

Thibaut Fiquet
Simon Basserue
Lucas Vanauberg
Benjamin Pluchart

Equipe Tuto.Java.Exe

Description du rapport de suivi de projet

- Le cahier des charges résumé tel que vous l'avez formalisé à partir des documents fournis .

Nous devons réaliser une application de gestion de frais. Il faut tout d'abord que l'application soit très précise, il faut que le visiteur puisse avoir accès à ses fiches de frais, renseigner ses fiches de frais et se connecter avec un accès spécial différents du comptable qui lui doit pouvoir valider les fiches de frais et suivre le paiement des frais. Chaque utilisateur doit posséder un accès différent avec des mots de passe et des matricules. Il y a différents types de frais (nuité, repas, kilométrique). Pour les frais de hors forfait le visiteur doit enregistrer la date, le montant et le libellé de la dépense. Il doit fournir au service comptable une facture acquittée. Le comptable devra ensuite valider les frais.

Quand l'utilisateur veut valider sa fiche, il doit y avoir une confirmation et une validation si le visiteur n'a aucune fiche de frais alors il y aura un message qui dit qu'il n'y a pas de fiche.

L'application doit d'abord être fonctionnelle avant de posséder une interface graphique.

Les utilisateurs doivent avoir la possibilité de changer de voiture ainsi que de secteur.

Il y doit y avoir une page de connexion ainsi que d'inscription avec le choix du secteur etc.

Le comptable doit pouvoir valider les frais entre le 10 et 20 du mois, après la clôture les visiteurs peuvent quand même accéder à leurs fiches mais ne peuvent pas les modifier.

Le système gardera les logs de connexion pour savoir qui est en ligne si le mot de passe est faux alors le système affichera une erreur.

Présenter un suivi daté des opérations menées par le service comptable (réception des pièces, validation de la demande de remboursement, mise en paiement, remboursement effectué).

L'application doit être accessible depuis un ordinateur que se soit visiteurs ou comptable.

Nous devons créer une documentation qui devra présenter l'arborescence des écrans pour chaque module, le descriptif des éléments, classes et bibliothèques utilisées, la liste des frameworks ou bibliothèques externes utilisés, la liste des divers outils de gestion de projet utilisés.

- La description technique détaillée des fonctionnalités correspondantes : terminées / en cours / non faites .

Tout d'abord ce qui est terminées nous avons les 2 premières interfaces (le login, changement de véhicules), la base de données(avec les valeurs intégrer) , la connexion de la base de données au java (on peut faire des update, et des Insert)

Ensuite les parties en cours de réalisation, l'interface des frais, nous avons aussi la partie java pour les frais, Select en java afin de vérifier les logins qui seront rentrée dans l'interface graphique, et pour finir le CRUD du Mysql qui est en cours.

Et pour finir les parties non faites , suivre le paiement des fiches de frais , la consultation des fiches, la validation des fiches frais.

- Les apports de chaque membre du groupe doivent être clairement indiqués .

Nous nous sommes diriger vers une démarche ou tout le monde touche à tout (Grâce à ça nous nous sommes rendu compte que certaines personne était plus à l'aise dans certains domaines)

Benjamin : Java début de la liaison de données / BDD Update/Insert/Creat/Delete dans la base de données(CRUD) et vérification de la base si il n'y a pas d'erreur / Diviser les tâches dans l'équipe

Simon : Java (fait la plus grosse partie en Java)

Thibaut : Java liaison de données finalisé, création des frais en Java / BDD création de la base de données .

Lucas : Interface graphique et liaison java .

- la description des outils de gestion de projet que vous avez mis en place pour sa réalisation .

