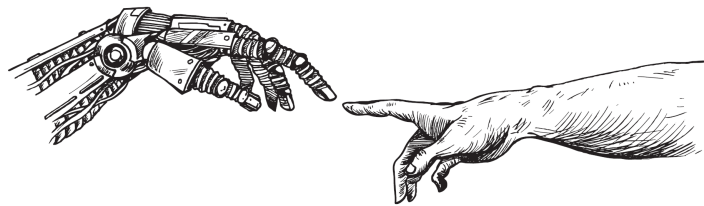


Rapport de Projet  
Interactions Homme-Machine



Marko MARTIC

Avril 2021

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Get Started . . . . .	2
1.2	Propos Introductifs . . . . .	2
1.3	Objectifs . . . . .	2
1.4	Notation . . . . .	3
1.5	Plan . . . . .	3
<b>2</b>	<b>Phase 1 : Compréhension du sujet</b>	<b>4</b>
<b>3</b>	<b>Phase 2 : Wireframes</b>	<b>5</b>
3.1	BONUS : Page de connexion + Préférences utilisateurs . . . . .	6
3.1.1	Wireframe . . . . .	6
3.1.2	Page de connexion . . . . .	7
3.1.3	Préférences utilisateur . . . . .	7
3.2	Tableau de Bord . . . . .	7
3.2.1	Wireframe . . . . .	8
3.2.2	Element Primordial : GroupBox . . . . .	9
3.2.3	Barre de Statut Haute . . . . .	9
3.2.4	Menu vertical droit . . . . .	9
3.2.5	Control Box Options . . . . .	9
3.2.6	Widgets . . . . .	10
3.2.7	A Améliorer (après coup) . . . . .	11
3.3	Stat (Statistiques) . . . . .	11
3.3.1	Wireframe . . . . .	12
3.3.2	Control Box . . . . .	12
3.3.3	Widgets . . . . .	13
3.4	Aide & Réglages . . . . .	13
3.4.1	Wireframes . . . . .	14
<b>4</b>	<b>Phase 3 : Photoshop + Charte Graphique</b>	<b>16</b>
4.1	Logo + Dark Mode . . . . .	16
4.2	Choix des couleurs . . . . .	18
4.3	Couleurs HSL . . . . .	18
4.4	Thème QT . . . . .	18

<b>5</b>	<b>Phase 4 : Programmation</b>	<b>19</b>
5.1	Graphes . . . . .	19
5.2	CheckBoxList . . . . .	19
<b>6</b>	<b>Remarques &amp; Conclusion</b>	<b>20</b>

# Chapitre 1

## Introduction

### 1.1 Get Started

Pré-requis :

QT 5.14.1 (aucune compatibilité n'a été vérifiée avec d'autres versions que celle-ci)

Pour démarrer démarrer rapidement :

1. Télécharger le projet du git : <https://git.unistra.fr/martic/i-hm>
  - (a) seuls trois enseignants ont été ajoutés au git (@scholet, @p.guehl, @rwitz)
2. Ouvrir le fichier sources/LaBanque.pro avec QT 5.14.1
3. Enjoy

### 1.2 Propos Introductifs

Dans le cadre de l'Unité d'Enseignement Interactions Homme-Machine (IHM) en troisième année de Licence d'Informatique à l'Université de Strasbourg, nous devons réaliser une interface utilisateur pour la célèbre entreprise fictive du secteur bancaire, sobrement nommée « La Banque ».

Les employés de La Banque utiliseraient notre logiciel pour avoir un compte rendu de vente trimestriel (pour chaque Comité Directeur), mais également pour voir la performance des différents banquiers.

### 1.3 Objectifs

Le sujet nous donne 3 objectifs principaux :

1. Afficher le volume de produits vendus sur chaque canal
2. Pouvoir sélectionner une ou plusieurs agences et voir ses volumes de ventes par canal
3. Une vue aussi détaillée que possible du canal « Prêt »

Egalement, il est compté sur notre force d'initiative et de prévoyance afin de trouver d'autres fonctionnalités pertinentes qui intéresseront les employés de la Banque, ce seront des fonctionnalités Bonus et devront être précisés comme telles dans le rapport.

Le logiciel doit être développé avec QT 5.14.1 en C++.

## 1.4 Notation

Concernant la notation, c'est plus ou moins équitablement partagé entre ce rapport-ci qui contient entre autre les wireframes, et la réalisation.

De mon côté, je préfère mettre l'accent sur l'IHM/wireframes, même si je n'arrive pas au bout de leur développement back-end, car le but pour moi ici ce serait de progresser en IHM, pas tant en POO.

## 1.5 Plan

Il nous est demandé de détailler toutes les phases de travail que nous avons suivies, elles seront détaillées dans les chapitres suivants.

