - **Attributs** : id: INT (PK), conferenceId: INT (FK), nombreParticipants: INT, satisfactionMoyenne: FLOAT, revenusTotal: DECIMAL.
- **1-n (Utilisateur-Conference)** : Un organisateur peut gérer plusieurs conférences.
- **1-n (Conference-Intervention)** : Une conférence peut avoir plusieurs interventions.
- **1-n (Utilisateur-Intervention)** : Un intervenant peut participer à plusieurs interventions.
- **1-n (Conference-Sponsor)** : Une conférence peut avoir plusieurs sponsors.
- **1-n (Utilisateur-Paiement)** : Un utilisateur peut effectuer plusieurs paiements.
- **1-n (Utilisateur-Notification)** : Un utilisateur peut recevoir plusieurs notifications.
- **1-1 (Conference-Statistique)** : Une conférence a un ensemble unique de statistiques.
- **1-n (Salle-Intervention)** : Une salle peut accueillir plusieurs interventions.

Détails des Relations

- **Utilisateur** : Classe centrale, liée à presque toutes les autres via des rôles spécifiques.
- **Conference** : Entité pivot reliant organisateurs, intervenants, sponsors, et statistiques.
- **Intervention** : Relie Conference, Intervenat (Utilisateur), et Salle avec des détails spécifiques (horaire, support).







Voici le diagramme de classes UML complet et détaillé pour le système de gestion de conférences :

Structure des classes

1. Utilisateur :

- Classe centrale représentant tous les types d'utilisateurs (participants, intervenants, organisateurs, administrateurs)
- Attributs incluant identifiant, informations personnelles et rôle
- Méthodes de gestion du compte et d'authentification

2. Conference :

- Entité pivot du système, liée à l'organisateur qui la crée
- Stocke les informations essentielles sur l'événement
- Méthodes pour la gestion complète du cycle de vie d'une conférence

3. Intervention :

- Représente une présentation spécifique dans une conférence
- Liée à la conférence, à l'intervenant et à la salle
- Inclut les détails logistiques et le support de présentation

4. Salle :

- Stocke les informations sur les espaces physiques
- Inclut des critères importants comme la capacité et l'accessibilité
- Méthodes pour la gestion de la disponibilité
- 5. **Sponsor :**
 - Gère les informations relatives aux financements externes
 - Liée à la conférence financée
 - Méthodes pour la gestion des montants et factures
- 6. **Paielement :**
 - Suit les transactions financières des utilisateurs
 - États possibles du paiement via une énumération
 - Méthodes de traitement des paiements
- 7. **Notification :**
 - Gère les communications avec les utilisateurs
 - Types variés (email, SMS, push)
 - Méthodes d'envoi et de planification
- 8. **Statistique :**
 - Collecte les KPIs pour chaque conférence
 - Relation one-to-one avec Conference
 - Méthodes d'analyse et de reporting
- 9. **Participation** (ajoutée) :
 - Classe d'association entre Utilisateur et Conference
 - Gère les inscriptions aux conférences
 - Suit la présence effective

Relations principales

- **Organisation** : Un utilisateur (organisateur) peut créer plusieurs conférences
- **Intervention** : Un utilisateur (intervenant) peut participer à plusieurs sessions
- **Participation** : Un utilisateur (participant) peut s'inscrire à plusieurs conférences
- **Planification** : Une conférence contient plusieurs interventions
- **Financement** : Une conférence peut avoir plusieurs sponsors
- **Localisation** : Une intervention se déroule dans une salle spécifique
- **Mesure** : Une conférence est associée à un unique ensemble de statistiques

9.2.3 Diagramme de Séquence

Le diagramme de séquence montre les interactions dynamiques entre les objets pour un scénario spécifique. Voici trois exemples clés : **Inscription**, **Planification d'une conférence**, et **Gestion des paiements**.

1. Inscription d'un Participant

- **Étapes :**
 1. Le **Participant** remplit le formulaire (nom, email, mot de passe).
 2. Le **Système** vérifie les informations (email unique, mot de passe valide).
 3. Le **Système** enregistre le compte dans la **Base de Données**.

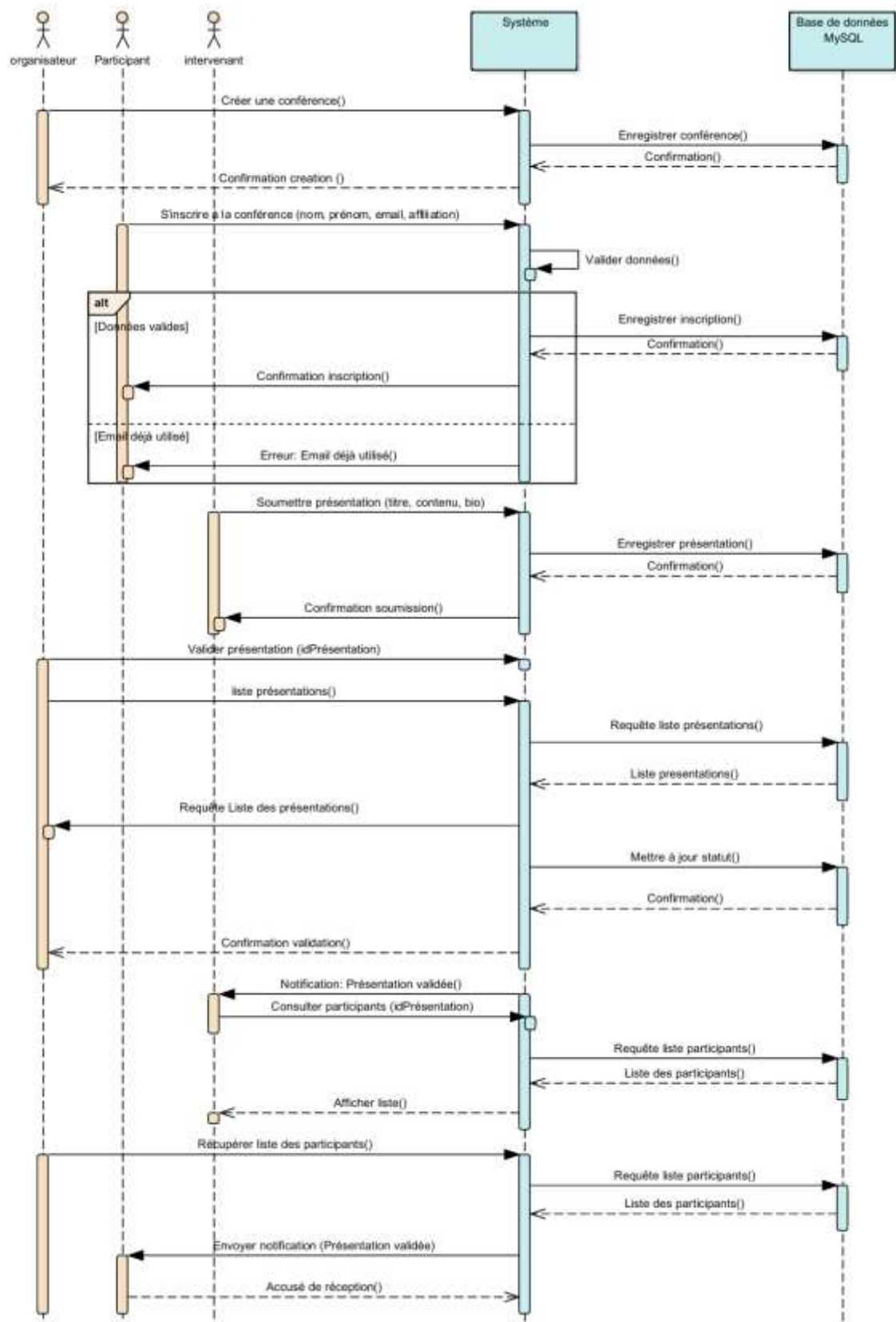
4. La **Base de Données** confirme l'enregistrement.
5. Le **Système** demande au **Service Notification** d'envoyer une confirmation (email).
6. Le **Participant** reçoit la confirmation.