Premièrement nous utilisons nos ordinateurs portables

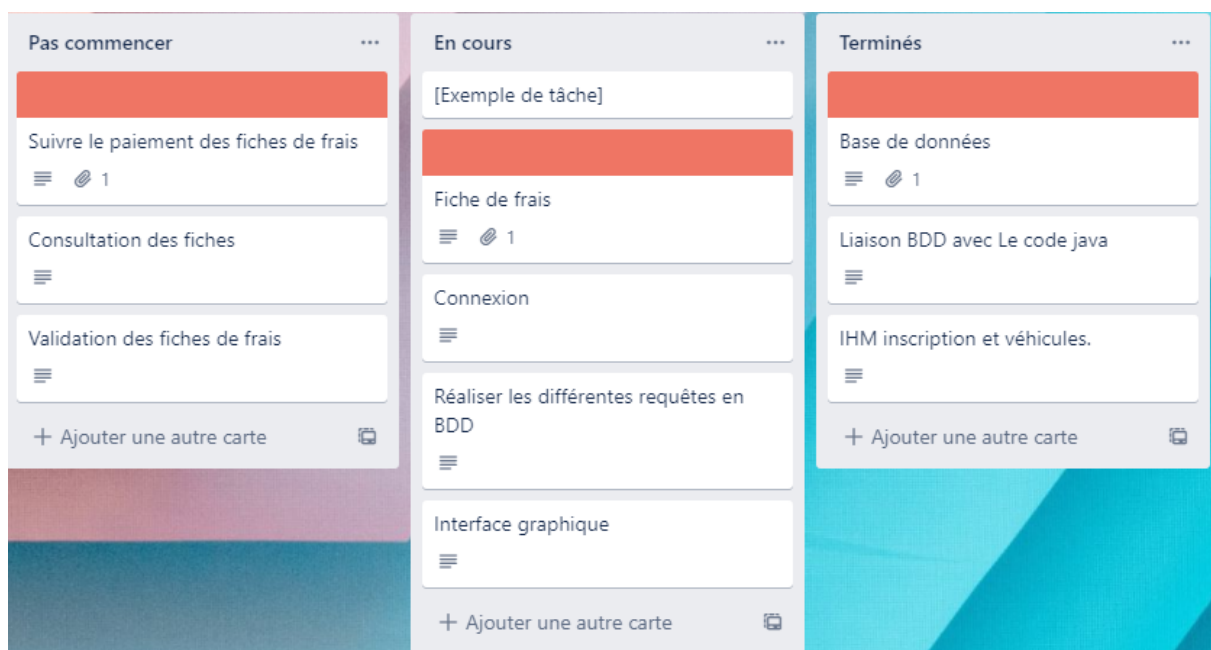
Aux niveaux logiciel nous utilisons scene builder pour la partie graphique, pour le code nous utilisons Visuals studio code

Pour la base de données on a MySQL workbench

Pour communiquer avec l'équipe et partager les différents fichiers, code, travaux nous utilisons discord, mail et github

Pour finir pour le cahier des charges et le brainstorming nous avons utilisé un google sheet, un google doc et finalement nous sommes partie sur trello pour faire tout ceci

- Notamment, un/des *screenshot(s)* de l'outil de suivi d'avancée et d'affectation des tâches (Trello ou autre), qui est censé synthétiser le premier point .



