BEN GAIED Hicheme
L3 Informatique
Groupe TP1

Rapport Projet IHM

TABLE DES MATIÈRES

1 - Fonctionnalités **2**

2 - Wireframe **2**
2.1 Wireframe 2
2.2 Organigramme 5

3 - Implémentation **6**
3.1 Fonctionnalités nécessaires 6
3.2 Amélioration Wireframe pendant la phase de développement 6
3.3 Bonus 9
3.4 Couleur 10
3.5 Difficultés rencontrées 11

4- Conclusion **12**

1 - Fonctionnalités

- Une vue générale avec les volumes de produits vendus sur chaque canal. Trois graphiques.
- Pour chaque banque on doit pouvoir visualiser le volume vendu, le chiffre d'affaires généré, et le pourcentage
- Volume, chiffre d'affaires, et pourcentage de chaque banquier.
- Les différents canaux aussi détaillés que possible.

2 - Wireframe

2.1 Wireframe

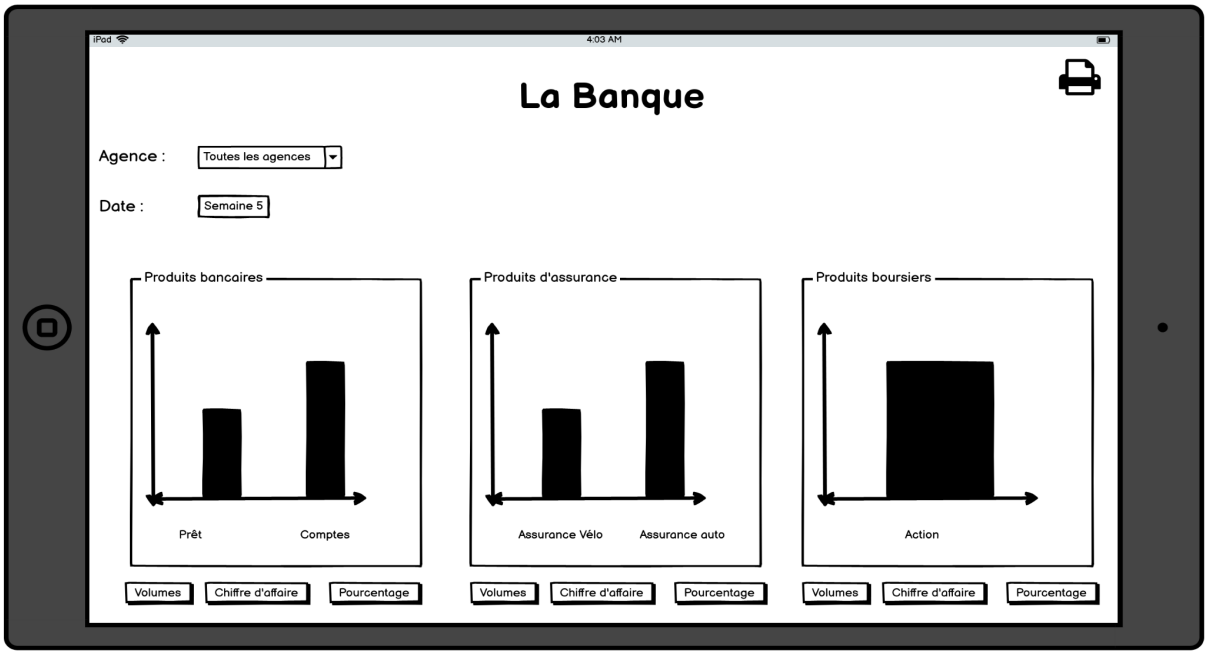


Figure 1.1 Page d'accueil

La page d'accueil est la première chose qu'un utilisateur voit quand il lance l'application bancaire, elle est composée de trois graphiques. Ces trois derniers représentent les différents canaux :

- bancaire
- assurance
- boursier

Chaque canal contient trois boutons :

- Volume : En choisissant cette option, l'utilisateur pourra ainsi voir le nombre de volumes vendu par chaque banquier pour le canal choisi.
- Chiffre d'affaires : Le bouton chiffre d'affaires permet à l'utilisateur de connaître les chiffres d'affaires produits par chaque banquier pour le canal souhaité.
- Pourcentage : Cette dernière option, va permettre à l'utilisateur de connaître l'implication de chaque banquier. Le pourcentage sera calculé en fonction du nombre de produits vendus par les banquiers.

La page principale possède également deux combobox :

- Combobox agence : Celle-ci va permettre à l'utilisateur de choisir l'agence pour laquelle il veut consulter les données.
- Combobox date : Cette dernière permet à l'utilisateur de choisir la semaine de visualisation des données de la banque choisie.