---

## Chapitre 2

# Phase 1 : Compréhension du sujet

Ici il s'agissait principalement de comprendre le langage métier et s'approprier les problématiques soulevées par le sujet



## Chapitre 3

### Phase 2 : Wireframes

#### 3.1 BONUS : Page de connexion + Préférences utilisateurs

##### 3.1.1 Wireframe

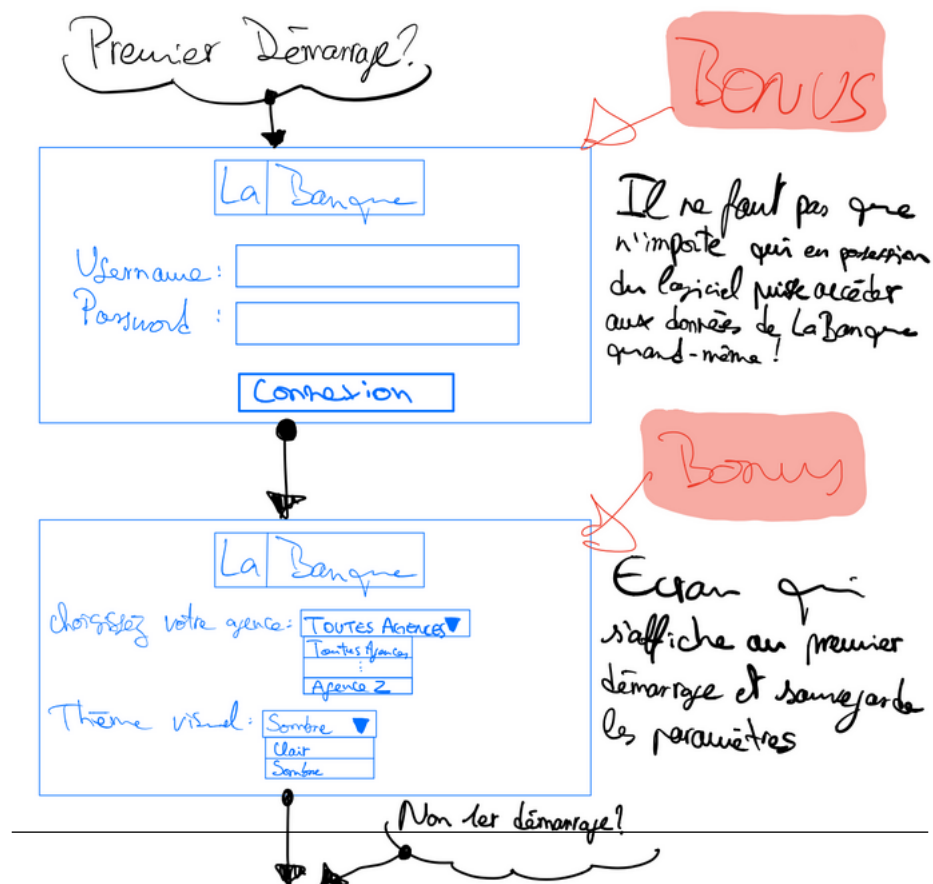


Figure 3.1: wireframe : connexion + préférences



### 3.1.2 Page de connexion

Afin d'éviter que n'importe qui en possession du logiciel puisse accéder aux données de La Banque, une vue pour la connexion est présente.

C'est une vue Bonus.

Une vue assez sommaire, sobre. Un logo y est présent, deux labels, deux textbox et un bouton de connexion.

Notre application est une application de "à risque" car permet d'accéder à des données très confidentielles sur la performance de La Banque. Pour cette raison, cette vue reste très primaire sur ses indications de connexion. Il n'y a aucun bouton pour récupérer son mot de passe, ni de de checkBox « Se Souvenir de moi » pour des raisons de sécurité. Si le combo username/password n'est pas bon on ne donne pas plus d'indications<sup>1</sup>, mais une msgBox ressurgit en disant que quelque chose d'imprévu s'est produit et stipule de contacter l'administrateur réseaux (avec éventuellement un code d'erreur à communiquer? Todo.).

Ici je rogne donc sur la partie Guidage du cours, pour privilégier la sécurité.

### 3.1.3 Préférences utilisateur

Au premier démarrage, il est demandé à l'utilisateur de choisir son agence (sur laquelle l'appli va démarrer ensuite) ainsi que son thème visuel (clair ou sombre). Cette vue n'est ensuite plus affichée, les paramètres sont sauvegardés dans un fichier en dur. Il est possible de revenir sur ces paramètres plus tard.

## 3.2 Tableau de Bord

Le but ici est de faire quelque chose d'efficace, intuitif, visuellement attractif et plaisant à utiliser. Cette vue privilégie la consultation rapide des stats de son agence, et leur impression pour le comité directeur, raison principale qui a poussé La Banque à nous contacter.