2. Planification d'une Conférence

- **Étapes :**
 1. L'**Organisateur** crée une conférence (titre, date, salle).
 2. Le **Système** vérifie la disponibilité de la salle dans la **Base de Données**.
 3. La **Base de Données** confirme la disponibilité.
 4. Le **Système** demande un lien visioconférence au **Service Visioconf** (Zoom/Teams).
 5. Le **Service Visioconf** renvoie le lien.
 6. Le **Système** enregistre la conférence dans la **Base de Données**.
 7. La **Base de Données** confirme.
 8. L'**Organisateur** reçoit la confirmation.

3. Gestion des Paiements

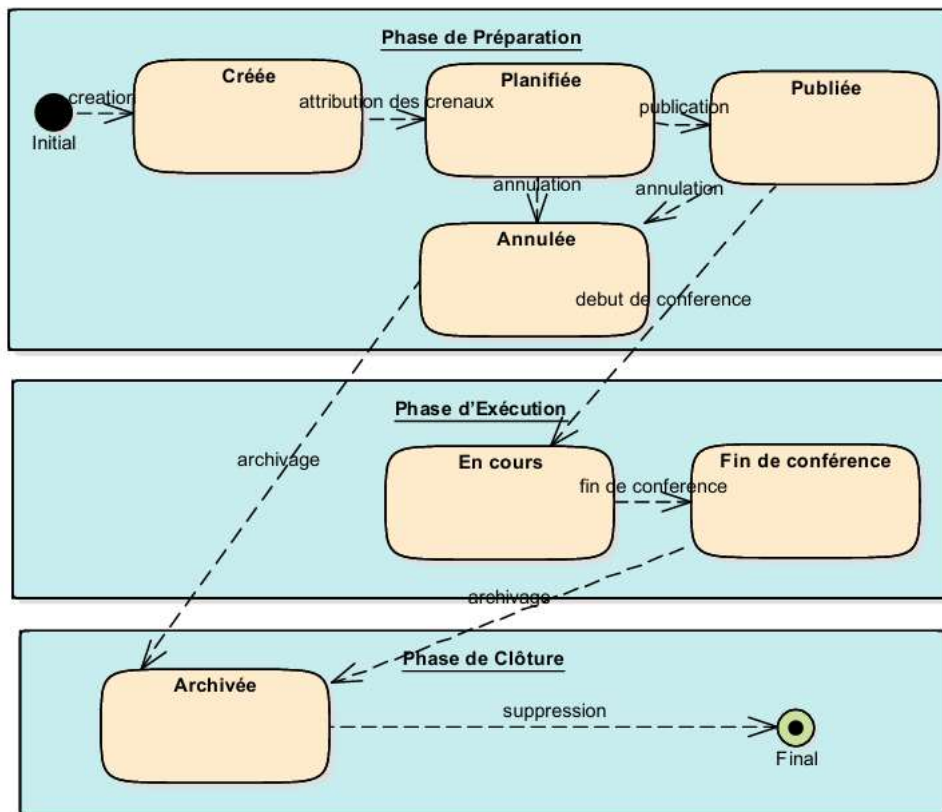
- **Étapes :**
 1. L'**Organisateur** initie un paiement (frais ou sponsoring).
 2. Le **Système** traite le paiement via le **Service Paiement** (Stripe).
 3. Le **Service Paiement** confirme la transaction.
 4. Le **Système** enregistre le paiement dans la **Base de Données**.
 5. La **Base de Données** confirme l'enregistrement.
 6. Le **Système** génère une facture (PDF) et la stocke.
 7. L'**Organisateur** reçoit la facture.



9.2.4 Diagramme d'état :

États du cycle de vie d'une Conférence organisés en 3 phases :

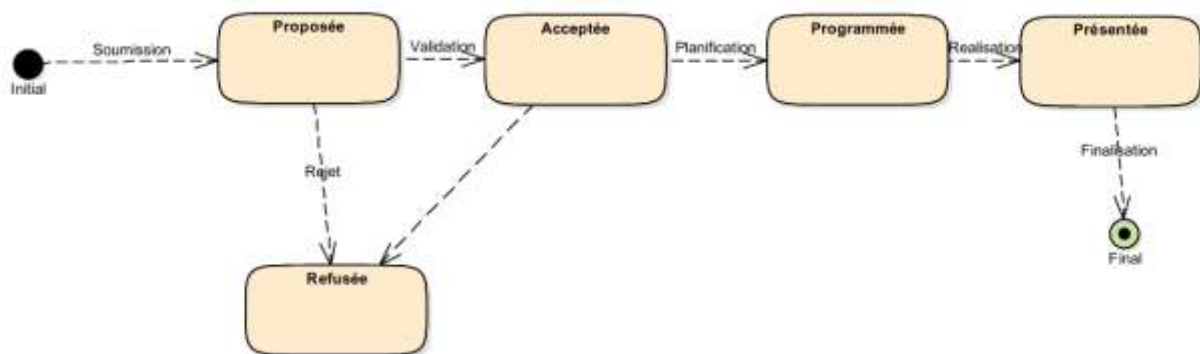
- **Phase de Préparation** : Créée → Planifiée → Publiée (avec possibilité d'annulation)
- **Phase d'Exécution** : En Cours → Terminée
- **Phase de Clôture** : Archivée → Suppression



États du cycle de vie d'une Intervention :

- Proposée → Acceptée → Programmée → Présentée
- Avec branche alternative : Proposée → Refusée

Le diagramme inclut toutes les transitions avec leurs étiquettes (Création, Attribution des créneaux, Publication, etc.) et les notes explicatives du diagramme original. Les états terminaux sont correctement représentés avec les cercles pleins.



9.3 Conception UX/UI

- **Pages clés** : Accueil, Dashboard organisateur, Agenda interactif.
- **Prototypes** : Réalisés via Figma, validés par tests utilisateurs.

Pages et Interfaces à Construire

Voici les pages et interfaces essentielles, organisées par type d'utilisateur et fonction principale, avec des détails sur leur contenu et leur conception.