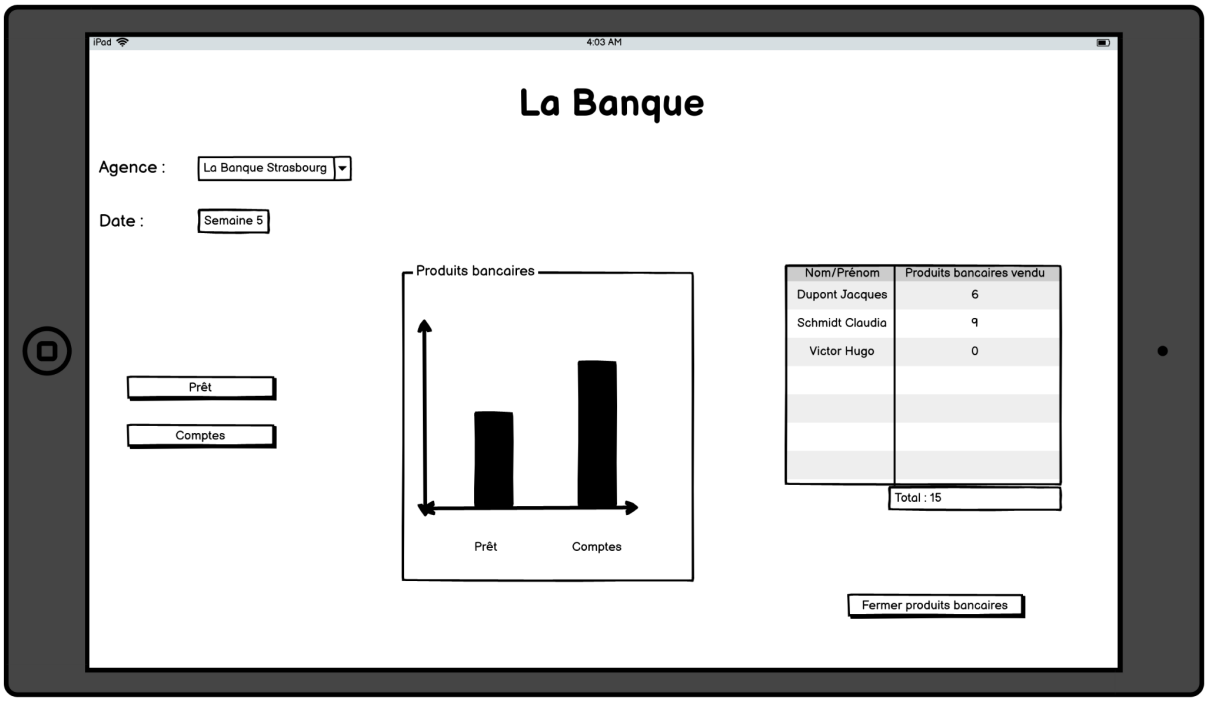


Figure 1.2 Sélection Option volume du canal bancaire.

Cette page est ce qui apparaît lorsqu'on souhaite visualiser le volume du canal bancaire. La configuration de la page aurait été similaire peu importe l'option et le canal choisi, excepté pour le canal boursier, car il ne possède qu'un seul produit à vendre, son action. Cette page contient :

- Un graphique représentant les volumes vendus de chaque produit du canal.

- Un tableau contenant les noms des banquiers, ainsi que le nombre de volumes vendu par banquier.
- Un bouton “Prêt”, et un bouton “Compte” : Ces deux boutons permettent à l’utilisateur d’avoir une vue encore plus détaillée. Chaque bouton mène à une nouvelle page.

- Un bouton “Fermer produit bancaire” : Ce bouton permet de revenir sur la page précédente (figure 1.2).
- Combobox Agence et Date - pour améliorer la navigation (par exemple : pour changer d’agence ou semaine, l’utilisateur ne doit pas revenir en arrière et choisir une autre agence)

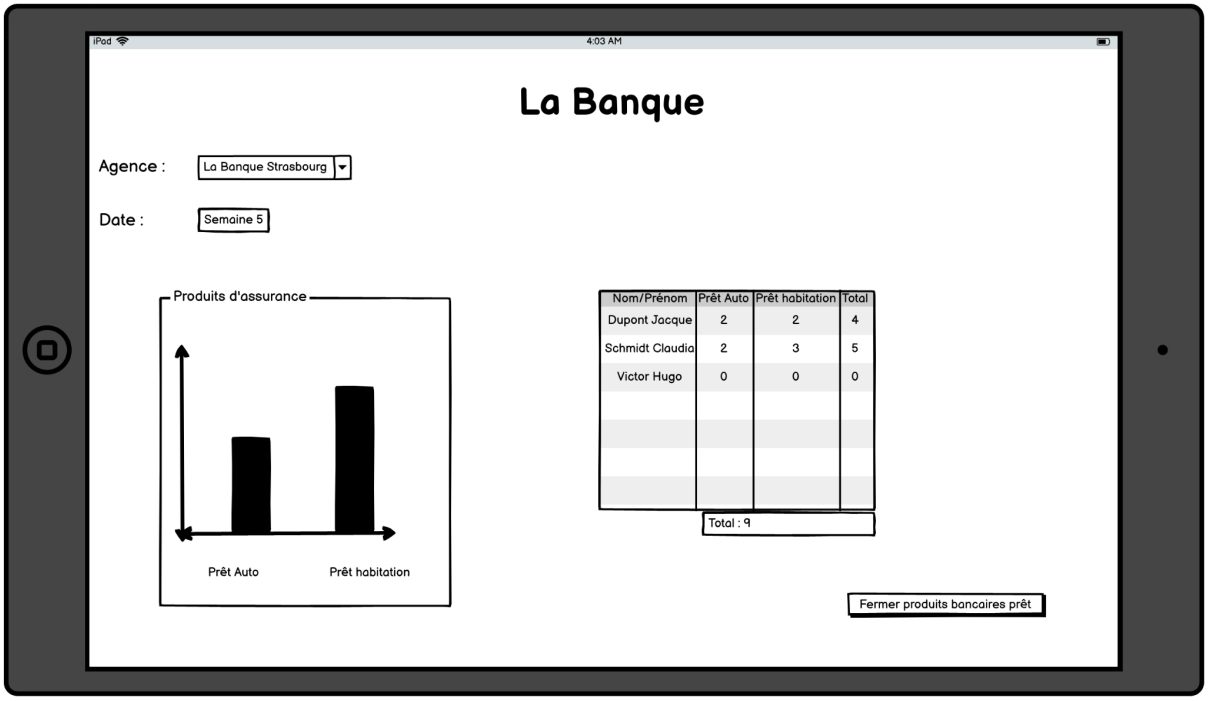


Figure 1.3 Vue détaillée du volume du produit prêt du canal bancaire.

