

# Cahier des Charges : Logiciel de Gestion des Délibérations du BFEM

Ce document détaille les spécifications techniques et fonctionnelles pour le développement d'un logiciel de gestion des délibérations du BFEM (Brevet de Fin d'Études Moyennes) au Sénégal. Il couvre le contexte, les objectifs, le périmètre, le public cible, la charte graphique, les fonctionnalités requises, l'architecture, le processus de délibération, la planification, les contraintes, la sécurité, les livrables et la répartition des tâches.



# Contexte et Objectifs du Projet

Le projet vise à automatiser la gestion des données et la délibération des candidats au BFEM au Sénégal. L'objectif principal est de simplifier la saisie des notes, le calcul des points, et la génération des résultats, tout en assurant le respect des règles métiers telles que les bonus/malus et le repêchage.

Le logiciel doit également produire des documents officiels comme les PV de délibération, les relevés de notes, et les listes d'anonymat. La centralisation des données se fera via une base SQLite et une interface graphique intuitive.

## Simplification

Saisie des notes et calcul des points.

## Conformité

Respect des règles métiers (bonus/malus).

## Production

Génération de documents officiels.



# Périmètre du Projet

Le projet inclut la gestion d'un seul jury (session Moderne PC/LV2), l'enregistrement des candidats et de leurs notes (1<sup>er</sup> et 2<sup>e</sup> tour), le calcul des points, bonus/malus, et règles de repêchage. Il comprend également la génération de PDF (listes, résultats, PV) et des statistiques basiques (taux de réussite, répartition des notes).

Le projet exclut la prise en charge de plusieurs jurys simultanés, la gestion des autres sessions (ex : Arabe), et le déploiement mobile ou web.

## Inclus

Gestion d'un jury.

Enregistrement des candidats et notes.

Calcul des points et règles de repêchage.

Génération de PDF.

Statistiques basiques.

## Exclus

Plusieurs jurys simultanés.

Autres sessions (Arabe).

Déploiement mobile ou web.



# Public Cible

Les utilisateurs principaux du logiciel sont les membres du jury (présidents, correcteurs), les administrateurs académiques (IEF, IA), et les secrétaires chargés de la saisie des données. Le logiciel doit être conçu pour répondre aux besoins spécifiques de ces différents utilisateurs.



## Membres du jury

Présidents,  
correcteurs.



## Administrateurs

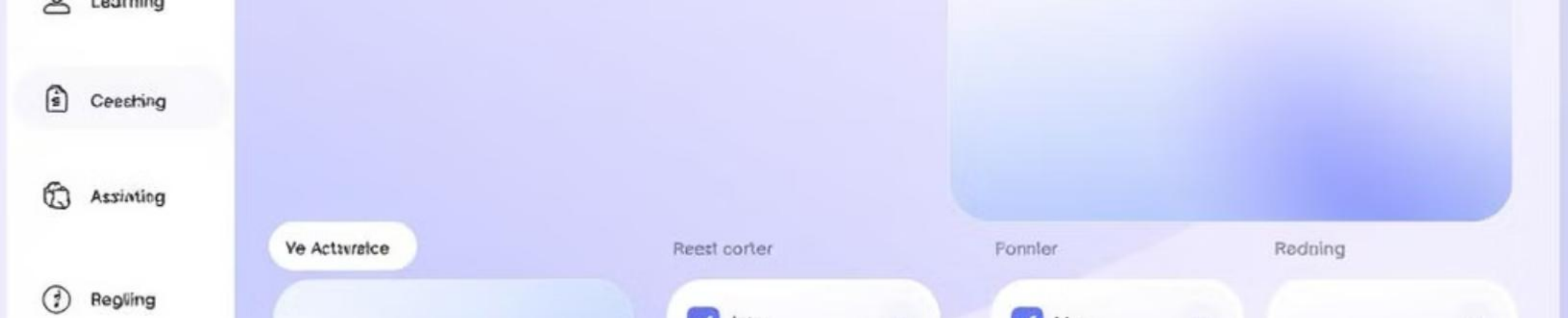
IEF, IA.



## Secrétaires

Saisie des données.





# Charte Graphique et Ergonomie

L'interface du logiciel sera développée avec PyQt5 pour un rendu professionnel. Le design sera minimaliste, avec une palette de couleurs institutionnelles (bleu, blanc). La navigation sera assurée par des menus clairs et des formulaires structurés par onglets. L'accessibilité sera prise en compte avec des tailles de police ajustables et des messages d'erreur explicites.

**1**

## Framework

Tkinter (rendu professionnel).

**2**

## Design

Minimaliste (bleu, blanc).

**3**

## Navigation

Menus clairs, formulaires structurés.

**4**

## Accessibilité

Tailles de police, messages d'erreur.

