

I- Contexte et besoins :

Solution à mettre en place :

Pour satisfaire les attentes de l'entreprise GSB, l'application va devoir respecter un certain nombre de contraintes.

Dans un premier temps, il faut que l'application fasse appel aux API (*Application Programming Interface*); l'application permettra donc aussi bien la visualisation que la modification, la suppression et l'ajout de données via une interface contenant le composant TableView.

L'application permettra aussi de retrouver une liste de médecins visités par des visiteurs de GSB en fonction de son nom. (On pourra faire une recherche en fonction du département)

Le fonctionnement de la solution repose sur le Framework JAVAFX sous SDK 21 ainsi que des langages JAVA et CSS. Les requêtes de récupération et d'envoi de données en direction de la base se feront donc par le biais de JDBC (JAVA Database connectivity).

1er Sprint :

Réaliser l'interface d'authentification

Les objectifs de cette étape c'est:

- ✓ S'authentifier
- ✓ Saisir une fiche de renseignement concernant un médecin :
 - Nom et prénom*
 - Adresse.
 - Date de naissance.
- ✓ Le visiteur pourra faire les opérations de base (s'authentifier- Ajout-Suppression- Modification- Recherche)
- ✓ Il pourra aussi avoir des statistiques concernant les nombres de médecin par rapport à l'année de naissance.

Quelques interfaces attendues:

Authentification:

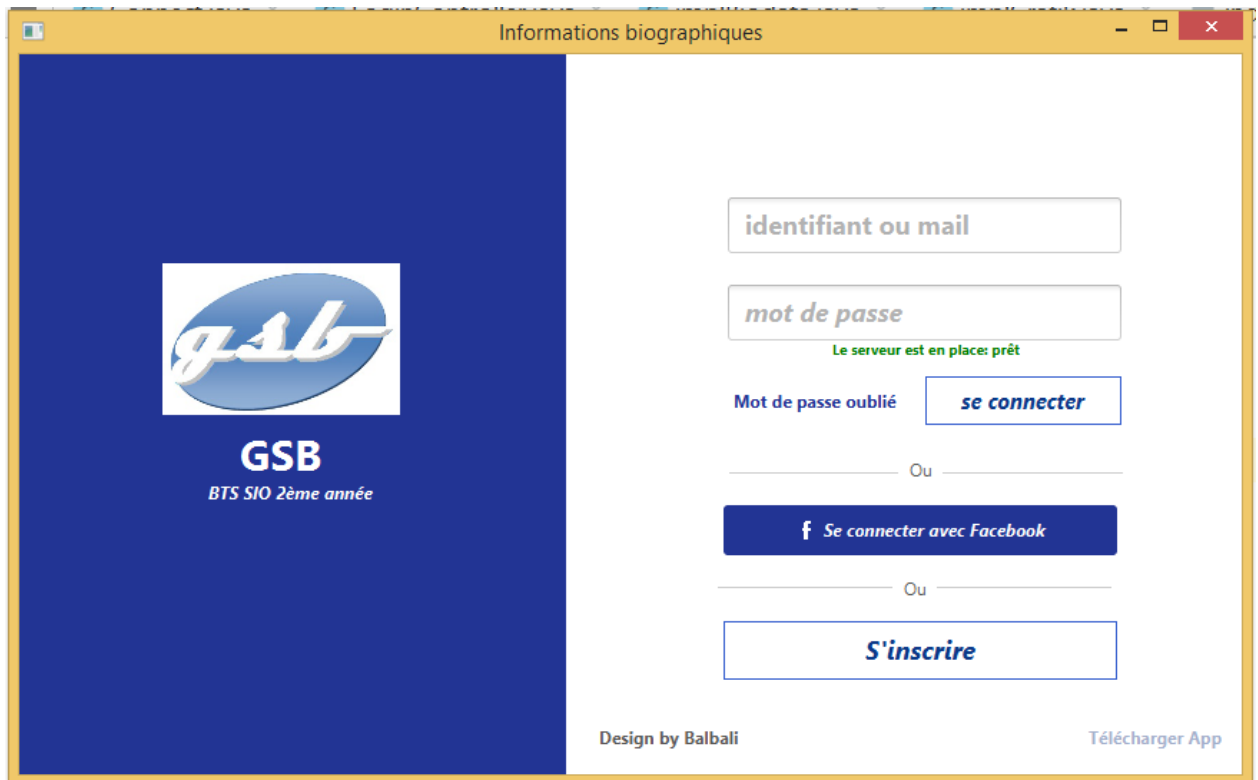
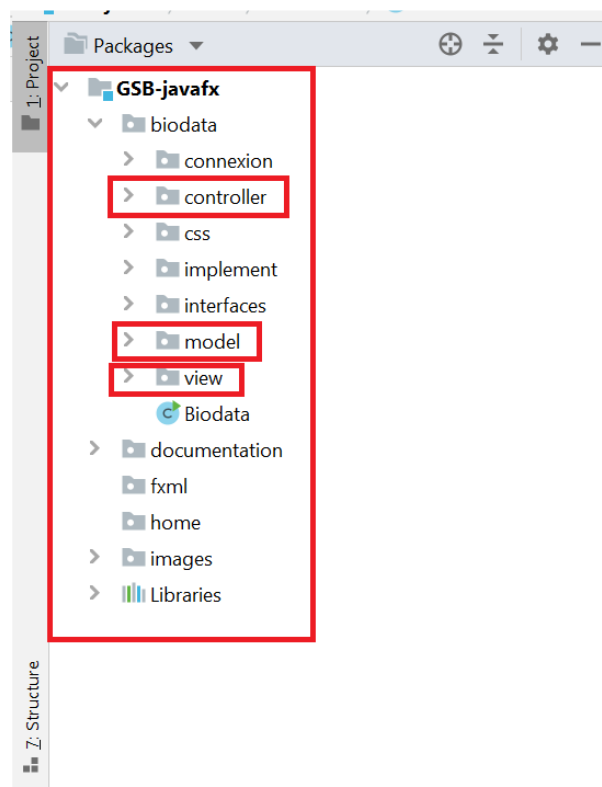