Cette page contient les données les plus détaillées de l’application, l’utilisateur pourra ainsi voir quel volume de produits d’une famille ont été vendus et par quel banquier. Par exemple ici, l’utilisateur pourra voir le nombre de prêts auto et habitation vendus par chaque employé, grâce à un tableau. Cette page contient aussi un bouton de retour “Fermer produits bancaire prêt”, qui permet de revenir à la page précédente (figure 1.2).

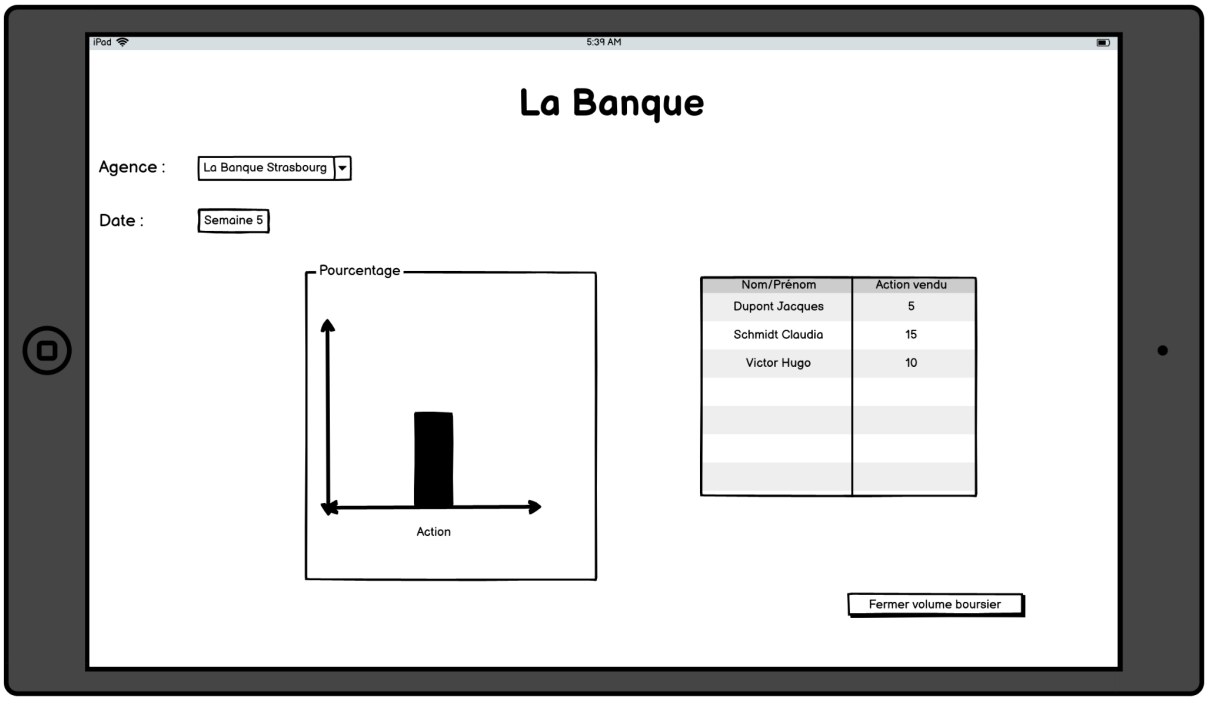
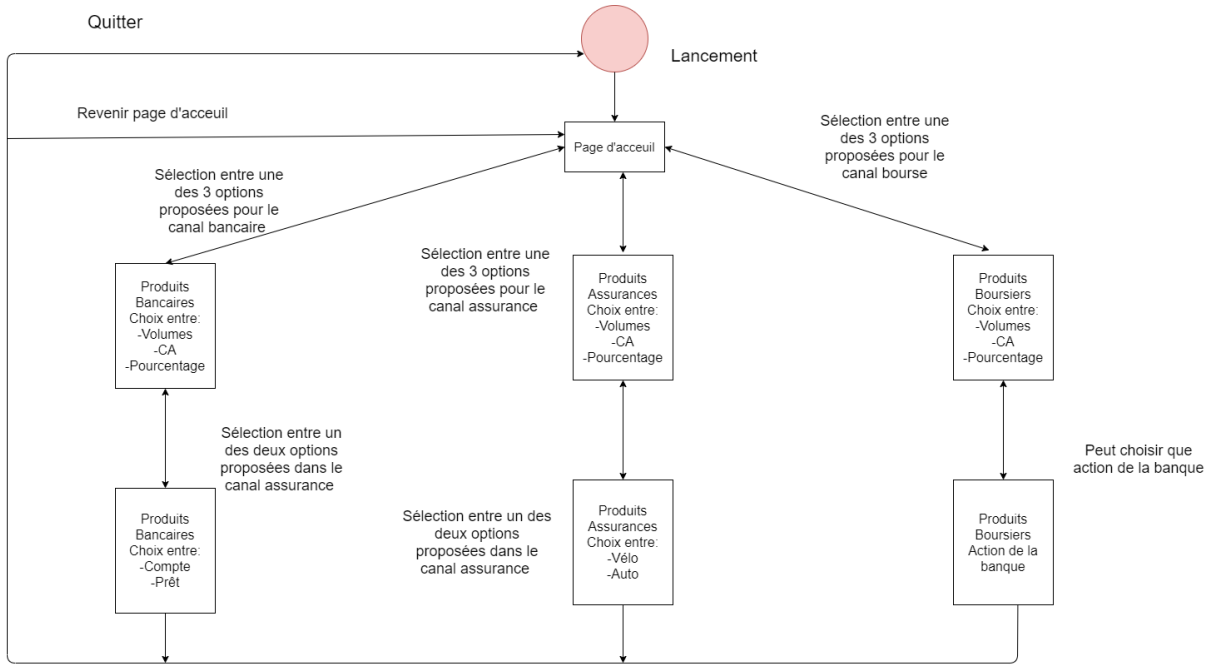


Figure 1.4 Volume Canal boursier

Lorsque l'utilisateur se retrouve sur la page principale (figure 1.1), s'il souhaite accéder au canal boursier, il aura directement une vue détaillée. Le canal bancaire contient qu'un seul produit à vendre, qui est sa propre action.

