**Plan détaillé du projet POO2 : Logiciel de gestion du BFEM en PyQt5**

**Membres du Groupe : Mohamet Lamine BA et Mouhamadou AL Bachir BA**

**📌 Introduction**

Ce projet consiste à concevoir un **logiciel de gestion des candidats et de la délibération des examens du BFEM**. Il intégrera une interface graphique en **PyQt5**, une **base de données SQLite**, et des fonctionnalités permettant la gestion des candidats, la saisie des notes, la génération d’anonymats et l’automatisation des décisions.

**📌 1. Architecture Générale**

**🔹 Technologie utilisée :**

* **Backend** : Python + SQLite
* **Frontend (Interface graphique)** : PyQt5
* **Export des données** : Génération de PDF (reportlab, PyQt5 QPrinter)
* **Gestion des fichiers** : Importation/exportation en Excel (pandas, openpyxl)

**🔹 Structure Modulaire du Projet**

1. **Base de données** : Stockage des informations des candidats, jurys et résultats.
2. **Interface utilisateur (UI)** : Gestion via des fenêtres PyQt5.
3. **Génération d’anonymats** : Attribution automatique des numéros pour la correction.
4. **Délibération automatique** : Calcul des résultats selon les règles métier.
5. **Statistiques et rapports** : Affichage des tendances et export en PDF.

**📌 2. Fonctionnalités du Logiciel**

**🔹 Paramétrage du Jury**

* Définition de l’**Inspection Académique (IA)**, de l’**IEF (Département)**, de la **localité**, du **centre d’examen** et du **président du jury**.
* Gestion des **membres du jury** (ajout, modification, suppression).

**🔹 Gestion des Candidats**

* **Ajout, modification, suppression, recherche de candidats** (CRUD).
* Enregistrement des informations personnelles :
  + **Numéro de table, prénom, nom, date/lieu de naissance, sexe, nationalité.**
  + **Choix de l’épreuve facultative et aptitude sportive.**

**🔹 Génération Automatique des Anonymats**

* Attribution **aléatoire** d’un numéro d’anonymat unique.
* Association des anonymats à chaque matière et tour d’examen.

**🔹 Gestion des Notes et Délibération**

* **Saisie des notes** pour chaque matière (premier et second tour).
* **Calcul automatique des résultats** selon les règles métier :
  + Passage d’office, passage au second tour, échec, repêchage.
  + Bonus/malus pour l’EPS et l’épreuve facultative.

**🔹 Gestion des Repêchages**

* Détection automatique des candidats repêchables.
* Gestion des conditions : moyenne de cycle, nombre de tentatives, seuils de points.

**🔹 Impression et Export**

* **Génération de listes PDF** :
  + Liste des candidats, liste des anonymats, relevés de notes, PV de délibération.
* **Export des résultats** en **Excel** pour archivage.

**🔹 Statistiques**

* Graphiques sur les taux de réussite.
* Comparaison des performances par matière et par centre d’examen.

**📌 3. Menu de l'Application**

L’interface PyQt5 comportera **une barre de menus** et **un tableau de bord central** avec des onglets pour chaque fonctionnalité.

**🔹 Menu Principal**

* **Fichier**
  + 📁 **Nouvelle session** (réinitialiser la base pour une nouvelle année).
  + 📤 **Exporter données** (PDF, Excel).
  + illustrations, cliparts, dessins animés et icônes de creative black open book icon vector design - livre ouvert**Ouvrir**
* **Gestion des Utilisateurs**
  + ➕ **Ajouter un utilisateur**
  + ❌ **Supprimer un utilisateur**
* **Gestion des Candidats**
  + ➕ Ajouter un candidat
  + 🔄 Modifier un candidat
  + ❌ Supprimer un candidat
  + 🔍 Rechercher un candidat
* **Notes et Délibération**
  + 🖊️ Saisie des notes (1er tour / 2nd tour)
  + 🔢 Calcul automatique des résultats
  + 🎓 Gestion des repêchages
* **Statistiques & Rapports**
  + 📊 Visualisation des tendances (taux de réussite, moyenne par matière)
  + 📄 Génération des PV et relevés de notes
* **Paramètres**
  + ⚙️ Configuration du jury
  + 🔄 Réinitialisation des données
* **Aide**
  + 📖 Manuel utilisateur
  + ℹ️ À propos
* **→ Déconnexion**
* ❌ **Quitter**

**📌 4. Base de Données SQLite**

Les tables principales seront :

1. **Candidats** (infos personnelles)
2. **Notes** (toutes les matières et coefficients)
3. **Jury** (gestion des examinateurs)
4. **Anonymats** (associations candidat/matière/tour)
5. **Délibération** (résultats finaux et repêchages)

**📌 5. Plan de Développement**

**Phase 1 : Conception et Modélisation (1 semaine)**

* Modèle conceptuel et relationnel (BD SQLite).
* Wireframe de l’interface graphique PyQt5.

**Phase 2 : Développement Back-end (2 semaines)**

* Création des classes en Python (POO).
* Connexion SQLite, gestion des requêtes SQL.
* Algorithmes pour l’anonymat et la délibération.

**Phase 3 : Développement Front-end (2 semaines)**

* Création des fenêtres PyQt5.
* Formulaires de saisie et validation des entrées.

**Phase 4 : Tests et Optimisation (1 semaine)**

* Vérification de la logique métier.
* Tests unitaires et correction des bugs.

**Phase 5 : Documentation et Livraison (1 semaine)**

* Manuel utilisateur et guide d’installation.
* Dépôt GitHub et finalisation du projet.

**📌 Conclusion**

Ce projet fournira un **logiciel complet** de gestion du BFEM, assurant une gestion efficace des candidats et de la délibération. **L’utilisation de PyQt5** garantira une interface intuitive et fluide. **L’intégration de statistiques et d’exports** facilitera l’analyse et la communication des résultats.