1. Page d'Accueil (publique)

- **Objectif** : Présenter le système et donner un accès rapide aux fonctionnalités principales.
- **Fonctionnalités** :
 - Aperçu des conférences à venir (liste ou carrousel avec titre, date, lieu).
 - Boutons d'action : “S’inscrire” (nouveau compte), “Se connecter”, “Explorer les conférences”.
 - Section “À propos” (résumé du logiciel, avantages).
 - Filtres rapides (par date, thème).
- **Conception** :
 - **Wireframe** : Header avec logo et navigation (Accueil, Connexion, Inscription), section héroïque avec un visuel attractif (image de conférence), liste de conférences en grille.
 - **UI** : Palette de couleurs sobres (bleu pour confiance, blanc pour clarté), typographie lisible (ex. : Roboto), bouton CTA (Call-to-Action) en vert ou orange.
 - **Responsive** : Grille ajustable (1 colonne sur mobile, 3 sur desktop).
- **Public cible** : Tous (participants, intervenants potentiels, visiteurs).

2. Page d'Inscription / Connexion (publique)

- **Objectif** : Permettre la création de compte et l'authentification sécurisée.
- **Fonctionnalités** :
 - **Inscription** : Formulaire avec champs (nom, email, mot de passe, rôle), case RGPD, bouton "Créer un compte".
 - **Connexion** : Champs (email, mot de passe), option "Mot de passe oublié", MFA (code SMS/email).
- **Conception** :
 - **Wireframe** : Mise en page centrée, formulaire compact, lien toggle entre Inscription/Connexion.
 - **UI** : Design minimaliste, champ avec validation en temps réel (icône verte si valide), bouton principal contrasté.
 - **Responsive** : Formulaire empilé sur mobile, aligné horizontalement sur desktop.
- **Public cible** : Participants, Intervenants, Organisateurs, Administrateurs.

3. Dashboard Participant

- **Objectif** : Offrir une vue personnalisée des activités du participant.
- **Fonctionnalités** :
 - Liste des conférences inscrites (titre, date, statut).
 - Agenda personnel (calendrier des sessions).
 - Accès aux supports de présentation (liens téléchargeables).
 - Notifications reçues (rappels, mises à jour).
 - Bouton "S'inscrire à une nouvelle conférence".
- **Conception** :
 - **Wireframe** : Sidebar gauche (menu : Mes Conférences, Agenda, Notifications), contenu principal avec onglets ou sections.
 - **UI** : Calendrier interactif (ex. : FullCalendar), notifications en pop-up ou liste déroulante, icônes claires (téléchargement, inscription).
 - **Responsive** : Sidebar masquée sur mobile (menu hamburger), contenu empilé.
- **Public cible** : Participants.

4. Dashboard Intervenant

- **Objectif** : Gérer les interventions et les supports de présentation.
- **Fonctionnalités** :
 - Liste des interventions planifiées (titre, date, conférence).
 - Formulaire de soumission (titre, résumé, fichier PDF/PPT).
 - Gestion du profil (bio, photo, liens réseaux sociaux).
 - Accès aux visioconférences (lien Zoom/Teams).
- **Conception** :
 - **Wireframe** : Sidebar (Mes Interventions, Profil), section principale avec tableau et formulaire.
 - **UI** : Bouton "Soumettre" en évidence, aperçu des fichiers uploadés, design épuré avec focus sur le contenu.
 - **Responsive** : Formulaire compact sur mobile, tableau scrollable.
- **Public cible** : Intervenants.

5. Dashboard Organisateur

- **Objectif** : Centraliser la gestion des conférences, paiements et notifications.
- **Fonctionnalités** :
 - **Gestion des Conférences** : Création/édition (titre, date, salle, visioconférence), attribution des intervenants.
 - **Agenda Global** : Vue calendrier de toutes les sessions.
 - **Paiements** : Suivi des frais et sponsoring, génération de factures (PDF).
 - **Notifications** : Envoi manuel/automatique (email, SMS) avec templates personnalisables.
 - **Statistiques** : Graphiques (participation, revenus, satisfaction).
- **Conception** :
 - **Wireframe** : Sidebar (Conférences, Agenda, Finances, Notifications, Stats), contenu avec onglets ou sections déroulantes.
 - **UI** : Graphiques interactifs (Chart.js), formulaires structurés, bouton “Envoyer” contrasté pour notifications.
 - **Responsive** : Onglets empilés sur mobile, graphiques redimensionnés.
- **Public cible** : Organisateurs.

6. Dashboard Administrateur

- **Objectif** : Superviser le système et gérer les utilisateurs.
- **Fonctionnalités** :
 - Gestion des utilisateurs (création, modification, suppression, rôles).
 - Configuration système (paramètres généraux, intégrations Zoom/Teams).
 - Suivi des performances (temps de réponse, logs).
 - Sauvegarde/Restauration des données.
- **Conception** :
 - **Wireframe** : Sidebar (Utilisateurs, Config, Performances), tableau pour les utilisateurs, formulaires pour la config.
 - **UI** : Design fonctionnel (gris pour sérieux), alertes visuelles pour performances critiques, boutons d’action clairs.
 - **Responsive** : Tableau scrollable sur mobile, formulaires simplifiés.
- **Public cible** : Administrateurs.

7. Page de Consultation des Conférences (publique/authentifiée)

- **Objectif** : Permettre l’exploration des conférences disponibles.
- **Fonctionnalités** :
 - Liste filtrable (date, thème, intervenant).
 - Détails de chaque conférence (description, horaire, lieu, lien inscription).
 - Bouton “S’inscrire” (redirige vers formulaire ou connexion).
- **Conception** :
 - **Wireframe** : Barre de filtres en haut, grille de cartes pour les conférences.
 - **UI** : Cartes avec image/thumbnail, filtres déroulants, bouton CTA en couleur vive.
 - **Responsive** : Filtres masqués sous un menu sur mobile, grille en colonne unique.
- **Public cible** : Tous (public non authentifié + utilisateurs connectés).