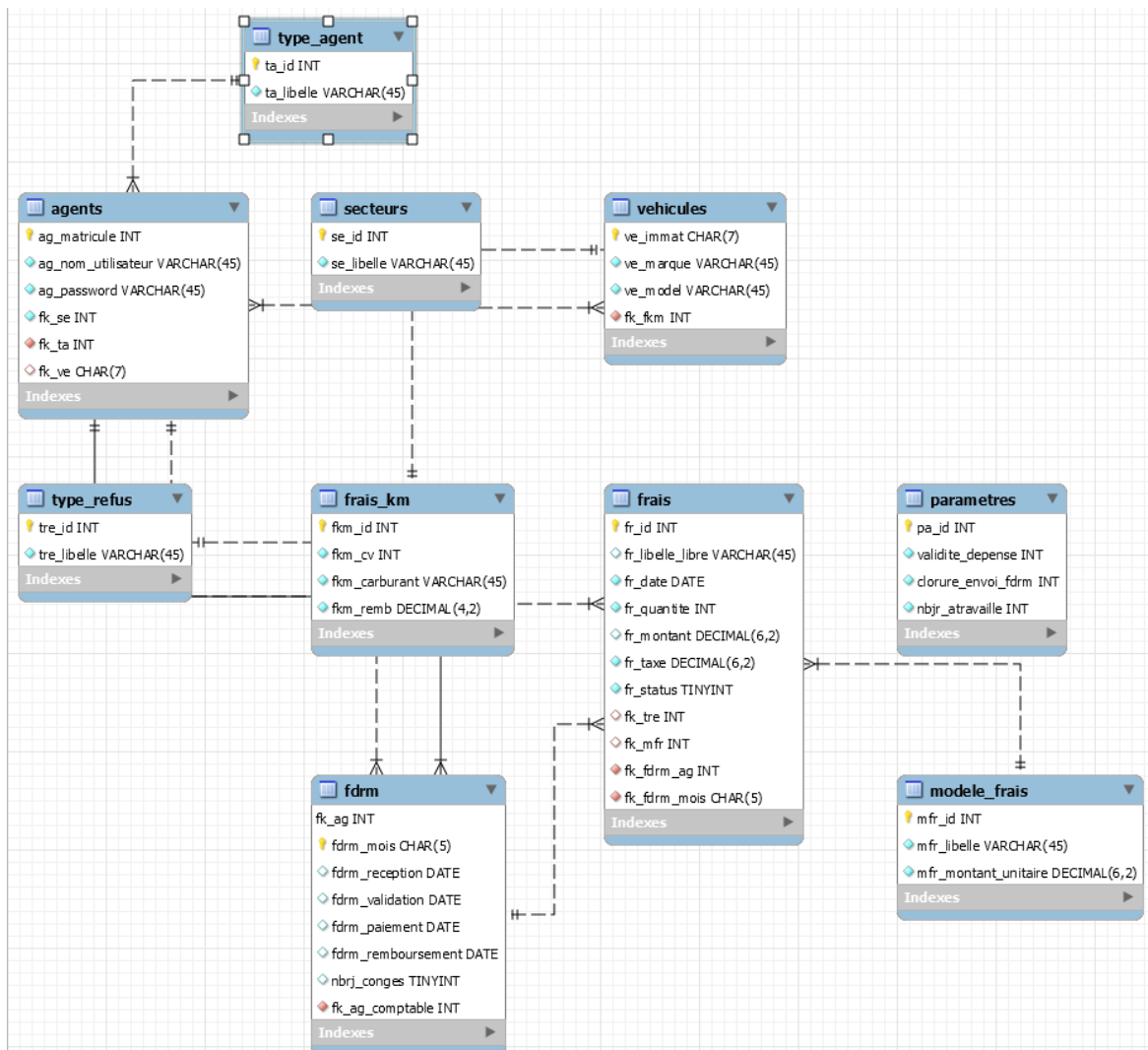


- La description des techniques, méthodes, outils de développement, de conception BDD (analyse, schémas relationnels...), librairies,... utilisés .

- **Bases de données**

Nous utilisons Mysql workbench comme logiciel pour créer cette base de données, Dans cette base nous avons dû la créer au début donc on a pu apprendre quelque manipulation d'une création de base de données

Schéma de la base de donnée :



Après avoir créé cette base nous avons du y ajouter des valeurs pré définies qui seront normalement pas modifier dans l'avenir (mais le seront possible si il y a besoins) bien sûr il y a possibilité de faire des update pour certaines choses

Donc tout d'abord j'ai fais des Insert pour ajouter des valeurs qui doivent être prédéfinis dans les tableaux (les commandes ce sont des exemples car les vraies valeurs du projet je les avais déjà ajouter mais sinon les manipulation sont les mêmes):

alors nous avons les frais km qui vont être pré défini

```
INSERT INTO frais_km (fkm_cv,fkm_carburant,fkm_remb)
VALUES (7,"electrique",0.80)
```

