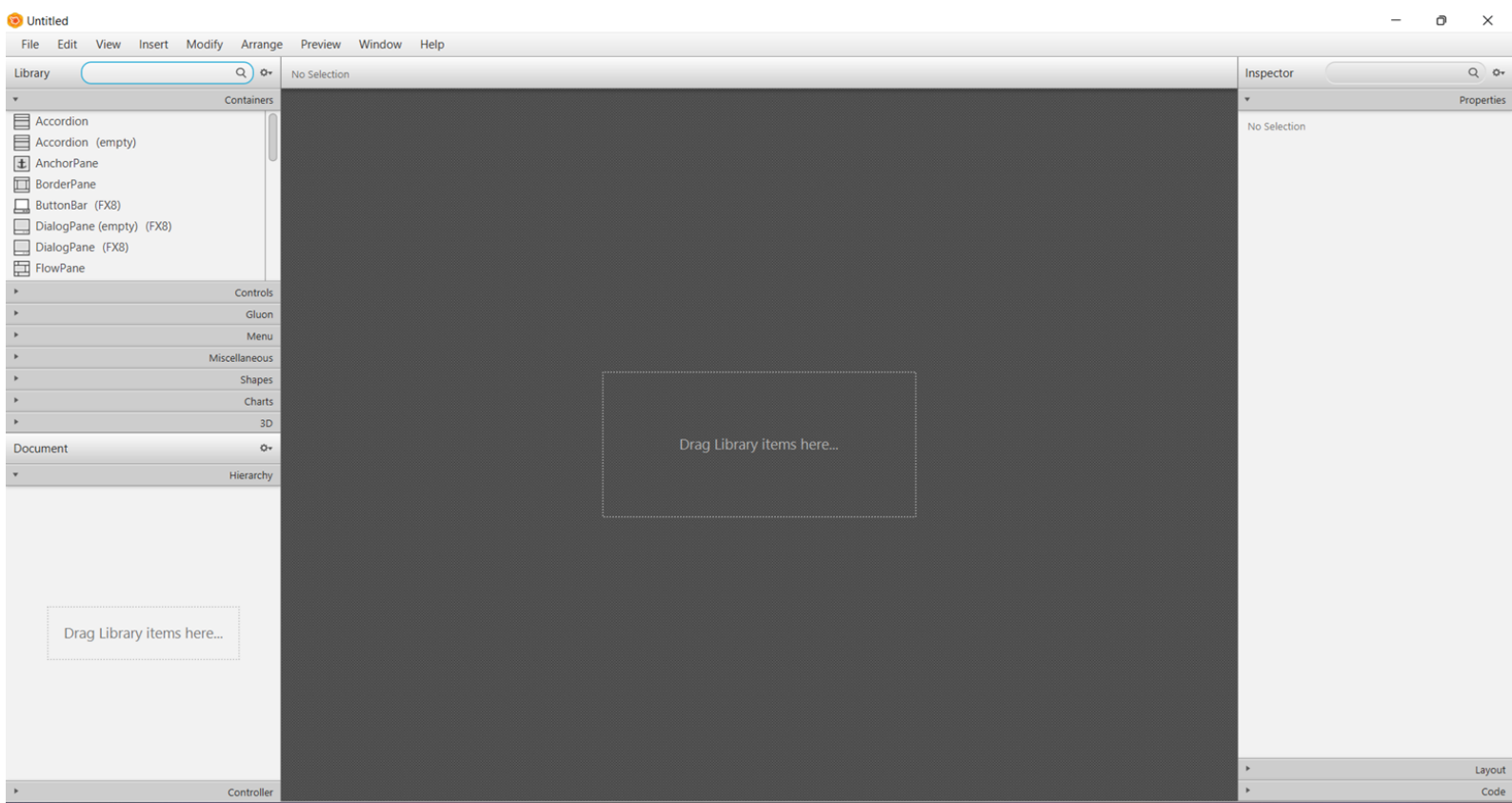


Henocque
Antoine

Compte-rendu du projet pour Mr Gahide.

Je me suis dans un premier temps familiarisé avec l'outil SceneBuilder (découverte de l'interface, test des différentes possibilités de création d'une page ...). Son utilisation est à la fois complexe sur certaines fonctionnalités, mais également instinctive.

Voici l'interface que l'on peut apercevoir lors du lancement de l'application, il ne reste plus qu'à créer une page :



J'ai eu la tâche de créer l'interface visiteur suite à sa connexion à l'application.