Illustration 1- Interface d'authentification

Architecture de l'application doit respecter le modèle MVC :



2ème Sprint :

Réaliser l'interface de gestion des données stockées des médecins visités par les visiteurs de GSB.

Les objectifs de cette étape c'est:

- ✓ de permettre d'afficher les informations à partir de la table **tablebiodata** dans la base de données **biodata** au niveau d'un **tableview** au niveau de l'interface **biodata.fxml**.

Nous allons commencer par le fait de manipuler que le nom, l'adresse et la date de naissance.

- ✓ effectuer la saisie à partir d'un formulaire contenant un champ ID, un champ Nom , un autre **textarea** contenant l'adresse, enfin, pour la date de naissance, on va utiliser un **DatePicker** pour récupérer la date automatiquement.

Chaque composant aura un label spécifique comme indiqué dans l'illustration 1 de l'interface ci-dessus sous **SceneBuilder**.

- ✓ effectuer la modification des enregistrements en cliquant sur **editer** au niveau de la **tableview**.
- ✓ effectuer la suppression à partir de la **tableView** directement avec confirmation.

Quelques interfaces attendues :



Illustration 2- Conception et réalisation de l'interface biodata.fxml sous ScenBuilder

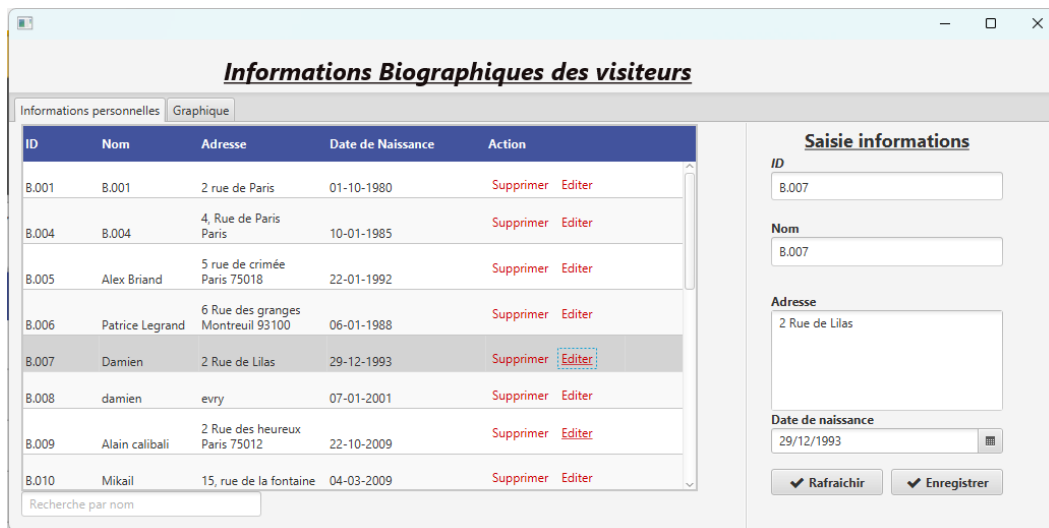


Illustration 3- L'interface de gestion des informations lors de l'exécution (onglet informations personnelles)

3^{ème} Sprint :

Utilisation avancée des graphiques sous JavaFX et réalisation des statistiques

Les objectifs de cette étape c'est de :

1. Permettre d'afficher un graphique en barre à partir des statistiques des données de la table tablebiodata dans la base de données biodata au niveau d'un tableview des informations affichées en deux colonnes (année, nombre de personnes), au niveau du graphique, il y a l'année (axe des abscisses X) et le nombre de personnes (axe des ordonnées Y au niveau de l'interface grafik.fxml).
2. Utiliser le composant StackedBarChart et la classe XYChart pour pouvoir afficher un graphique à barres pour récupérer le nombre de personnes en fonction des années. (voir l'interface grafik.fxml et l'implémentation de l'interface interGrafik au niveau implGrafik).
3. effectuer la modification de l'affichage en cliquant sur le bouton rafraîchir au niveau de l'interface grafik.fxml si on modifie les données de naissance dans l'interface biodata.fxml.

Quelques interfaces attendues :

