

# **Cahier des Charges**

Développement d'une Application d'Aide Comptable  
en Java

**Équipe de Développement : Année 2023-2024**

**May 21, 2024**

Citation : ... le sel ...

Son absence rend le plat fade, mais à quoi bon le rendre indigeste ?  
Ainsi, seuls de grands cuisiniers se sont exercés à le doser avec sagesse.

# Cahier des Charges : Développement d'une Application d'Aide Comptable en Java

Équipe de Développement

May 21, 2024

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contexte . . . . .	3
1.2	Objectif . . . . .	3
<b>2</b>	<b>Présentation du Projet</b>	<b>3</b>
2.1	Nom du Projet . . . . .	3
2.2	Équipe Projet . . . . .	3
<b>3</b>	<b>Périmètre du Projet</b>	<b>3</b>
3.1	Fonctionnalités Principales . . . . .	3
<b>4</b>	<b>Spécifications Fonctionnelles</b>	<b>4</b>
4.1	Gestion des Factures . . . . .	4
4.2	Gestion des Paiements . . . . .	5
4.3	Rapports Financiers . . . . .	5
4.4	Gestion des Comptes . . . . .	5
4.5	Sécurité et Authentification . . . . .	6
<b>5</b>	<b>Spécifications Techniques</b>	<b>6</b>
5.1	Technologies Utilisées . . . . .	6
5.2	Architecture . . . . .	6
<b>6</b>	<b>Plan de Projet</b>	<b>6</b>
6.1	Phase de Planification (2 semaines) . . . . .	6
6.2	Phase de Développement (8 semaines) . . . . .	7
6.3	Phase de Test (2 semaines) . . . . .	7
6.4	Phase de Déploiement (2 semaines) . . . . .	7
6.5	Phase de Maintenance et Support (Ongoing) . . . . .	8
<b>7</b>	<b>Gestion de Projet</b>	<b>8</b>
7.1	Outils de Gestion . . . . .	8
7.2	Méthodologie Agile . . . . .	8

<b>8</b>	<b>Annexes</b>	<b>8</b>
8.1	Diagrammes UML . . . . .	8
8.2	Mockups des Interfaces Utilisateur . . . . .	8

# 1 Introduction

## 1.1 Contexte

Les petites et moyennes entreprises (PME) rencontrent souvent des difficultés pour gérer efficacement leur comptabilité. L'objectif est de développer une application comptable en Java qui facilitera la gestion des factures, des paiements, des rapports financiers et d'autres tâches comptables courantes.

## 1.2 Objectif

Créer une application conviviale, sécurisée et extensible pour répondre aux besoins comptables des PME.

# 2 Présentation du Projet

## 2.1 Nom du Projet

Application d'Aide Comptable (AAC)

## 2.2 Équipe Projet

- Chef de Projet
- Développeurs Frontend (JavaFX)
- Développeurs Backend (Spring Boot)
- Spécialistes Base de Données (MySQL/PostgreSQL)
- Spécialistes Sécurité (Spring Security)
- Testeurs

# 3 Périmètre du Projet

## 3.1 Fonctionnalités Principales

### 1. Gestion des Factures

- Création, édition et suppression de factures
- Envoi de factures par email
- Suivi des factures payées et impayées

### 2. Gestion des Paiements

- Enregistrement des paiements
- Génération de reçus
- Rappels de paiement pour les factures en attente

### 3. Rapports Financiers

- Génération de rapports financiers mensuels et annuels
- Analyse des flux de trésorerie
- Bilan et compte de résultat

### 4. Gestion des Comptes

- Suivi des comptes clients et fournisseurs
- Gestion des écritures comptables
- Réconciliation bancaire

### 5. Sécurité et Authentification

- Gestion des utilisateurs et des rôles (administrateur, comptable, etc.)
- Sécurisation des données sensibles

## 4 Spécifications Fonctionnelles

### 4.1 Gestion des Factures

- **Création de Factures**

- Formulaire pour saisir les détails de la facture (client, produits/services, montant, date)
- Validation des champs obligatoires
- Génération d'un PDF de la facture

- **Édition et Suppression de Factures**

- Modification des factures existantes
- Suppression des factures avec confirmation

- **Envoi par Email**

- Intégration avec une API d'envoi d'emails
- Personnalisation du message d'email

- **Suivi des Factures**

- Marquage des factures comme payées ou impayées
- Filtrage et recherche des factures

## 4.2 Gestion des Paiements

- **Enregistrement des Paiements**
  - Formulaire pour enregistrer les paiements reçus
  - Association des paiements aux factures correspondantes
- **Génération de Reçus**
  - Création de reçus de paiement au format PDF
  - Envoi de reçus par email
- **Rappels de Paiement**
  - Génération automatique de rappels pour les factures en attente
  - Envoi de rappels par email