# Fonctionnalités Requises et Techniques

Le logiciel doit permettre le paramétrage du jury (Région, IEF, Président), le CRUD des candidats (ajout, modification, suppression), et la génération d'anonymat (numéros uniques). Il doit gérer le calcul automatique des points, les tours de délibération, et générer des rapports PDF. Des statistiques (moyennes, taux de réussite) et l'import/export depuis Excel sont requis.

Techniquement, le logiciel sera développé en Python 3.10+ avec une base de données SQLite. Les bibliothèques utilisées incluent Tkinter (GUI), ReportLab (PDF), et pandas (statistiques).

1

## Paramétrage

Informations du jury.

2

## CRUD

Gestion des candidats.

3

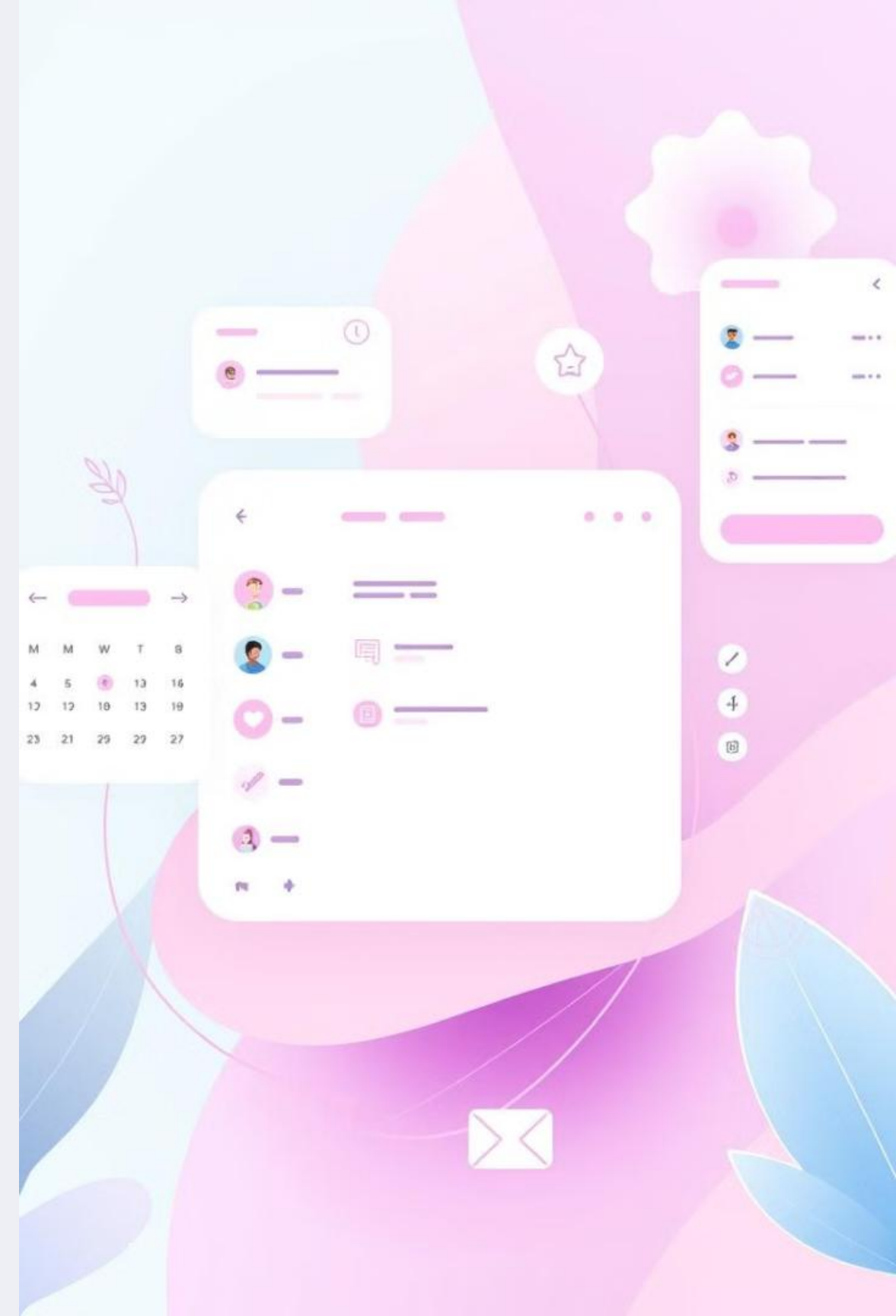
## Délibération

Calcul des points, tours.