Pour m'aider, j'ai pris comme modèle le document laissé en pièce jointe sur le Github *rose-line*.

La page n'est pas complètement terminée, il me reste quelques zones de texte à insérer dessus.

ETAT DES FRAIS ENGAGES

A retourner accompagné des justificatifs au plus tard le 10 du mois qui suit l'engagement des frais

Visiteur

Matricule :

Nom :

Mois

Frais forfaitaires	Quantité	Montant unitaire	Total
Aucun contenu dans la table			

Autres frais

Date	Montant	Libellé
Aucun contenu dans la table		

Signature :

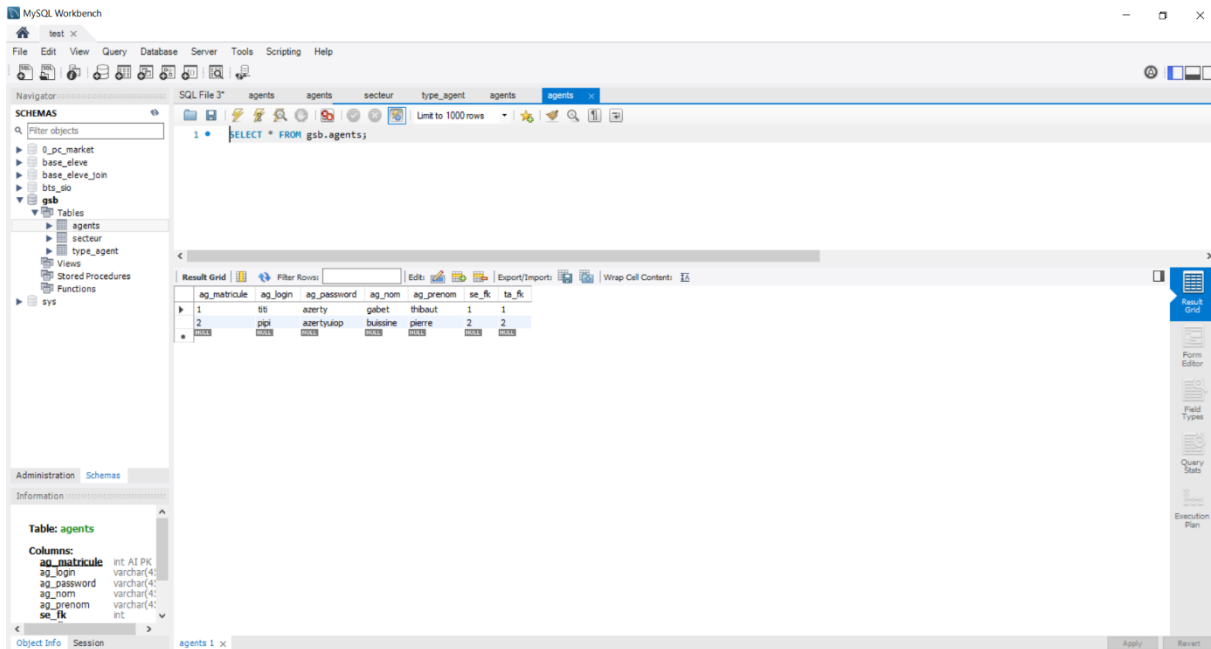
Pour le moment, la plus grosse difficulté que j'ai pu rencontrer fût la création des deux tableau ci-dessous.

Frais forfaitaires	Quantité	Montant unitaire	Total
Aucun contenu dans la table			
Autres frais			
Date	Montant	Libellé	
Aucun contenu dans la table			

Je ne parvenais pas à insérer plusieurs colonnes dans le tableau, mais j'y suis parvenu en regardant quelques tutos sur internet pour réaliser un tableau correct et agréable à regarder pour le visiteur. L'objectif ici, étant d'afficher à l'utilisateur l'ensemble des informations stockées dans la base de donnée.

Formulaire de connexion (Thibaut Gabet) :

Tout d'abord j'ai créé la base de données sur Worbench.



Ensuite j'ai créé les éléments nécessaires au formulaire avec SceneBuilder.

The screenshot shows a login form titled 'login'. It contains two input fields: 'user :' and 'mdp :'. Below the input fields is a button labeled 'connexion'.

login

user :

mdp :

connexion

Puis je suis passé aux codes du formulaire avec JavaFX.