2.2 Organigramme



Organigramme Figure 1.5

3 - Implémentation

3.1 Fonctionnalités nécessaires

Dans chaque projet, le MOA demande une liste de fonctionnalités nécessaires qui doivent être implémentées, ces dernières sont listées dans la première partie du rapport. Cependant, on a été assez libres de choisir le moyen de créer ces fonctionnalités.

Pour ma part, pour coder les différentes fenêtres de l'application, j'ai opté pour une StackWidget qui est assez pragmatique, facile à comprendre et à utiliser. Dans mon application, ma StackWidget comprend vingt-deux pages répartie de la manière suivante :

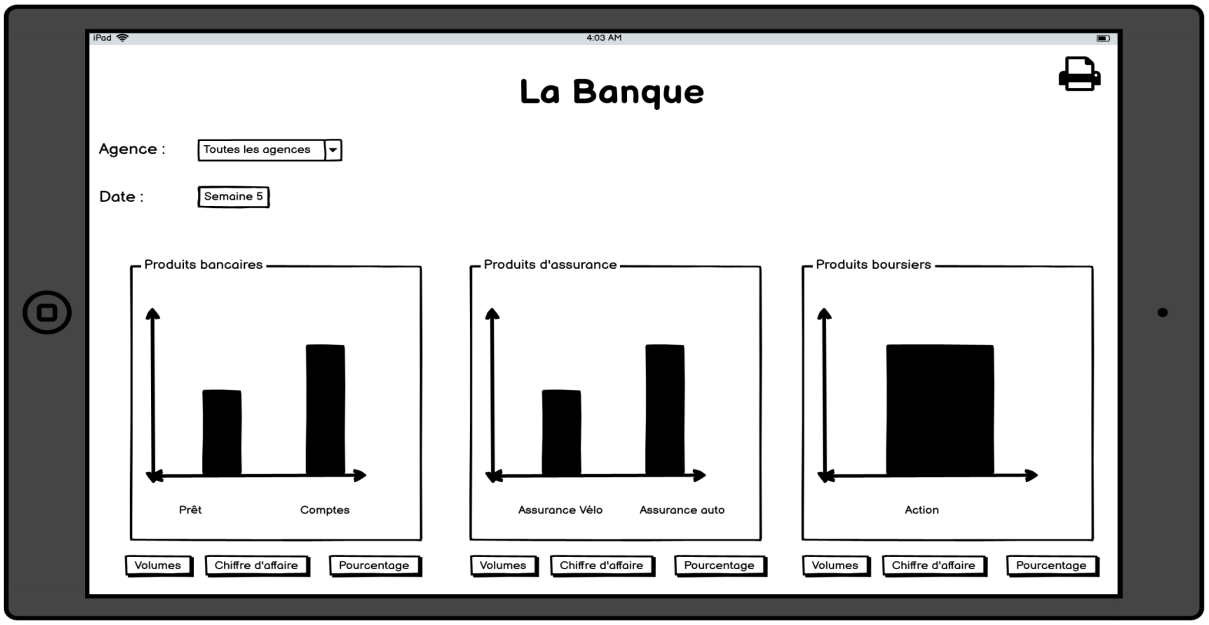
- Une pour la vue d'ensemble
- Neuf pour le canal bancaire
- Neuf pour le canal assurance
- Trois pour le canal boursier

Autant de stackWidget pour les canaux bancaire et assurance me permettent d'avoir des vues détaillées au maximum.

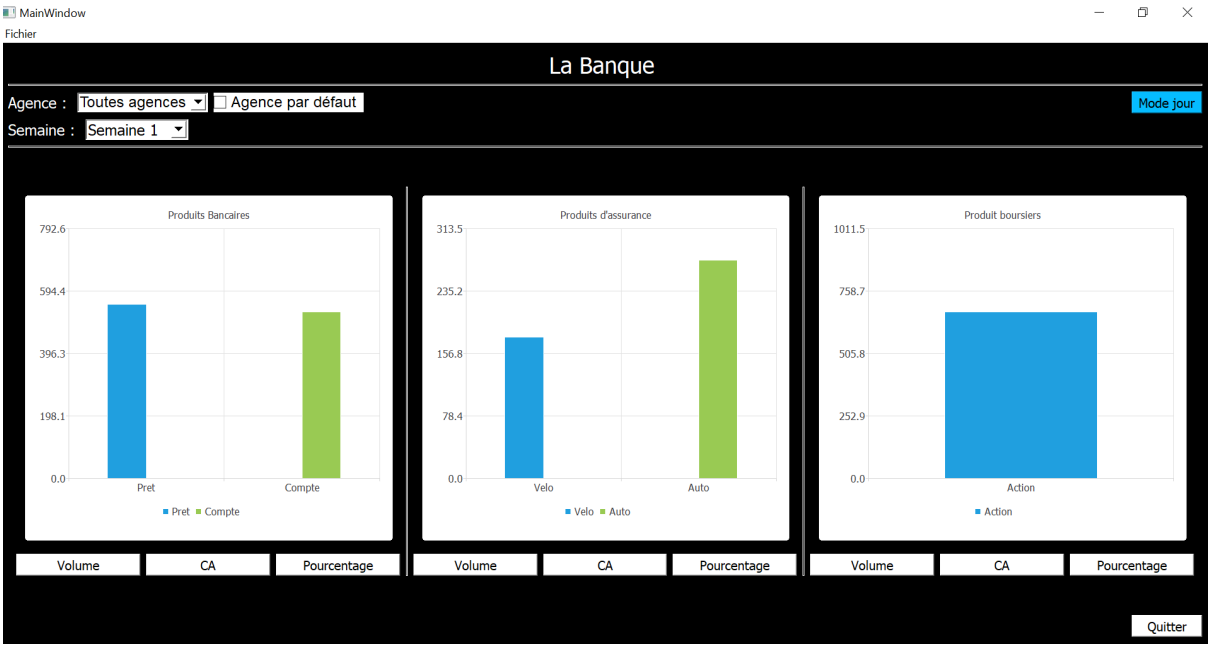
Pour la génération des données, j'ai créé une fonction qui me donne des entiers aléatoires que j'ai préalablement bornée. Ensuite, je remplis mes tableaux des différents canaux avec les données. Je stocke également ces données dans un tableau afin d'avoir un graphique et des pourcentages qui coïncident entre eux. Les données sont aussi chargées aléatoirement à chaque fois que l'on change d'agence ou de semaine. On a ainsi les performances respectives de chaque banquier, les volumes vendus, les chiffres d'affaires réalisés, et son pourcentage d'implication dans le profit de la banque.