On a pas besoin de mettre le fkm_id car elle est auto incrémenté

Comme nous pouvons voir que sur les screen les valeurs ce sont bien ajouter

fk_id	fk_cv	fk_carburant	fk_remb
1	4	Diesel	0.52
2	5	Diesel	0.58
3	6	Diesel	0.58
4	4	Essence	0.62
5	5	Essence	0.67
6	6	Essence	0.67
7	7	electrique	0.80
NULL	NULL	NULL	NULL

Après nous allons faire ceci pour véhicules, fk_se (secteur), fk_ta (type agent), agents, modele_frais, frais et fdrm

Donc pour véhicules on aura

```
INSERT INTO vehicules (ve_immat,ve_marque,ve_model,fk_fkm)
VALUES ("AAAAAAH","Test8","truc8",3)
```

ve_immat	ve_marque	ve_model	fk_fkm
AAAAAAA	Test1	Truc1	1
AAAAAAB	Test2	Truc2	2
AAAAAAC	Test3	Truc3	3
AAAAAAD	Test4	Truc4	4
AAAAAAE	Test5	Truc5	5
AAAAAAF	Test6	Truc6	6
AAAAAAG	Test7	Truc7	1
AAAAAAH	Test8	truc8	3
NULL	NULL	NULL	NULL

Ensuite pour fk_se (secteur) nous avons

pour ve_immat il faut mettre 7 caractère

```
INSERT INTO secteurs (se_libelle)
VALUES ("COMPTABLE R4")
```

se_id on ne doit pas le mettre car il est auto incrémenté

se_id	se_libelle
1	R1
2	R2
3	R3
4	COMPTABLE R1
5	COMPTABLE R2
6	COMPTABLE R3
7	COMPTABLE R4
NULL	NULL

Après pour fk_ta (type agent) on a

```
INSERT INTO type_agent (ta_libelle)
VALUES ("Test")
```

ta_id est auto incrémenté donc on ne le met pas

	ta_id	ta_libelle
▶	1	Visiteur
	2	Comptable
	3	Test
•	NULL	NULL

Puis nous avons les agents (qui seront prédéfinis bien sûr ne pourront y en rajouter si il y doit y avoir des nous agents)

```
INSERT INTO agents (ag_nom_utilisateur,ag_password,fk_se,fk_ta,fk_ve)
VALUES ("Arthur","password6",6,2,"AAAAAAG")
```

On va y entrer un nom d'utilisateur et un mot de passe après on va entrer des valeurs qu'on a déjà ajoutées dans les tableaux précédents que nous avons complété

Donc on a fk_se (on va utiliser le se_id qui est dans le tableau secteur)

Après nous avons fk_ta(on va utiliser le ta_id qui est dans le tableau type_agent)

Pour finir on a fk_ve (qui va être le matricule de la voiture qui est ve_immat qu'on va retrouver dans le tableau vehicules)

Ensuite nous allons nous occuper des frais qui doivent être ajouter aussi

```
INSERT INTO modele_frais (mfr_id,mfr_libelle,mfr_montant_unitaire)
VALUES (10,"Test", "20.00")
```

	mfr_id	mfr_libelle	mfr_montant_unitaire
▶	1	Repas midi (R1)	23.00
	2	Repas midi (R2)	25.00
	3	Repas midi (R3)	29.00
	4	Nuitée simple (R1)	66.00
	5	Nuitée simple (R2)	71.00
	6	Nuitée simple (R3)	80.00
	7	Repas soir + nuitée (R1)	77.00
	8	Repas soir + nuitée (R2)	82.00
	9	Repas soir + nuitée (R3)	92.00
	10	Test	20.00

Après avoir fais cela on va faire pour fdrm

```
INSERT INTO fdrm (fk_ag,fdrm_mois,fk_ag_comptable)

VALUES (2,"03/20",4)
```

Donc dedans on va y entrer le mois pour fk_ag, après le jour et l'année pour fdrm_mois, et ensuite on va y mettre un comptable