8. Agenda Interactif (partagé)

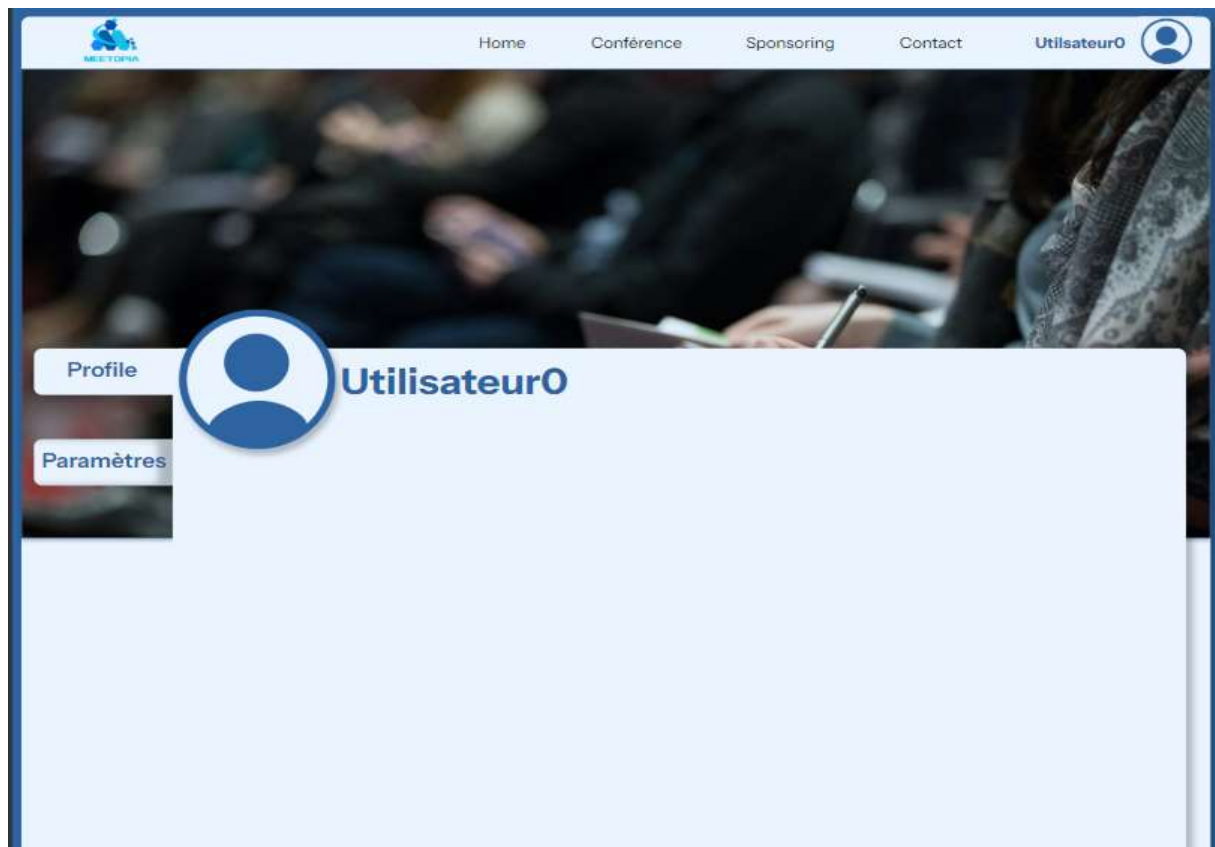
- **Objectif** : Afficher un calendrier global ou personnel des événements.
- **Fonctionnalités** :
 - Vue mensuelle/hebdomadaire/journalière.
 - Filtres (conférence, intervenant, salle).
 - Détails au clic (horaire, lieu, lien visioconférence).
- **Conception** :
 - **Wireframe** : Calendrier pleine page, barre d'outils (filtres, vues).
 - **UI** : Couleurs distinctes par type d'événement, pop-up pour détails, navigation fluide.
 - **Responsive** : Vue simplifiée sur mobile (liste ou mini-calendrier).
- **Public cible** : Participants, Intervenants, Organisateurs.

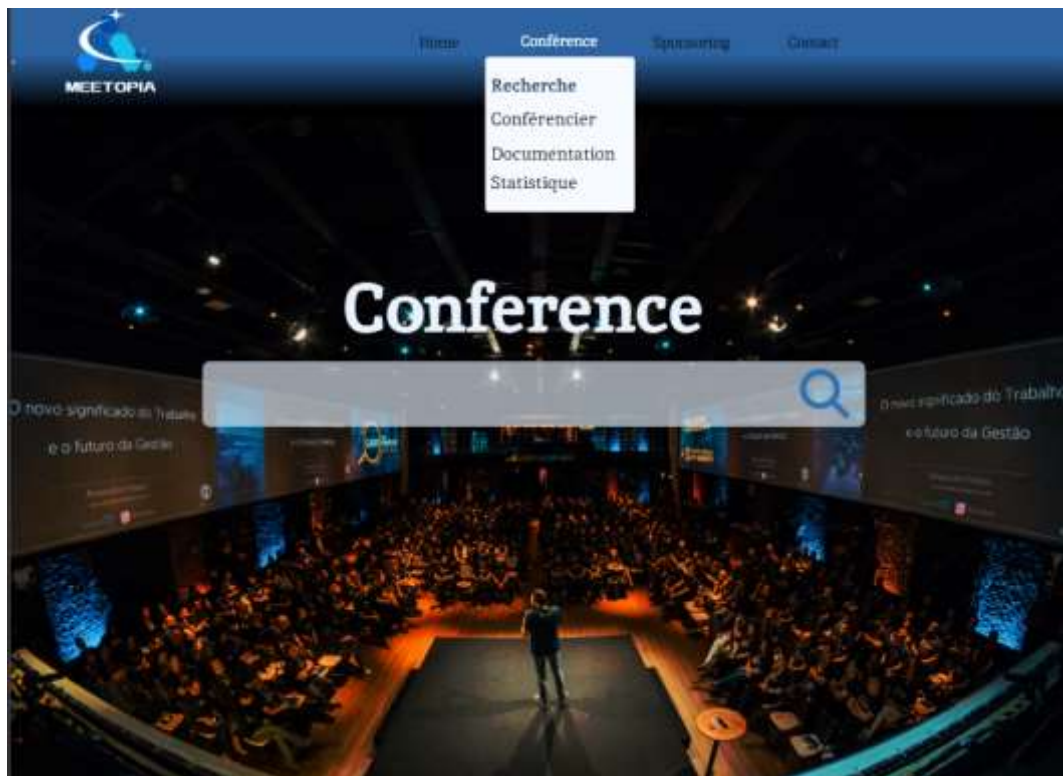
9. Interface de Paiement

- **Objectif** : Faciliter les transactions sécurisées.
- **Fonctionnalités** :
 - Formulaire de paiement (montant, carte bancaire via Stripe).
 - Confirmation de paiement (récapitulatif, facture PDF).
 - Statut des paiements passés (en attente, complété).
- **Conception** :
 - **Wireframe** : Formulaire centré, section récapitulative.
 - **UI** : Design sécurisant (icône cadenas, HTTPS visible), bouton "Payer" en vert.
 - **Responsive** : Formulaire empilé sur mobile.
- **Public cible** : Participants, Organisateurs (pour suivi).

10. Interface de Notifications

- **Objectif** : Gérer et consulter les notifications.
- **Fonctionnalités** :
 - Liste des notifications reçues (Participant/Intervenant).
 - Formulaire d'envoi (Organisateur) : destinataires, contenu, type (email/SMS).
 - Prévisualisation des templates.
- **Conception** :
 - **Wireframe** : Liste déroulante ou pop-up pour réception, formulaire avec aperçu pour envoi.
 - **UI** : Icônes par type (enveloppe pour email, téléphone pour SMS), bouton "Envoyer" clair.
 - **Responsive** : Liste compacte sur mobile, formulaire ajusté.
- **Public cible** : Tous.





[Home](#)
[Conférencier](#)
[Sponsoring](#)
[Contact](#)

Conference

Mastering Public Speaking

Inscrivez-vous dès maintenant pour réserver votre place !