4

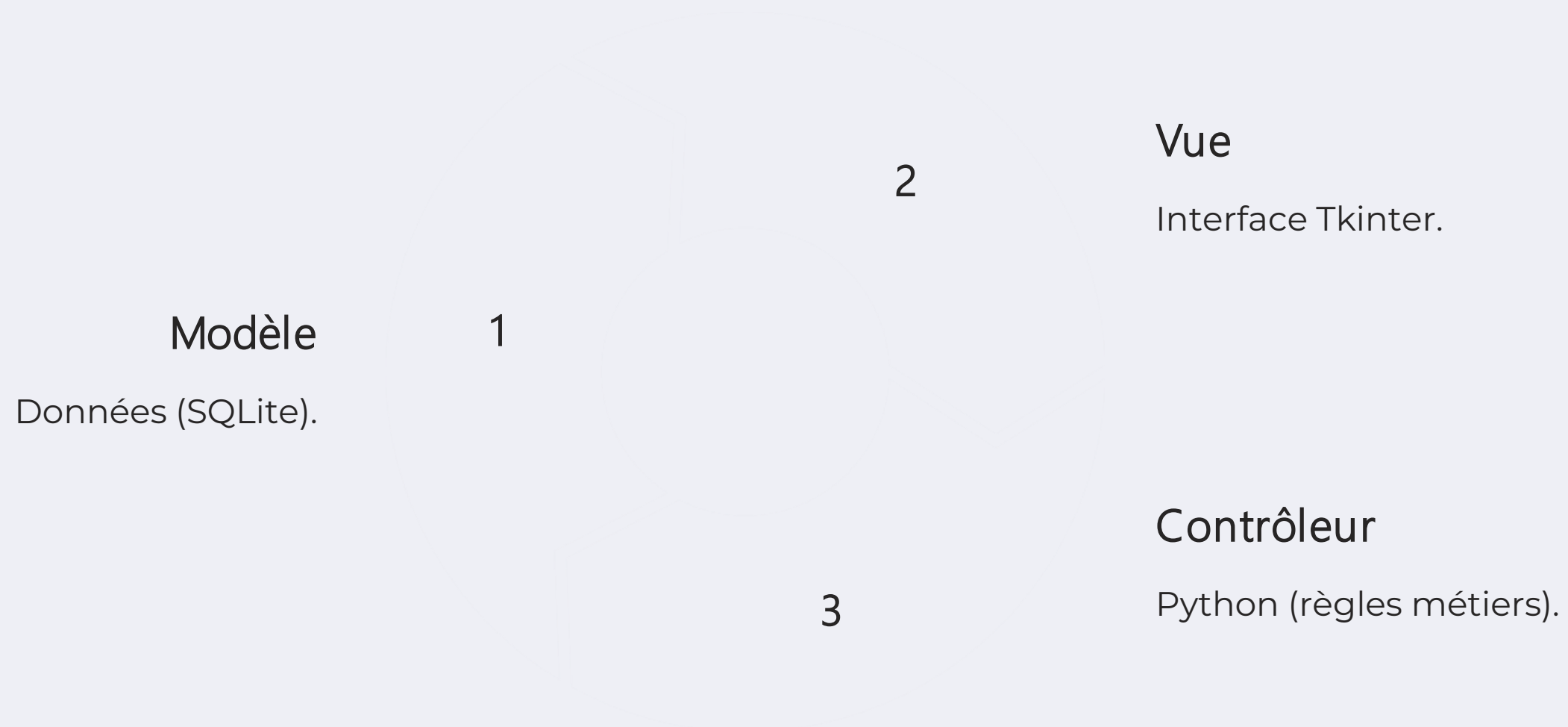
## Rapports

Génération de PDF.



# Architectures et Technologies

L'architecture du logiciel suivra le modèle MVC (Modèle-Vue-Contrôleur). La couche données utilisera SQLite (tables Candidat, LivretScolaire, Notes). La couche métier sera en Python (calcul des points, règles de repêchage). La couche présentation sera l'interface Tkinter. SQLAlchemy sera utilisé pour l'ORM et Git pour le versionnage.



# Processus de Délibération et Édition des Résultats

Au 1<sup>er</sup> tour, les points totaux sont calculés (notes × coefficients + bonus EPS/EF). Les candidats sont classés en Admis ( $\geq 180$  pts), Repêchable (171–179,9 pts), Second tour (153–179,9 pts), et Échec ( $< 153$  pts). Au 2<sup>e</sup> tour, une réévaluation est effectuée avec des coefficients spécifiques (Français ×3, Maths ×3, PC/LV2 ×2). Les candidats sont admis si  $\geq 76$  pts. Les relevés et PV sont générés automatiquement en PDF.

1

## 1<sup>er</sup> Tour

Calcul des points, classement.

---

2

## 2<sup>e</sup> Tour

Réévaluation, admission.