## 4.3 Rapports Financiers

- **Rapports Mensuels et Annuels**
  - Génération de rapports financiers détaillés
  - Visualisation graphique des données financières
- **Analyse des Flux de Trésorerie**
  - Suivi des entrées et sorties de trésorerie
  - Prévisions de trésorerie
- **Bilan et Compte de Résultat**
  - Calcul automatique des bilans et comptes de résultat
  - Exportation des rapports au format Excel/PDF

## 4.4 Gestion des Comptes

- **Comptes Clients et Fournisseurs**
  - Création et gestion des fiches clients et fournisseurs
  - Suivi des transactions par compte
- **Écritures Comptables**
  - Saisie des écritures comptables
  - Validation et vérification des écritures
- **Réconciliation Bancaire**
  - Importation des relevés bancaires
  - Correspondance automatique des transactions

## 4.5 Sécurité et Authentification

- **Gestion des Utilisateurs**
  - Création de comptes utilisateurs avec rôles définis
  - Gestion des permissions d'accès
- **Sécurisation des Données**
  - Chiffrement des données sensibles
  - Sauvegardes régulières et restauration des données

## 5 Spécifications Techniques

### 5.1 Technologies Utilisées

- **Langage de Programmation** : Java
- **Interface Utilisateur** : JavaFX
- **Backend** : Spring Boot
- **Base de Données** : MySQL ou PostgreSQL
- **Sécurité** : Spring Security
- **Contrôle de Version** : Git/GitHub

### 5.2 Architecture

- **Architecture MVC** : Séparation des couches Modèle, Vue et Contrôleur
- **API REST** : Utilisation de Spring Boot pour exposer des services RESTful
- **Base de Données Relationnelle** : Utilisation de MySQL ou PostgreSQL avec JPA/Hibernate

## 6 Plan de Projet

### 6.1 Phase de Planification (2 semaines)

- **Définition des Exigences**
  - Collecter les besoins fonctionnels et non-fonctionnels
  - Rédiger les user stories et les cas d'utilisation
- **Conception de l'Architecture**
  - Définir l'architecture globale de l'application
  - Créer des diagrammes UML (cas d'utilisation, classes, séquences)



## 6.2 Phase de Développement (8 semaines)

- **Mise en Place de l'Environnement**
  - Configurer les environnements de développement et de test
  - Mettre en place l'intégration continue avec Jenkins ou GitHub Actions
- **Développement du Frontend**
  - Développer les interfaces utilisateur avec JavaFX
  - Intégrer les formulaires et les vues
- **Développement du Backend**
  - Implémenter les services RESTful avec Spring Boot
  - Gérer les connexions à la base de données et la logique métier

## 6.3 Phase de Test (2 semaines)

- **Tests Unitaires**
  - Écrire des tests unitaires pour les services backend avec JUnit et Mockito
  - Utiliser TestFX pour les tests unitaires du frontend
- **Tests d'Intégration**
  - Valider les interactions entre les différentes couches de l'application
- **Tests Utilisateurs**
  - Réaliser des tests utilisateurs pour valider l'expérience utilisateur

## 6.4 Phase de Déploiement (2 semaines)

- **Documentation**
  - Créer une documentation utilisateur détaillée
  - Documenter le code et les API
- **Formation des Utilisateurs**
  - Organiser des sessions de formation pour les utilisateurs finaux
- **Déploiement**
  - Déployer l'application sur un serveur de production
  - Configurer la surveillance et les alertes

## **6.5 Phase de Maintenance et Support (Ongoing)**

- **Support Utilisateur**

- Mettre en place un support utilisateur pour résoudre les problèmes et répondre aux questions

- **Mises à Jour**

- Planifier des mises à jour régulières pour améliorer l'application et ajouter de nouvelles fonctionnalités

## **7 Gestion de Projet**

### **7.1 Outils de Gestion**

- Utilisation de Trello pour le suivi des tâches
- Communication via Slack
- Contrôle de version avec GitHub

### **7.2 Méthodologie Agile**

- Utilisation des principes Agile pour gérer le projet
- Division du développement en sprints de 2 semaines
- Réunions quotidiennes (scrum meetings) pour faire le point sur les progrès et les obstacles

## **8 Annexes**

### **8.1 Diagrammes UML**

- Cas d'utilisation
- Diagrammes de classes
- Diagrammes de séquence

### **8.2 Mockups des Interfaces Utilisateur**

- Écrans de gestion des factures
- Écrans de gestion des paiements
- Rapports financiers