Participez à une session inspirante pour perfectionner vos compétences en prise de parole ! Découvrez des techniques essentielles pour captiver votre audience et livrer des présentations percutantes.

MASTERING PUBLIC SPEAKING

Inscrivez-vous dès maintenant pour réserver votre place !

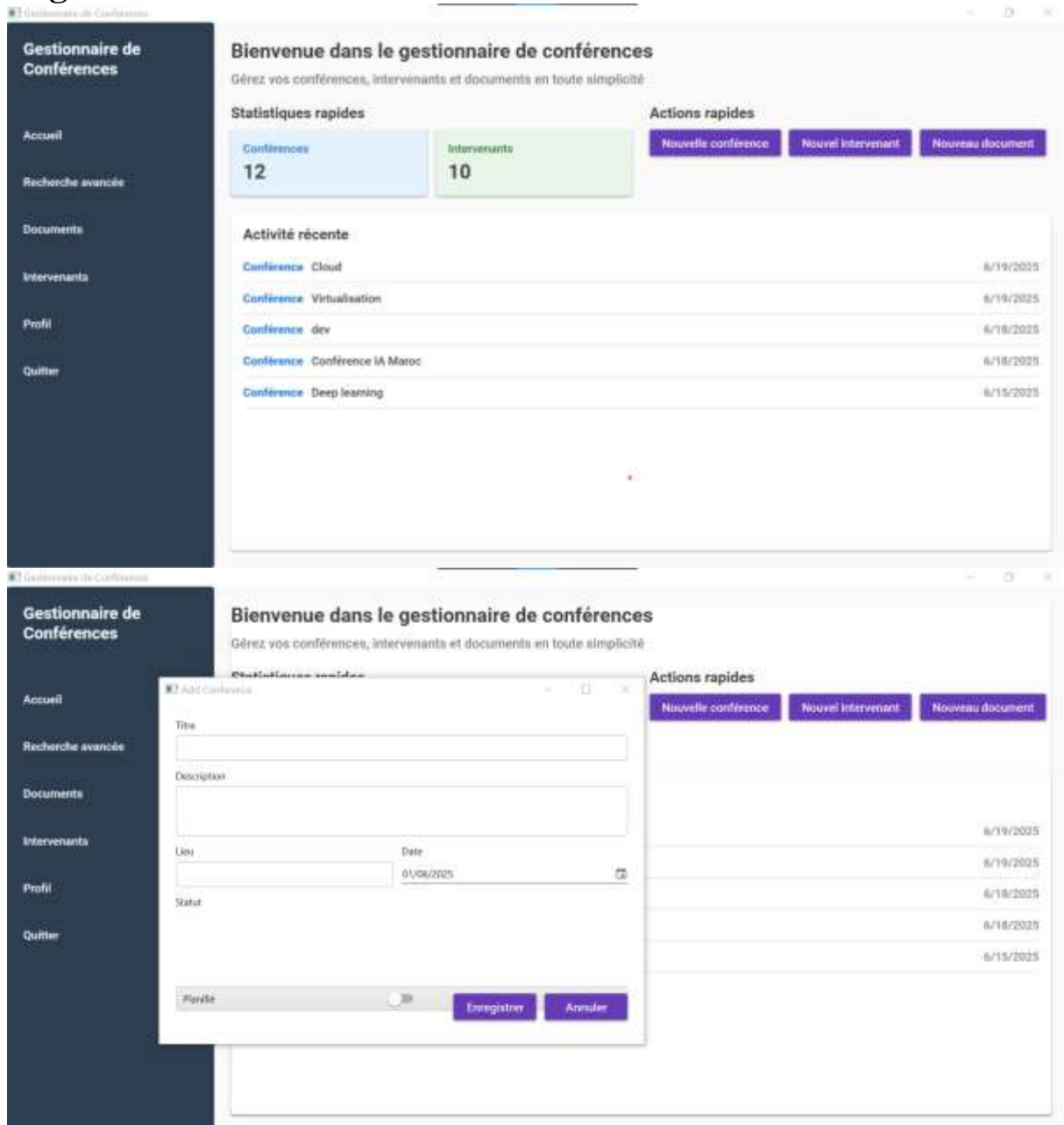
Participez à une session inspirante pour perfectionner vos compétences en prise de parole ! Découvrez des techniques essentielles pour captiver votre audience et livrer des présentations percutantes.

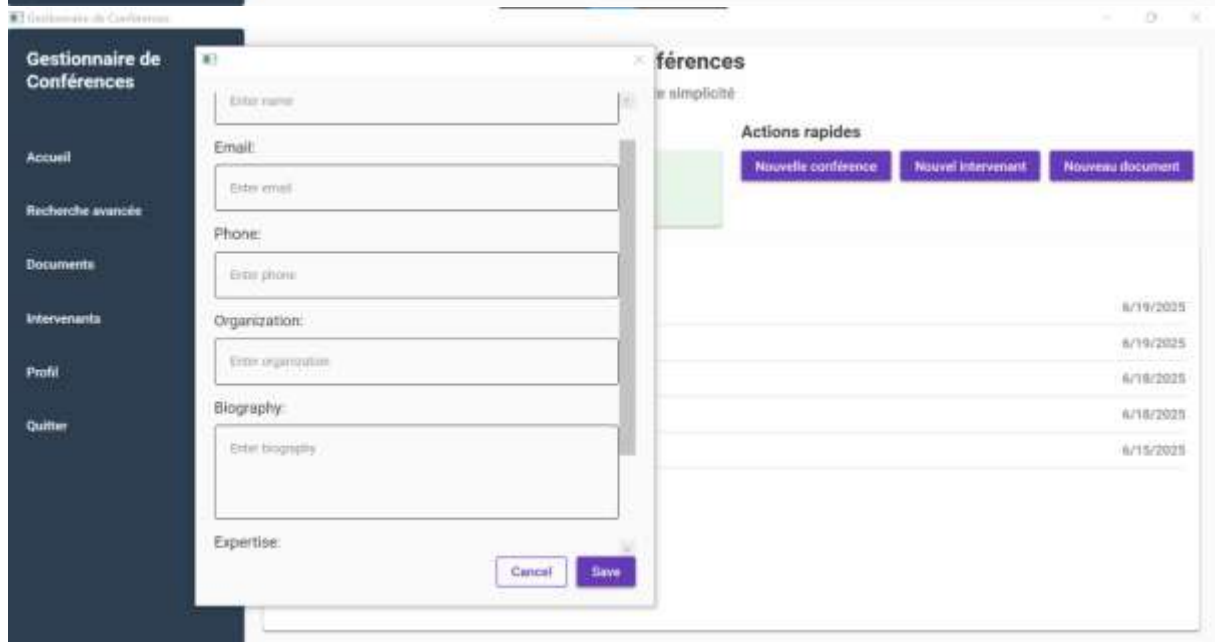
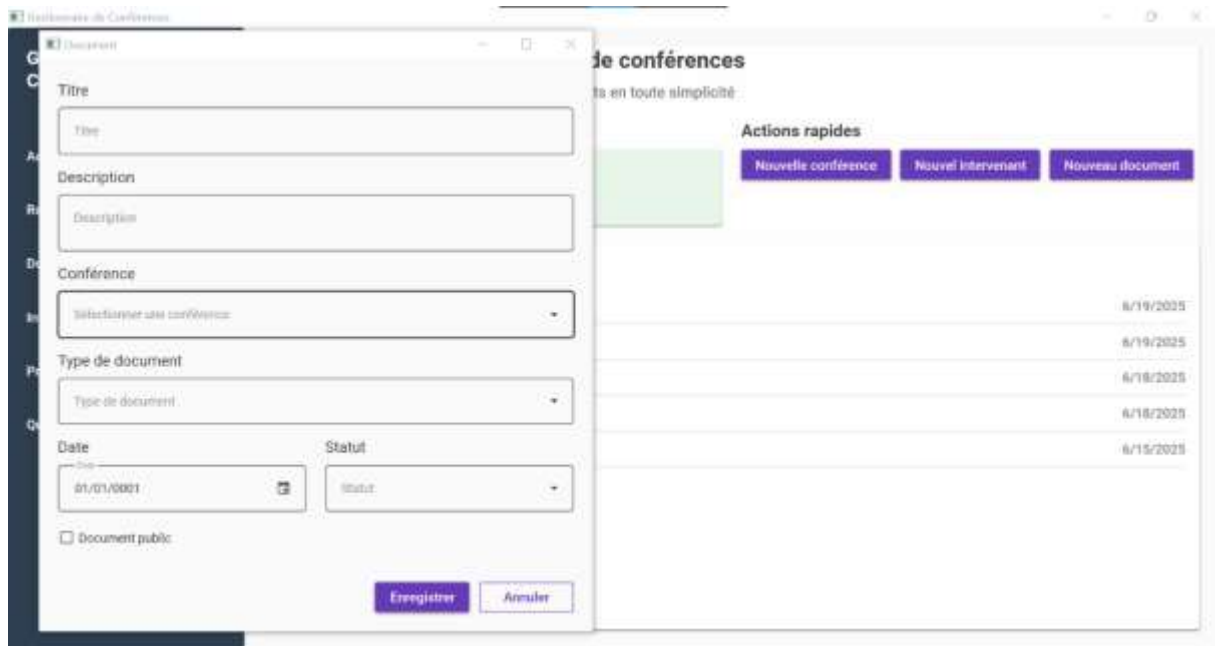
conférencier