3.2 Amélioration Wireframe pendant la phase de développement

Au fur et à mesure que j'avançais dans le développement de l'application, certaines modifications ont eu lieu afin d'améliorer l'IHM et la rendre plus pratique, plus agréable à regarder, ces modifications ne sont pas visibles sur les wireframes préparées avant l'implémentation. Pour ce faire, j'ai fait tester l'application par certains amis. J'ai ensuite pris en compte leur ressenti concernant le visuel, la simplicité d'utilisation, et puis ce qu'ils aimeraient avoir en plus dans cette application.



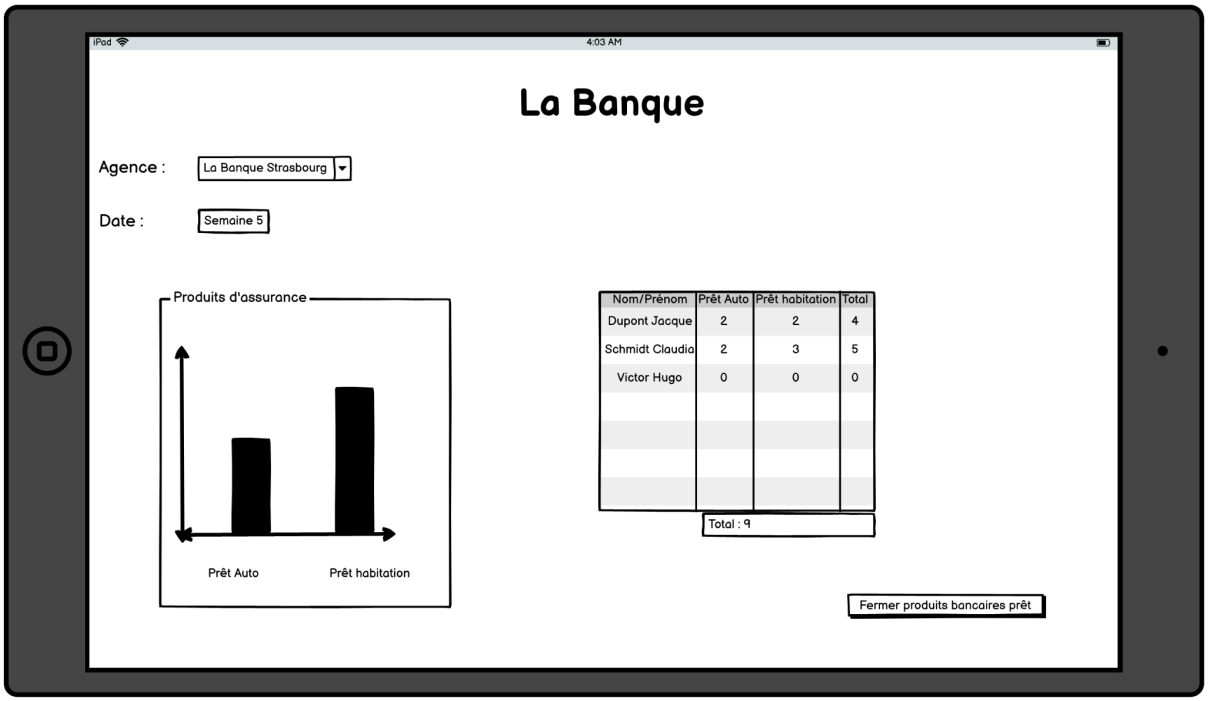
WireFrame Vue d'ensemble Figure 1.6



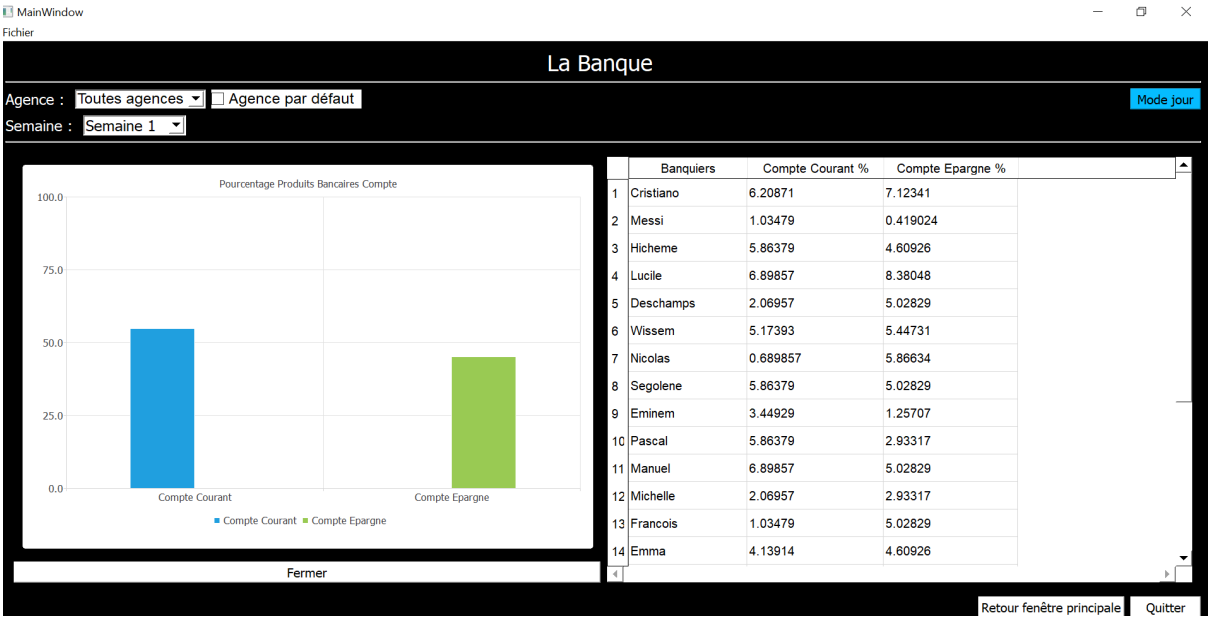
Vue d'ensemble pendant la phase d'implémentation Figure 1.7

On peut jouer au jeu des différences en regardant la Wireframe initiale, et l'évolution de la page pendant l'implémentation du code :

- L'ajout des lignes verticales entre les graphiques a pour but de pouvoir distinguer les trois canaux, ainsi que les boutons qui y sont associés. Quant à l'ajout des lignes horizontales, elles ont pour objectif de faire office d'en-tête, elle contient les deux combobox : la checkbox, et le bouton pour basculer du mode jour au mode nuit et inversement.



Wireframe vue la plus détaillée Figure 1.8



Vue la plus détaillée phase d'implémentation Figure 1.9

On peut voir que dans la figure 1.9, la place à notre disposition est exploitée au maximum alors que sur la wireframe (figure 1.8) j'utilisais à moitié la place à ma disposition. En plus de cela, des boutons ont été implémentés comme "Retour fenêtre principale". Ce bouton nous permet d'éviter d'avoir à refaire tout le chemin (Acceuil->Volume-> Prêt) dans le sens inverse et passer de prêt à l'accueil en un seul clic. Un bouton "Quitter" à également fait son apparition, il va permettre à l'utilisateur de quitter l'application sans avoir à cliquer sur la croix en haut à droite.