Grace au code si dessous je récupéré le texte qu'a écrit l'utilisateur dans les casses login et mot de passe. S'il n'y a rien alors je ne me connecte pas à la bdd.

```
System.out.println( C_login.getText());
System.out.println( C_mdp.getText());

if (!C_login.getText().equals("") && !C_mdp.getText().equals("")) {

    String dbURL = "jdbc:mysql://localhost:3306/gsb";
    String username = "root";
    String password = "";

    try {

        Connection conn = DriverManager.getConnection(dbURL, username, password);

        if (conn != null) {
            System.out.println("Connected");
        }
    }
}
```

Je récupère les données de la Table agents pour les mettre dans des variable String.

```
//étape 3: créer l'objet statement
Statement stmt = conn.createStatement();
String sql = "SELECT ag_matricule, ag_login, ag_password, ag_prenom, ag_nom FROM agents";
ResultSet res = stmt.executeQuery(sql);

while(res.next()){
    //Récupérer par nom de colonne
    String matricule = res.getString("ag_matricule");
    String login = res.getString("ag_login");
    String mdp = res.getString("ag_password");
    String prenom = res.getString("ag_prenom");
    String nom = res.getString("ag_nom");
}
```

Ensuite je vérifie si les données de la Table agents concordent avec les données d'entrée du login et du mot de passe de l'utilisateurs qui souhaite se connecter. Si c'est le cas l'utilisateur et connecter et change de page.

(j'ai décidé d'afficher les données pourvoir vérifier que tout fonctionne correctement)

```
if (C_login.getText().equals(login) && C_mdp.getText().equals(mdp)) {  
System.out.print(", matricule: " + matricule);  
System.out.print(", login: " + login);  
System.out.print(", password: " + mdp);  
System.out.print(", prénom: " + prenom);  
System.out.print(", nom: " + nom);  
  
App.setRoot("secondary");
```

	ag_matricule	ag_login	ag_password	ag_nom	ag_prenom	se_fk	ta_fk
►	1	titi	azerty	gabet	thibaut	1	1

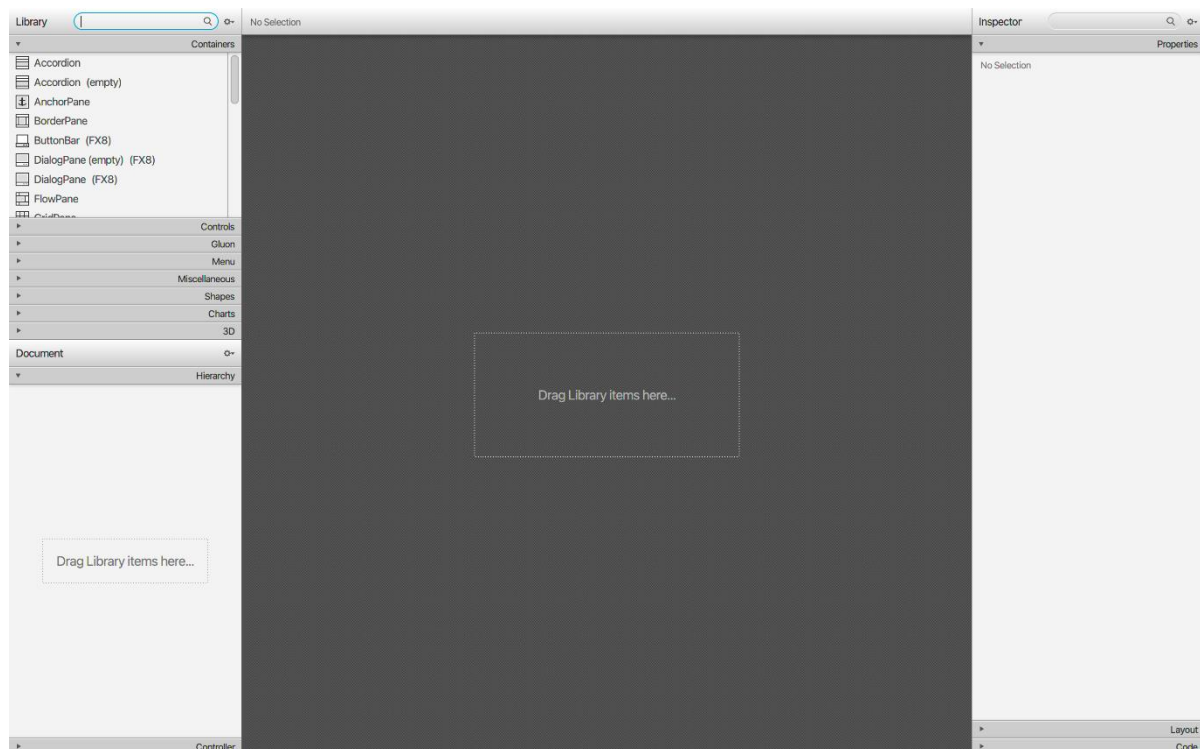
```
titi  
azerty  
Connected  
, matricule: 1, login: titi, password: azerty, prénom: thibaut, nom: gabet
```