LES PAGES REELLES DE APPLICATION :

Page Accueil :





Page Gestion d'intervants :

Gestionnaire de Conférences

Accueil

Recherche avancée

Documents

Intervenants

Profil

Quitter

Ajouter Supprimer

| Id | Name | Email | Phone | Organization | Status | Biography | Expertise | ProfilePic |
|----|------|---------------------------|------------|--------------------------|--------|----------------------|---------------------------|------------|
| 9 | | test@gmail.com | 06774312 | | | | T | |
| 12 | | test@gmail.com | 0634567 | | | | X | |
| 13 | | uyth@gmail.com | 062345899 | | | | X | |
| 14 | | uyth@gmail.com | 062345899 | | | | X | |
| 15 | | yuk@gmail.com | 06543213 | | | | HJK | |
| 16 | | yuk@gmail.com | 06543213 | | | | HJK | |
| 17 | | khewla.hacma@gmail.com | 0600000001 | Université de Rabat | Actif | Chercheuse en IA | Intelligence Artificielle | |
| 18 | | mayeouna.oukrim@gmail.com | 0600000002 | Université de Casablanca | Actif | Spécialiste Big Data | Big Data | |
| 19 | | fatima.kasri@gmail.com | 0600000003 | Université Hassan II | Actif | Experte Cloud | Cloud Computing | |
| 20 | | marwa.amrani@gmail.com | 0600000004 | INPT | Actif | Ingénieure réseaux | Réseaux | |