Pour finir on va faire frais

```
INSERT INTO frais
(fr_libelle_libre,fr_date,fr_quantite,fr_montant,fr_taxe,fr_status,fk_fdrm_ag,fk_f
drm_mois)

VALUES ("Test3","2020-03-02",4,"20.00","5.00",0,2,"03/20")
```

Cela va permettre de créer une fiche de frais, donc on va y entrer un libelle une date (qui doit êtres la même que dans la parti fdrm), mettre une quantite, un montant, les taxe, fk_tre et fk_fdrm_ag peut être null donc on met rien dedans, fk_fdrm_ag donc l'agent concerné, ensuite fk_fdrm_mois le mois qu'on a mis dans fdrm

Donc après avoir les Insert Into nous allons passer au Select

Ce Select va servir pour la connexion d'un utilisateur

```
SELECT
ag_nom_utilisateur as 'nom',
ag_password as 'password',
fk_ta as 'type_agents'
FROM ap2_gsb.agents
JOIN type_agent ON fk_ta = ta_id;
```

Ce select va nous servir dans la parti secteur (pour pouvoir faire après un update si il y a un changement de secteur)

```
SELECT
ag_matricule as 'matricule',
ag_nom_utilisateur as 'nom',
se_libelle as 'secteur'
FROM ap2_gsb.agents
JOIN secteurs ON fk_se = se_id;
```

Ce select va nous servir dans la parti véhicules (pour pouvoir faire après un update si il y a un changement de véhicules)

```
SELECT
fkm_id as id ,
```



```
fk_m_carburant as carburant  
FROM ap2_gsb.frais_km;
```

Donc passons aux Update

Comme je l'ai dit juste avant on pourra faire des update

Donc faire un update a secteur et un update véhicules

Donc l'update si changement de secteur (on va y entrer le nouveau secteur "fk_se", et le matricule "ag_matricule" de l'agent qui change de secteur).

```
UPDATE agents  
SET fk_se = 2  
Where ag_matricule = 1
```

Ensuite l'update si changement de véhicules (on va y entrer la nouvelle immatriculation du nouveau véhicule "fk_ve", et le matricule "ag_matricule" de l'agent qui change de véhicules).

```
Update agents  
set fk_ve ="AZE58RT"  
Where ag_matricule = 1 ;
```

- Java :

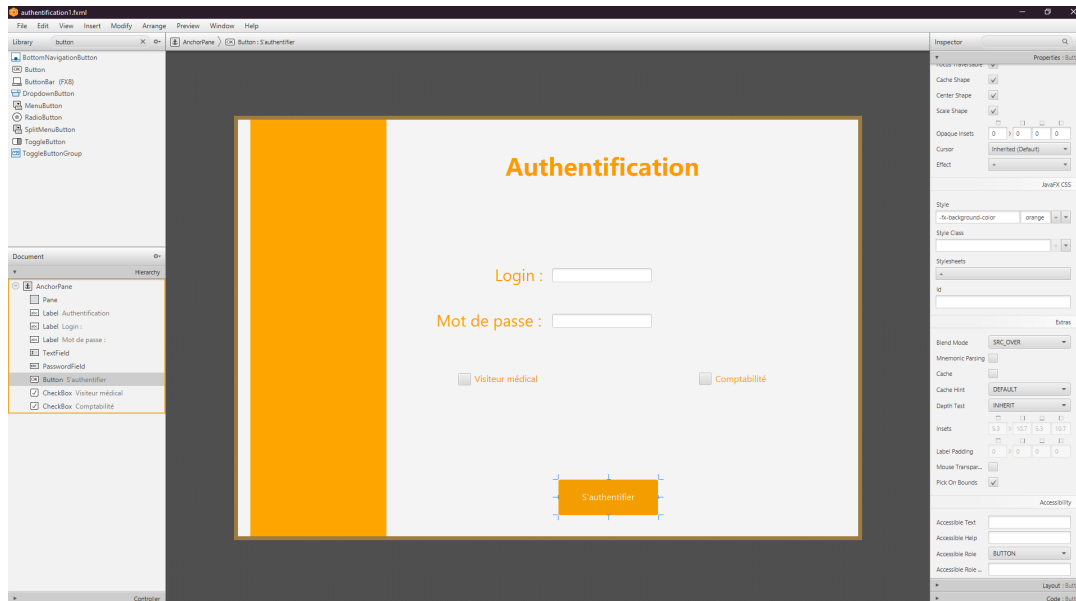
Nous avons réalisé un début de code où l'utilisateur rentre ses frais et ses frais de hors forfait et doit les valider ensuite il a la possibilité de les modifier s'il y a fait des erreurs. A la fin, le programme récapitule la totalité des frais que l'utilisateur vient de rentrer. Pour cela j'ai utilisé la bibliothèque de Java util et de java swing. Pour voir le début du code fonctionnel il est disponible dans le github sous le nom de fiche de frais.