3.3 Bonus

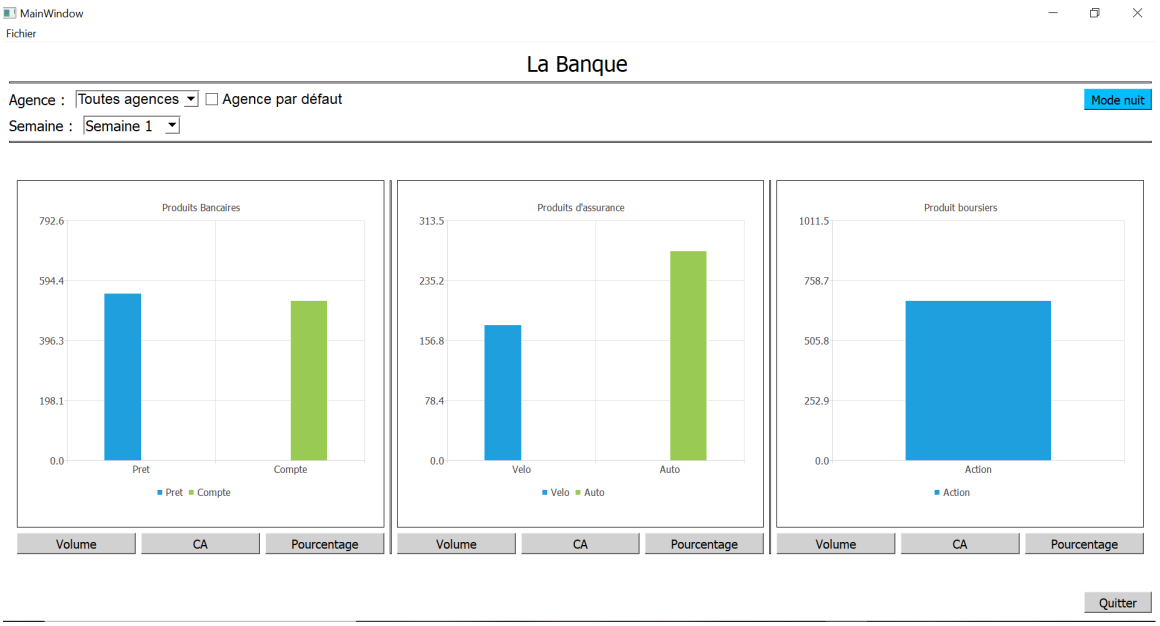


figure 2.0 Switch vers le mode jour

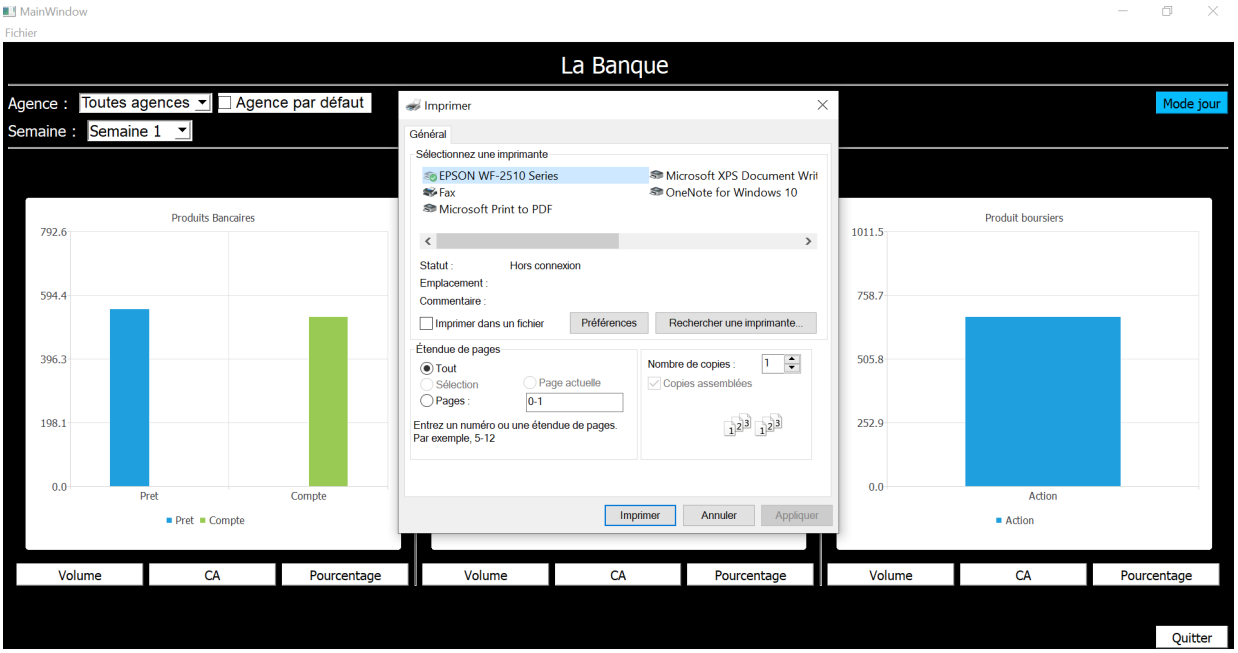


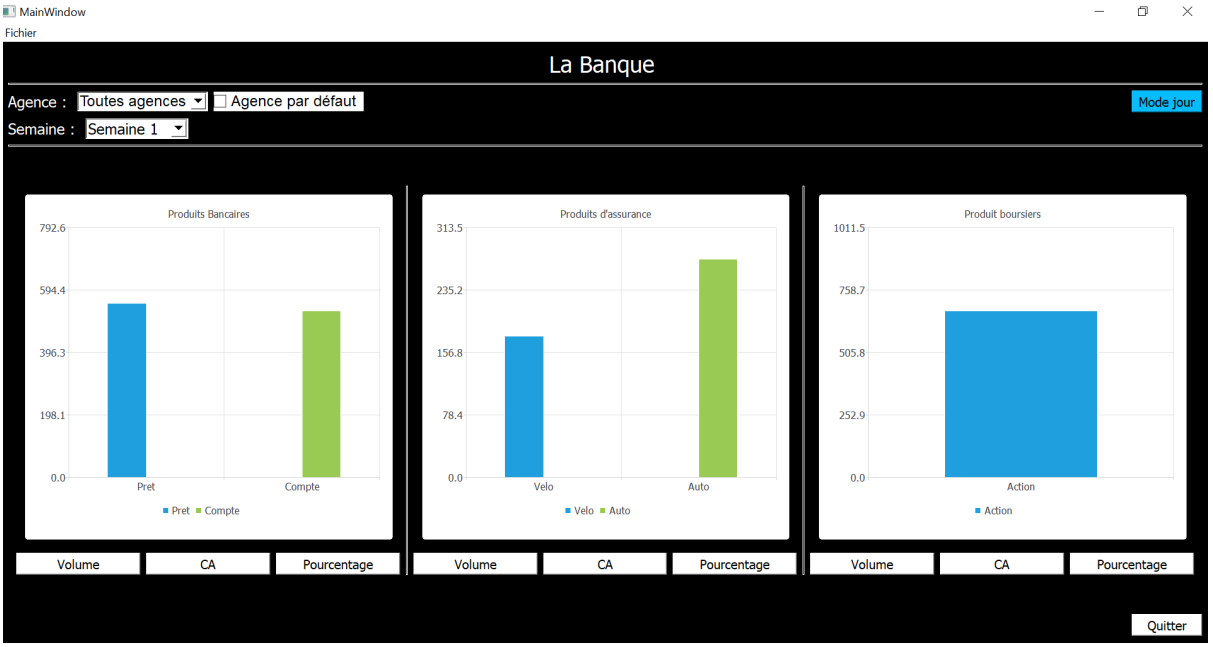
figure 2.1 Imprimer page courante

Également deux options bonus ont été développées :

- L'implémentation d'un bouton "Mode jour" : ce bouton permettra à l'utilisateur de choisir entre un mode jour et un mode nuit en fonction de sa préférence. Lorque ce bouton est cliqué, le fond devient blanc et un changement de couleur des boutons s'opère. Le contenu du bouton "Mode jour" contiendra ainsi le texte "Mode nuit".