---

3

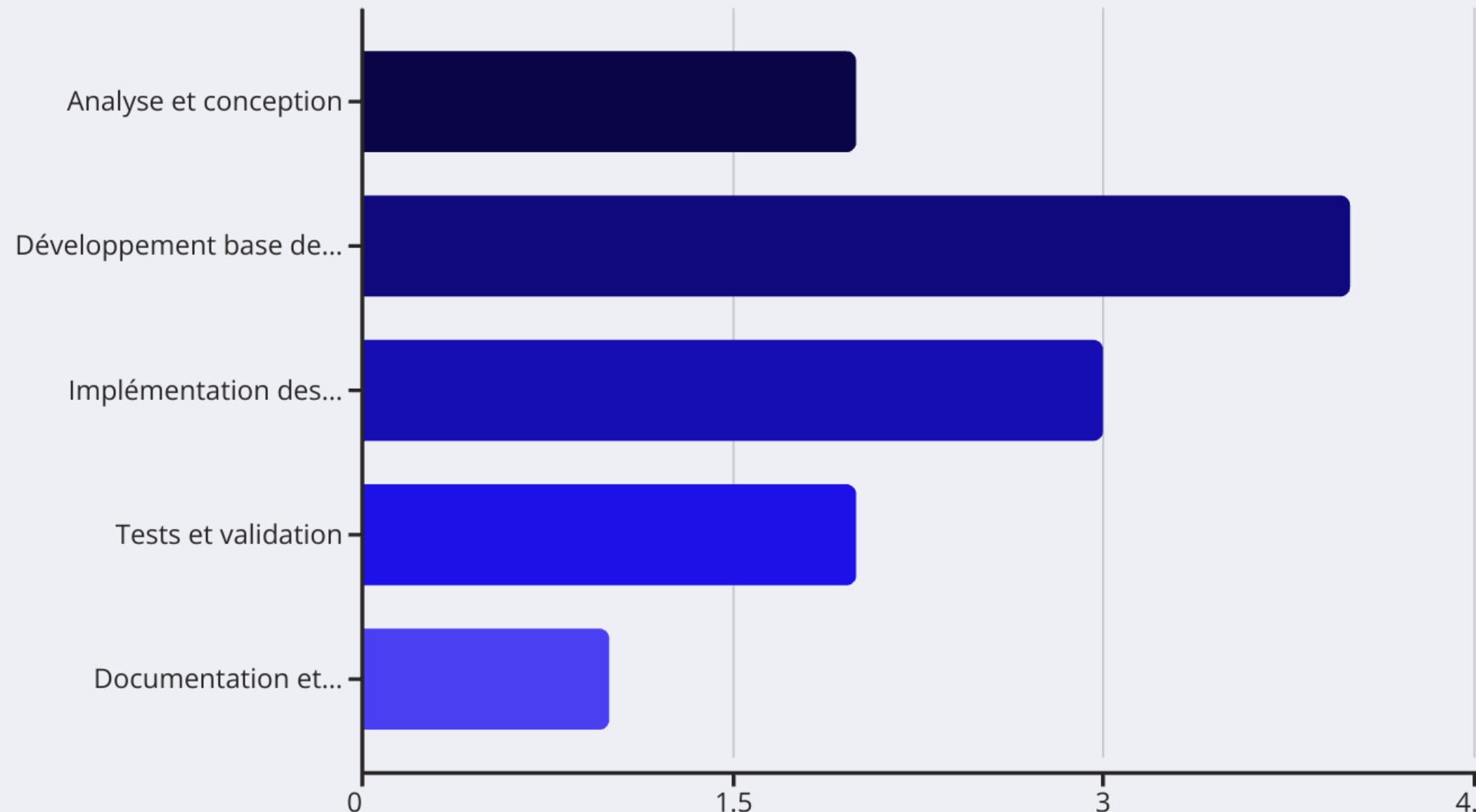
## Édition

Génération des relevés et PV.



# Planification et Contraintes Techniques

Le projet est planifié en plusieurs phases : analyse et conception (2 mois), développement base de données + GUI (4 mois), implémentation des règles métiers (3 mois), tests et validation (2 mois), documentation et déploiement (1 mois). La livraison est prévue pour le 23/02/2025. Le logiciel doit être compatible avec Windows 10/11 et gérer jusqu'à 1 000 candidats sans latence. L'installation des bibliothèques Python requises est une contrainte.



# Sécurité, Confidentialité et Livrables

L'anonymat est assuré par l'attribution de numéros uniques pour les corrections. L'accès est protégé par l'authentification des administrateurs. Les données sont sécurisées par le chiffrement de la base SQLite. Les livrables incluent le code source (dépôt GitHub), l'exécutable (.exe), la documentation (manuel utilisateur, guide technique), et les données de test (fichier Excel simulé).



## Sécurité

Chiffrement de la base SQLite.



## Code source

Dépôt GitHub.



## Documentation

Manuel utilisateur, guide technique.

# Membres du groupe



ARAME DIENG



IBRAHIMA THIAM



MARIE SOLANGE S.T. DIOUF