Nous avons aussi créé un code qui permet d'écrire dans la base de donnée comme des "INSERT INTO" "UPDATE" "SELECT" qui permettront ensuite d'ajouter des agents qui seront soit des comptable soit des client, nous avons rajouté des bibliothèque de java qui permet de nous connecter à mysql grâce à java JDBC et maven

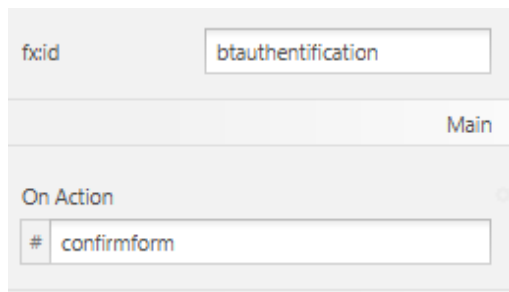
- Partie Graphique (Java Fx):

Logiciel utilisé Scenebuilder et un IDE (Visual Studio Code)

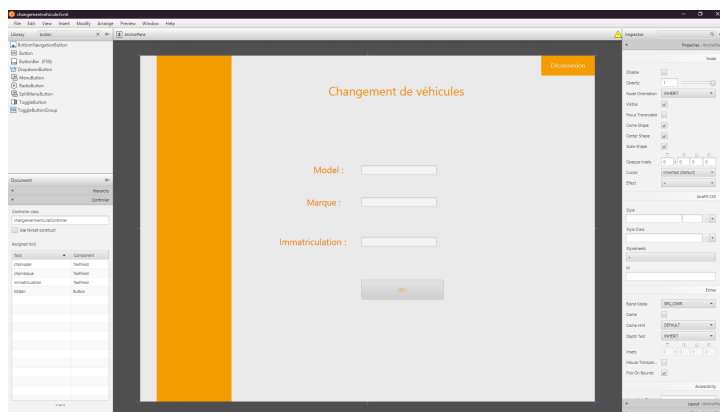
Nous avons choisi une solution avec javafx et le logiciel Scenebuilder après s'être familiariser avec le logiciel la première page a été créée la page d'authentification.



Pour faire fonctionner ce bouton il faut remplir sont fx:id et son action.



La page du changement véhicule est aussi réalisée.



Les controllers servent à réaliser les actions des boutons par exemple et de réaliser des liaisons avec d'autres pages fxml.

```

1 package javafxml;
2 import java.io.IOException;
3 import javafx.event.ActionEvent;
4 import javafx.fxml.FXML;
5 import javafx.scene.control.Button;
6 import javafx.scene.control.CheckBox;
7 import javafx.scene.control.PasswordField;
8 import javafx.scene.control.TextField;
9
10 public class authenticationController {
11
12     @FXML
13     private TextField loginfield;
14
15     @FXML
16     private PasswordField pwdfield;
17
18     @FXML
19     private Button btauthentication;
20
21     @FXML
22     private CheckBox choixmed;
23
24     @FXML
25     void confirmform(ActionEvent event) throws IOException {
26         App.setRoot("changementvehicule");
27     }
28
29     @FXML
30     void takechoixmed(ActionEvent event) {
31     }
32
33     @FXML
34     void takeloginfield(ActionEvent event) {
35     }
36
37     @FXML
38     void takepwdfield(ActionEvent event) {}
39
40 }
41
42
43
44
45
46
47