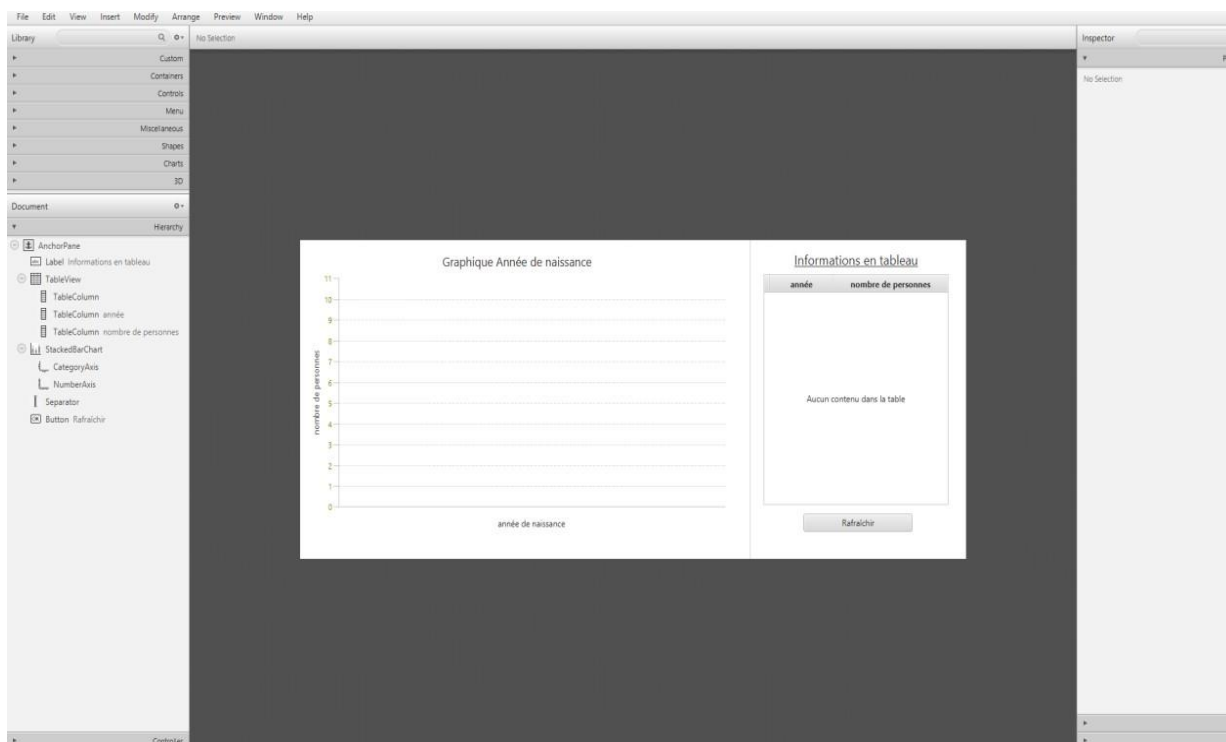


Illustration 4- L'interface grafik.fxml sous ScenBuilder

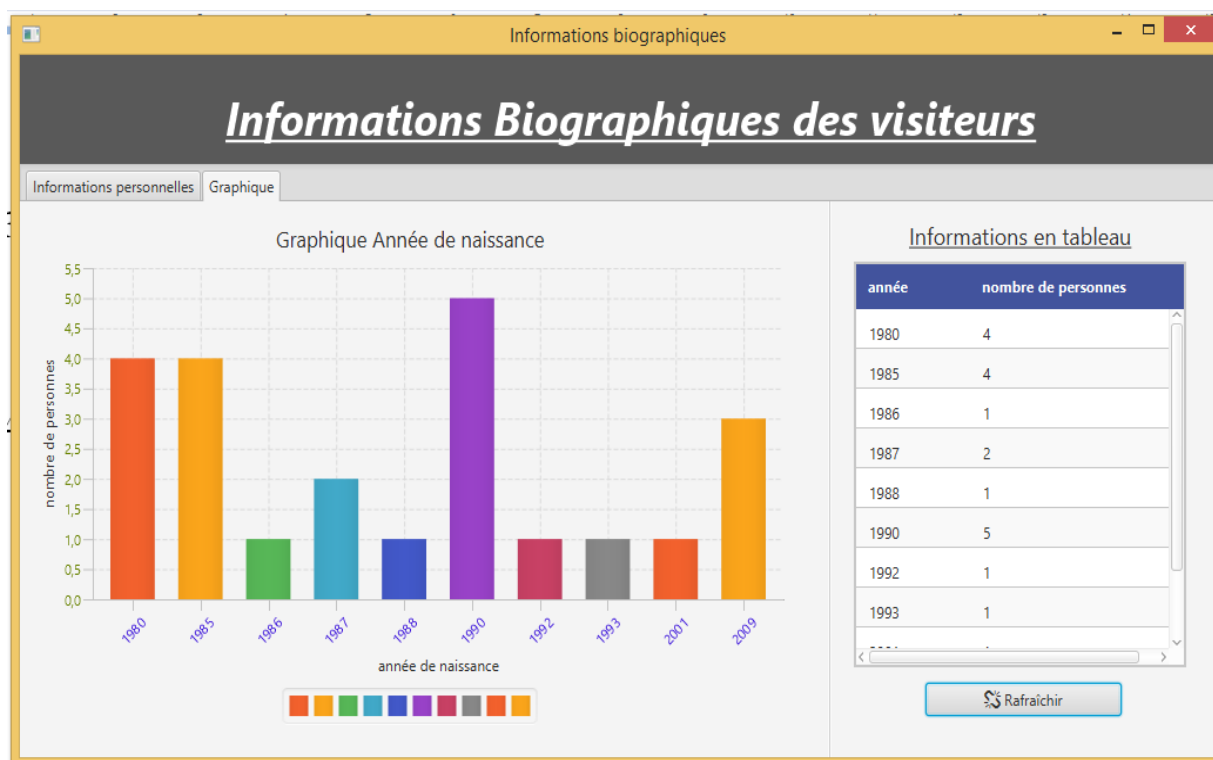


Illustration 5- L'interface de gestion des informations lors de l'exécution (onglet Graphique)

II – Description de l’environnement de réalisation

Environnement de développement

Le langage de script retenu est le Java en utilisant le framework JavaFx.

La solution utilise le SGBD MySql pour la base de données de l’application avec une connexion à l’aide de l’interface JDBC sous Eclipse (IntelliJ ou Visual Studio).

Environnement de production

- N’importe quel système d’exploitation, avec un serveur apache et un SGBD Mysql/MariaDB
- Le code source est partagé sur github.