Tâche que j'ai réalisée (Simon Ravaut) :

À l'aide de l'application SceneBuilder, ma tâche était de réaliser la page de connexion de l'application.

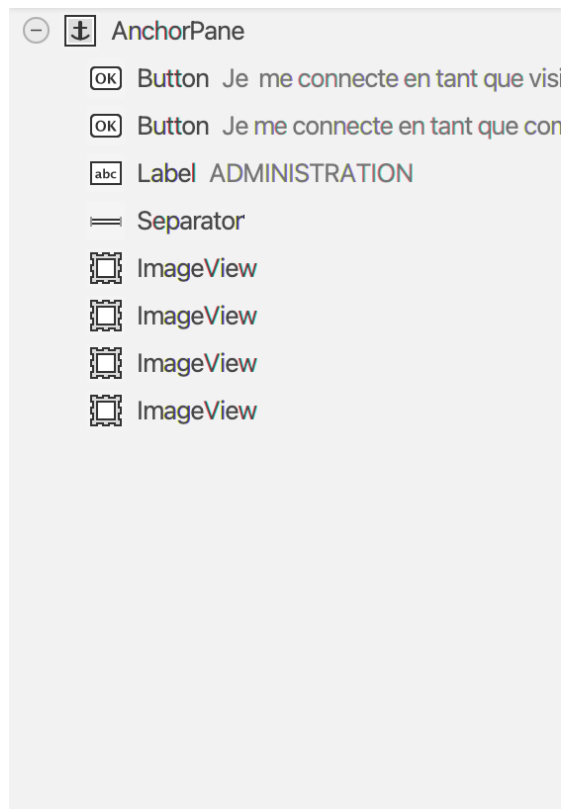
Cette page va permettre de s'identifier en tant que visiteur ou comptable. Selon le choix de connexion, la personne sera redirigée vers une autre page qui sera approprié au statut de la personne.

Voici ci-dessous l'application scène builder permettant de s'occuper de la partie graphique :



Ce logiciel permet d'exécuter des pages fxml .FXML est un format de données textuelles, dérivé du format XML, qui permet de décrire une interface utilisateur pour des applications conçus avec JavaFX.

Ci-dessous, on peut voir les objets que j'ai utilisés afin de conceptualiser la page de connexion. On y retrouve le type de page / bouton / label (texte) / ligne séparateur / ainsi que des images.



La page de connexion une fois terminée, sera reliée à la page d'accueil mais avant cela on va devoir la connecter à la base de données afin que les visiteurs ou les comptables puissent s'identifier avec de vrai login et mot de passe.

Sur Visual studio code, la bibliothèque javafx nous crée des pages FXML qu'on va pouvoir modifier afin de rajouter des actions pour des boutons ou encore relier différente page FXML lors d'une clique sur un bouton. Cela va permettre de passer d'une page à une autre.

On peut voir ci-dessous la page FXML que j'ai édité afin que lorsque l'on clique sur un bouton, il nous renvoie sur une deuxième page.

```
src > main > java > fr > simon2 > PrimaryController.java > ...
1  package fr.simon2;
2
3  import java.io.IOException;
4
5  import javafx.event.ActionEvent;
6  import javafx.fxml.FXML;
7  import javafx.scene.control.Button;
8
9  public class PrimaryController {
10
11      @FXML
12      private Button Cbutton;
13
14      @FXML
15      private Button Vbutton;
16
17      @FXML
18      void SwitchToComptable(ActionEvent event) throws IOException {
19          App.setRoot("secondary");
20      }
21
22      @FXML void fr.simon2.PrimaryController.SwitchToVisiteur(ActionEvent event) throws IOException
23      void SwitchToVisiteur(ActionEvent event) throws IOException {
24          App.setRoot("secondary");
25      }
26
27  }
28
```


Ci- dessous on peut voir le résultat graphique de nôtre page de connexion :