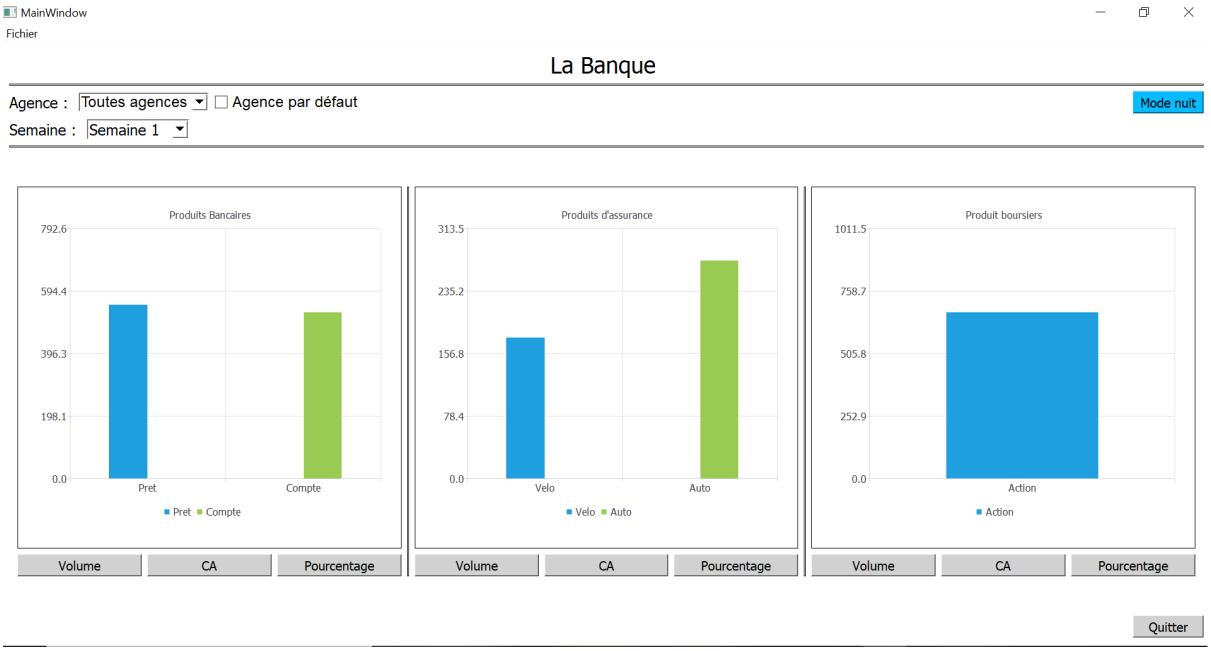
- L’ajout d’une barre de menu : cette dernière contient les options “enregistrer”, “imprimer” et “importer un fichier” pour charger des données. Cependant uniquement l’option “imprimer” est fonctionnelle et permet l’impression de la page courante.

3.4 Couleur



Pour mes choix de couleurs, je suis resté très sobre, comme le blanc se marie bien avec le noir, j’ai décidé de mettre les boutons en blanc, les comboboxs ainsi que les labels.

Pour ce qui est des graphiques, j’ai pris en compte le fait qu’une personne peut être malvoyante. Si une personne ne peut pas distinguer les couleurs, elle pourra distinguer les barres des graphiques avec leur nom respectif, situé en dessous de chaque barre concernée.



Pour les couleurs du mode jour, j'ai opté pour un fond blanc et des boutons en gris clair. Le gris clair va permettre de rendre le texte des boutons plus lisible.

Le bouton "mode nuit" est d'une couleur qui attire un peu l'œil, pour que l'utilisateur prenne conscience du fait qu'il peut basculer vers un mode nuit s'il le souhaite.

3.5 Difficultés rencontrées

Lors de ce projet, j'ai rencontré quelques difficultés, n'étant pas très à l'aise avec les langages objets, j'ai dû travailler énormément le c++ qui était un nouveau langage pour moi avant de me lancer dans le projet.

Hormis cette difficulté, une autre m'a donné beaucoup de mal, celle de comprendre et connaître une bonne partie des librairies offertes par Qt Creator, afin de savoir quelle fonction utiliser, quel Widget est le plus approprié. Pour réussir à surmonter ce problème j'ai dû consulter énormément la documentation de Qt qui est elle bien fournie et surtout explicite. J'ai également regardé beaucoup de tutoriels pour comprendre la bonne manière d'utiliser les graphiques et les tableaux et ainsi insérer les données à l'intérieur.

Au commencement du projet, je n'ai pas utilisé la StackWidget. J'étais parti sur l'idée d'utiliser plusieurs fenêtres qui auraient eu chacune un fichier .cpp, .h, .ui et .pro. Je me suis rendu compte que cela allait être compliqué de devoir gérer autant de fichiers. Je me suis alors renseigné sur internet, et j'ai vu qu'il y avait une manière beaucoup plus simple et qui allait me faciliter la tâche, la StackWidget.

4- Conclusion

Malgré certaines difficultés rencontrées, j'ai apprécié ce projet. Il m'a permis d'apprendre le C++ et de m'améliorer dans ce qui est orienté objet. J'ai aimé le fait qu'on soit libre dans nos choix d'IHM mais qu'on ait également des directives à respecter. Grâce à ce projet, j'ai appris qu'il fallait toujours mettre ces choix en question et ne pas rester figé sur une idée. Je me suis mis à la place des utilisateurs en me posant les questions suivantes :

- "Est-ce facile d'utilisation ?"
- "Est-ce intuitif ?"
- "Est-ce agréable à regarder ?"
- "Quelle fonctionnalité en plus l'utilisateur aimerait avoir ?"

J'ai également fait appel à des utilisateurs pour qu'ils essayent mon application et j'ai fait évoluer mon application en fonction de leur ressenti. J'ai pu ainsi obtenir une meilleure IHM que celle des wireframes de départ.