Page Recherche Avancée

Gestionnaire de Conférences

Accueil

Recherche avancée

Documents

Intervenants

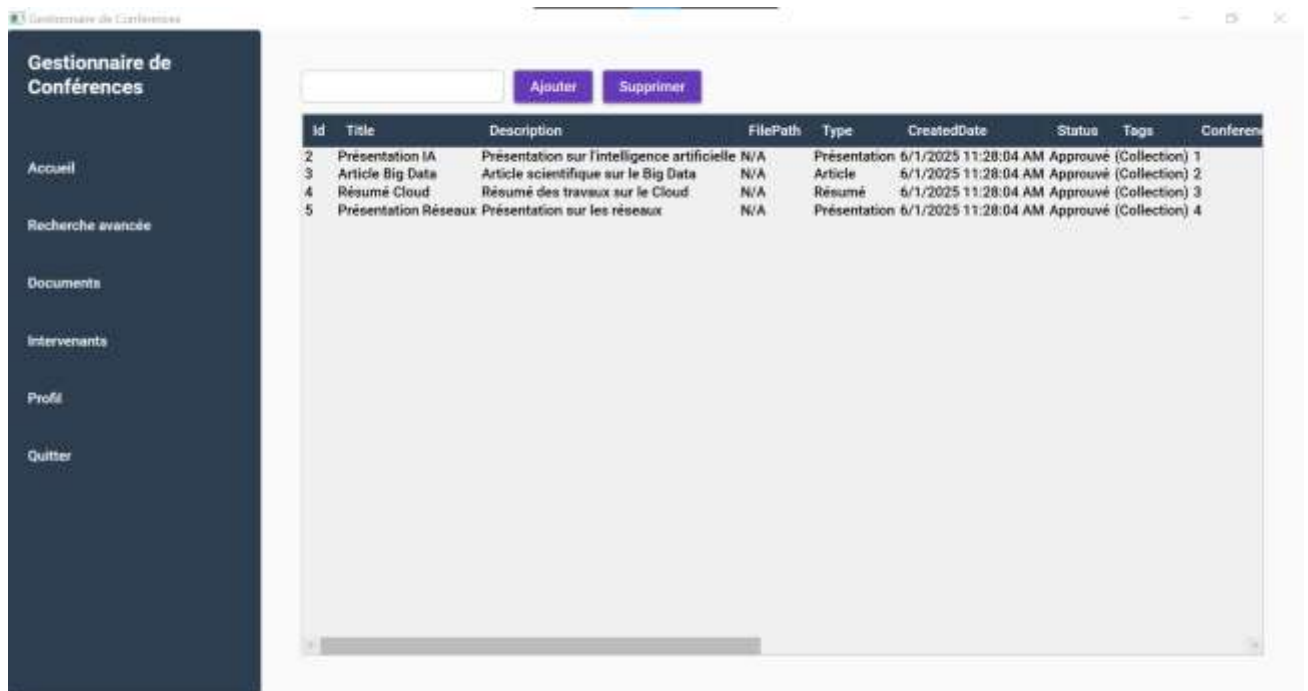
Profil

Quitter

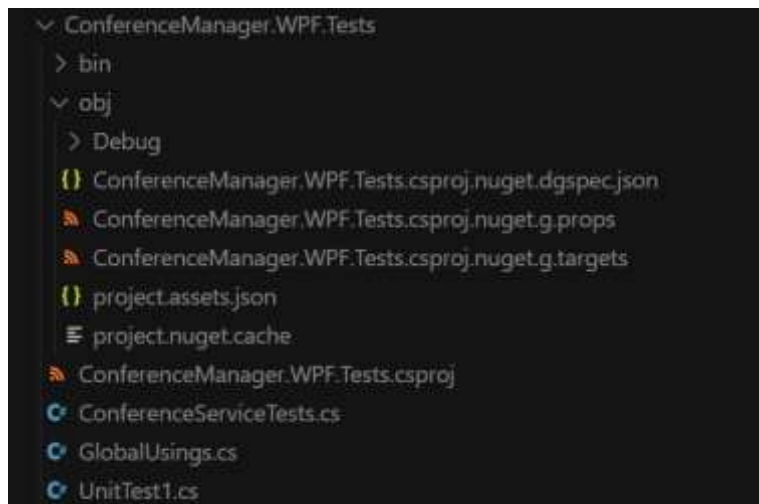
Cloud Rechercher

| Id | Title | Description | Date | StartDate | EndDate | Location | Status |
|----|--------------|-----------------------------------|-----------------------|-----------------------|-----------------------|-----------|----------|
| 6 | Cloud | cloud | 6/19/2025 12:00:00 AM | 1/1/0001 12:00:00 AM | 1/1/0001 12:00:00 AM | Rabat | Planifié |
| 9 | Cloud Expo | Exposition sur le Cloud Computing | 6/14/2025 12:00:00 AM | 6/14/2025 12:00:00 AM | 6/15/2025 12:00:00 AM | Marrakech | Ouvert |
| 19 | | | | | | | Actif |
| 4 | Résumé Cloud | Résumé des travaux sur le Cloud | 6/1/2025 11:28:04 AM | | | | Approuvé |

Page de Gestion Documents



9.4 Conception des Tests Unitaires



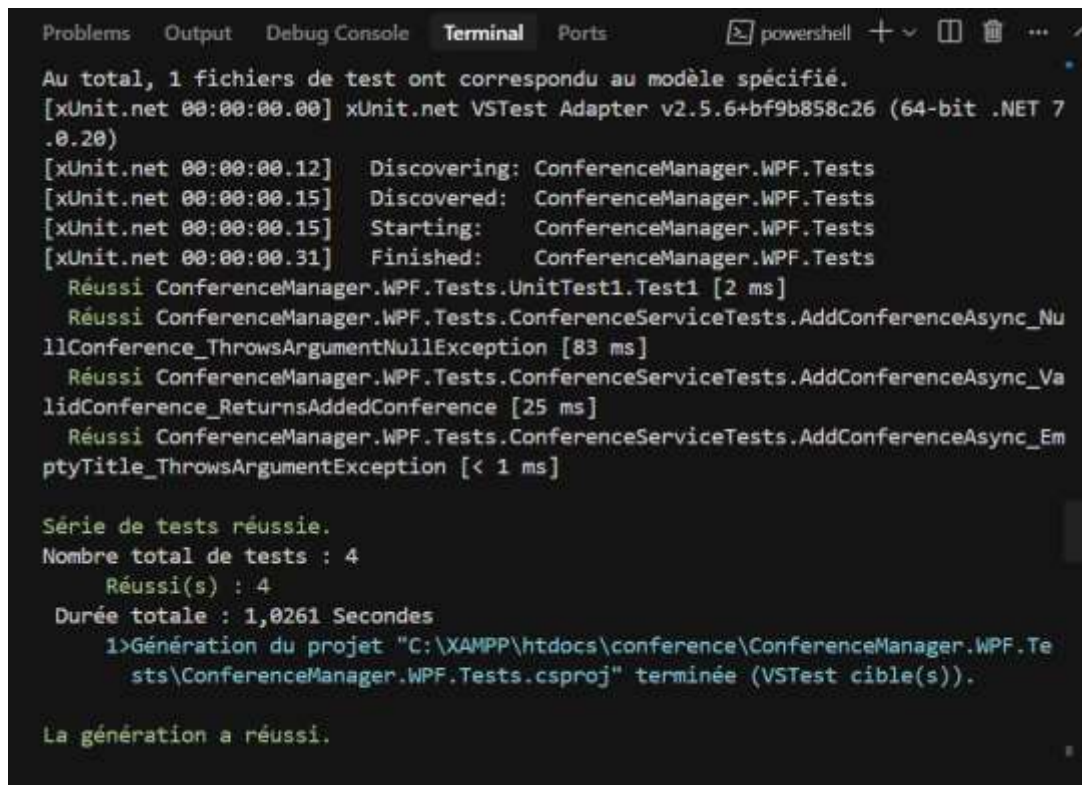
RESULTAT DES TESTS:

4 TESTS ONT ETE EXECUTES ET TOUS ONT REUSSI :

- TEST1 (2 MS)

- ADDCONFERENCEASYNC_NULLCONFERENCE_THROWSARGUMENTNULLEXCEPTION (83 MS)
- ADDCONFERENCEASYNC_VALIDCONFERENCE_RETURNSADDEDCONFERENCE (25 MS)
- ADDCONFERENCEASYNC_EMP

En résumé, tous les tests que nous avons créés pour la méthode AddConferenceAsync fonctionnent correctement, ce qui confirme que la logique de gestion des conférences est robuste.



```

Problems  Output  Debug Console  Terminal  Ports
Au total, 1 fichiers de test ont correspondu au modèle spécifié.
[xUnit.net 00:00:00.00] xUnit.net VSTest Adapter v2.5.6+bf9b858c26 (64-bit .NET 7
.0.20)
[xUnit.net 00:00:00.12]   Discovering: ConferenceManager.WPF.Tests
[xUnit.net 00:00:00.15]   Discovered:   ConferenceManager.WPF.Tests
[xUnit.net 00:00:00.15]   Starting:     ConferenceManager.WPF.Tests
[xUnit.net 00:00:00.31]   Finished:     ConferenceManager.WPF.Tests
    Réussi ConferenceManager.WPF.Tests.UnitTest1.Test1 [2 ms]
    Réussi ConferenceManager.WPF.Tests.ConferenceServiceTests.AddConferenceAsync_Nu
llConference_ThrowsArgumentNullException [83 ms]
    Réussi ConferenceManager.WPF.Tests.ConferenceServiceTests.AddConferenceAsync_Va
lidConference_ReturnsAddedConference [25 ms]
    Réussi ConferenceManager.WPF.Tests.ConferenceServiceTests.AddConferenceAsync_Em
ptyTitle_ThrowsArgumentException [< 1 ms]

Série de tests réussie.
Nombre total de tests : 4
    Réussi(s) : 4
Durée totale : 1,0261 Secondes
1>Génération du projet "C:\XAMPP\htdocs\conference\ConferenceManager.WPF.Te
sts\ConferenceManager.WPF.Tests.csproj" terminée (VSTest cible(s)).

La génération a réussi.
  