```

```

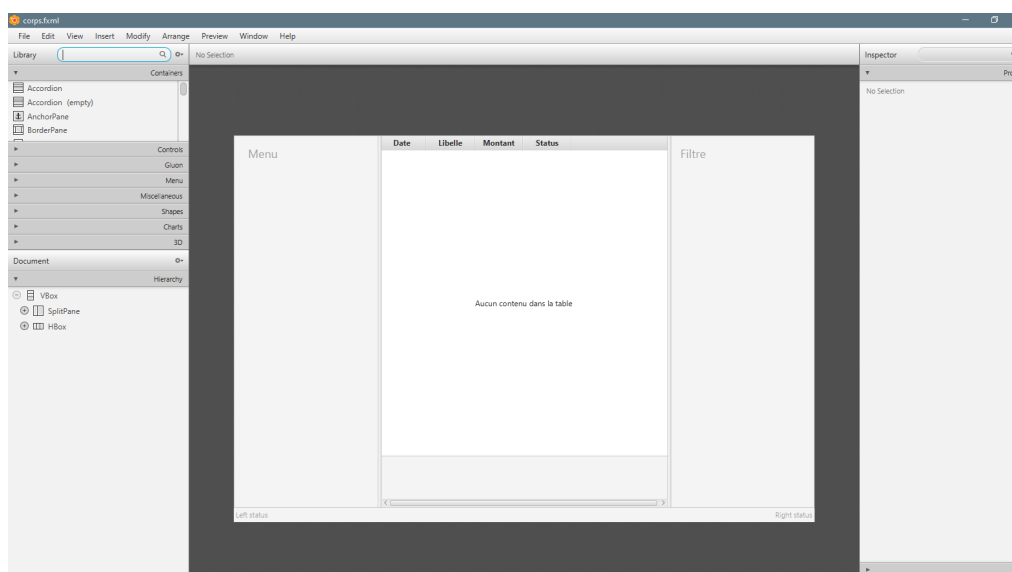
1 package javafxml;
2
3 import javafx.event.ActionEvent;
4 import javafx.fxml.FXML;
5 import javafx.scene.control.Button;
6 import javafx.scene.control.TextField;
7
8 public class changementvehiculeController {
9
10     @FXML
11     private TextField chpmodel;
12
13     @FXML
14     private TextField chpmarque;
15
16     @FXML
17     private TextField immatriculation;
18
19     @FXML
20     private Button GObtn;
21
22     @FXML
23     void chpimmatriculation(ActionEvent event) {
24     }
25
26     @FXML
27     void ectchpmodel(ActionEvent event) {
28     }
29
30     @FXML
31     void envoieformvehic(ActionEvent event) {
32     }
33
34     @FXML
35     void etcchpmarque(ActionEvent event) {
36     }
37
38 }
39
40
41
42
43

```

Ici le bouton s'authentifier amène à changement de véhicules.

App java permet de lancer le javafx et d'afficher les pages voici son code.

Maintenant le formulaire de gestion de frais reste à réaliser mais en bonne voie.



- Une analyse des difficultés rencontrées, comment vous les avez surmontées .

On a eu certains problème au début pour la liaisons de la base de données on a était sur des sites et regarder quelques vidéo qui expliquer comment faire donc on a vite régler ce problème .

On a eu aussi quelques problèmes au niveau de la création de la base donnée avec les clés primaires et les clés étrangères pareil pour ce problème nous avons regardé des vidéos et des sites qui parle de cela.

Problèmes avec le chemin des fichiers dans vs code la page fxml de s'affichait pas car ne trouvait pas le fichier controller de ce fait il suffisait de rentrer le dossier.le fichier controller dans le fxml des pages.