```

```
Problems Output Debug Console Terminal Ports powershell + - [ ] [ ] ... ^
● PS C:\XAMPP\htdocs\conference> dotnet test ConferenceManager.WPF.Tests/Conference
Manager.WPF.Tests.csproj --verbosity normal
La génération a démarré 01/06/2025 17:14:42.
1>Projet "C:\XAMPP\htdocs\conference\ConferenceManager.WPF.Tests\Conference
Manager.WPF.Tests.csproj" sur le nœud 1 (Restore cible(s)).
1>_GetAllRestoreProjectPathItems:
    Determining projects to restore...
Restore:
    La validation de la chaîne de certificats X.509 utilisera le magasin de
    confiance par défaut sélectionné par .NET pour le code de signature.
    La validation de la chaîne de certificats X.509 utilisera le magasin de
    confiance par défaut sélectionné par .NET pour le timestamp.
    Le fichier d'actifs n'a pas changé. Écriture du fichier d'actifs ignoré
    e. Chemin : C:\XAMPP\htdocs\conference\ConferenceManager.WPF.Tests\obj\
    project.assets.json
    Le fichier d'actifs n'a pas changé. Écriture du fichier d'actifs ignoré
    e. Chemin : C:\XAMPP\htdocs\conference\ConferenceManager.WPF\obj\projec
    t.assets.json
    Restauration effectuée de C:\XAMPP\htdocs\conference\ConferenceManager.
    WPF.Tests\ConferenceManager.WPF.Tests.csproj (en 72 ms).
    Restauration effectuée de C:\XAMPP\htdocs\conference\ConferenceManager.
    WPF\ConferenceManager.WPF.csproj (en 72 ms).

Fichiers de configuration NuGet utilisés :
```

9.5 Modèle Physique de la Base de Données

- Tables et contraintes définies dans le dictionnaire (Section 2.4)
- Index sur email et conferenceId pour optimiser les requêtes.

```
-- Table des Conférences
CREATE TABLE Conferences (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Title NVARCHAR(MAX) NOT NULL,
    Description NVARCHAR(MAX),
    StartDate DATETIME2 NOT NULL,
    EndDate DATETIME2 NOT NULL,
    Location NVARCHAR(MAX),
    MaxAttendees INT,
    IsActive BIT NOT NULL DEFAULT 1,
    CreatedAt DATETIME2 NOT NULL DEFAULT GETDATE(),
    UpdatedAt DATETIME2 NOT NULL DEFAULT GETDATE()
);
```



```
CREATE TABLE Attendees (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    FirstName NVARCHAR(MAX) NOT NULL,  
    LastName NVARCHAR(MAX) NOT NULL,  
    Email NVARCHAR(MAX) NOT NULL,  
    PhoneNumber NVARCHAR(MAX),  
    Organization NVARCHAR(MAX),  
    CreatedAt DATETIME2 NOT NULL DEFAULT GETDATE(),  
    UpdatedAt DATETIME2 NOT NULL DEFAULT GETDATE()  
);
```

-- Table des Intervenants

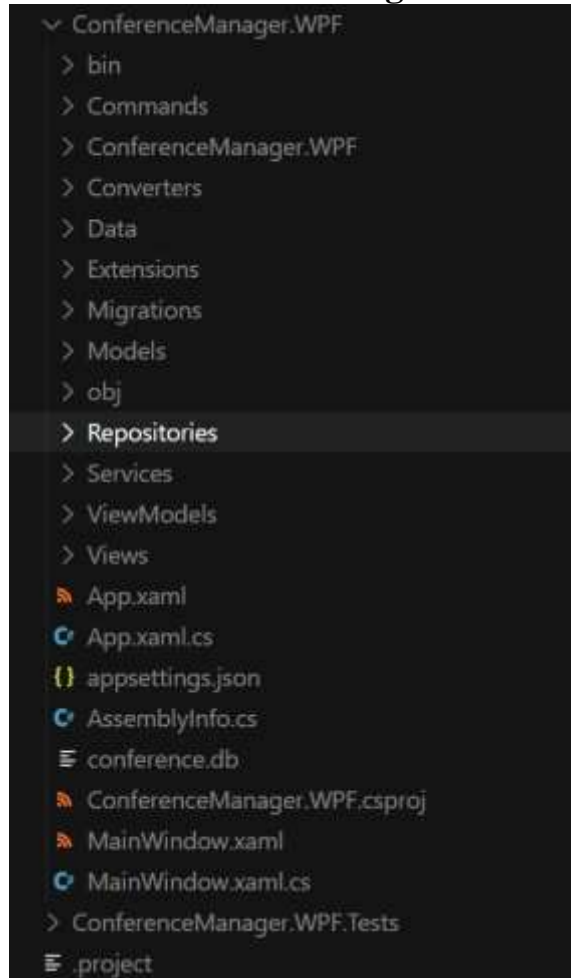
```
CREATE TABLE Speakers (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    FirstName NVARCHAR(MAX) NOT NULL,  
    LastName NVARCHAR(MAX) NOT NULL,  
    Email NVARCHAR(MAX) NOT NULL,  
    Bio NVARCHAR(MAX),  
    CreatedAt DATETIME2 NOT NULL DEFAULT GETDATE(),  
    UpdatedAt DATETIME2 NOT NULL DEFAULT GETDATE()  
);
```

```
CREATE TABLE ConferenceAttendees (  
    ConferenceId INT NOT NULL,  
    AttendeeId INT NOT NULL,  
    RegistrationDate DATETIME2 NOT NULL DEFAULT GETDATE(),  
    Status NVARCHAR(MAX) NOT NULL,  
    PRIMARY KEY (ConferenceId, AttendeeId),  
    FOREIGN KEY (ConferenceId) REFERENCES Conferences(Id),  
    FOREIGN KEY (AttendeeId) REFERENCES Attendees(Id)  
);
```

-- Table de jonction Conférences-Intervenants

```
CREATE TABLE ConferenceSpeakers (  
    ConferenceId INT NOT NULL,  
    SpeakerId INT NOT NULL,  
    Topic NVARCHAR(MAX) NOT NULL,  
    ScheduledTime DATETIME2 NOT NULL,  
    Duration INT NOT NULL,  
    PRIMARY KEY (ConferenceId, SpeakerId),  
    FOREIGN KEY (ConferenceId) REFERENCES Conferences(Id),  
    FOREIGN KEY (SpeakerId) REFERENCES Speakers(Id)  
);
```


10.Développement d'une Sous-Partie du Logiciel



Analyse des Risques et Mitigation

- **Défaillance visioconférence** : Tester les API Zoom/Teams, prévoir un lien manuel.
- **Retard développement** : Marge de 10 % dans le planning.
- **Fuite de données** : Chiffrement AES-256, audit RGPD.

Glossaire

- **AGL** : Atelier de Génie Logiciel.
- **RGPD** : Règlement Général sur la Protection des Données.
- **KPI** : Indicateur Clé de Performance.
- **MFA** : Authentification Multi-Facteurs.