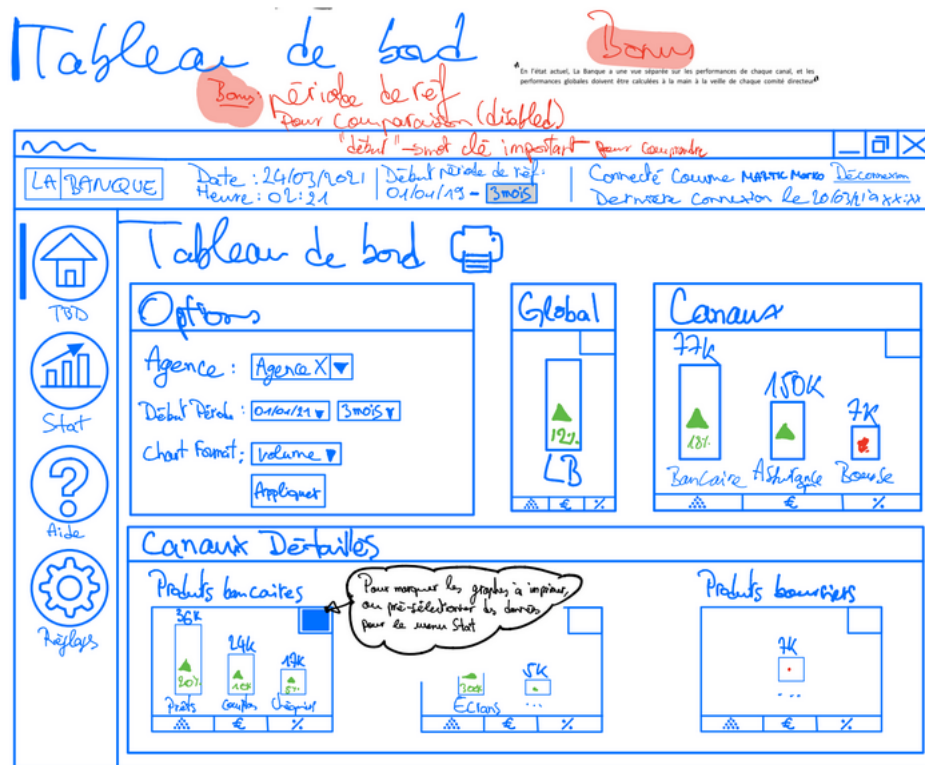
Avec quelques bonus à noter :

- affichage du chiffre d'affaire global et par canal
- affichage pour chaque barre de chaque diagramme à barres le pourcentage d'augmentation ou de baisse par rapport à la période de référence (3 derniers mois par défaut, configurable)

---

<sup>1</sup>contrairement à Facebook par exemple, qui quand vous entrez l'adresse mail et un mauvais mot de passe, va jusqu'à afficher la photo de profil+nom+prénom du compte associé à l'adresse mail.... Très mauvais pour l'anonymat

## 3.2.1 Wireframe



*Bonus*: Comparaison par rapport à la période de référence

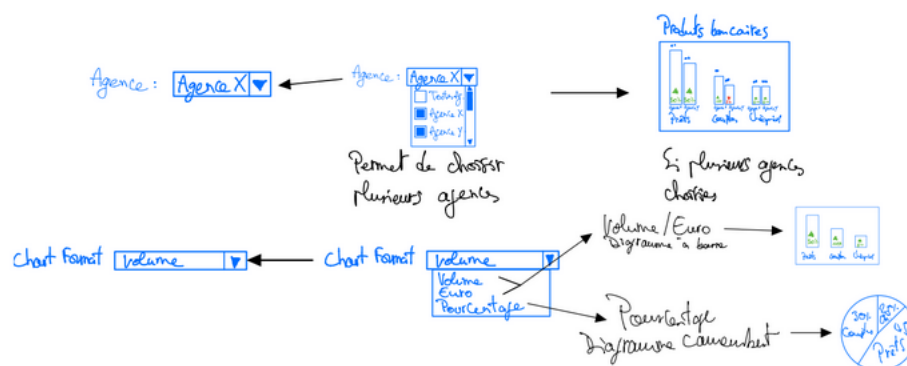


Figure 3.2: wireframe : Tableau De Bord

### 3.2.2 Element Primordial : GroupBox

La GroupBox est pour moi l'élément qui ajoute une grande plus value dans une interface. Elle permet de délimiter une zone, de lui donner une couleur de fond, (un nom, )son contenu est facilement identifiable par l'utilisateur comme faisant partie d'un même ensemble. C'est un élément qui à lui seul peut changer le statut d'une interface « amateur et mal conçue » en « bien faite et agréable à regarder/utiliser ».

### 3.2.3 Barre de Statut Haute

Ici il est question de la barre du haut fixe à travers les menus, qui contient le Logo, la date et heure à la seconde près (mise à jour en temps réel), la période de référence(bonus), son nom d'utilisateur, la date de sa dernière connexion, et le bouton de déconnexion.

Les différents éléments ont différentes utilités :

- La date et heures servent à rassurer l'utilisateur, lui faire comprendre même si aucun graphique ne bouge, l'application est en cours de fonctionnement et sera réactive au besoin.
- La période de référence est la période de comparaison par rapport à la période actuelle choisie dans le widget « Options ».
- Son nom d'utilisateur pour vérifier qu'on est bien sur son compte et sur sa tablette, et non sur la tablette d'un collègue qui est similaire à la nôtre.
- La date/heure de sa dernière connexion pour rassurer l'utilisateur que personne d'autre que lui n'a accès à son compte (ou au contraire l'avertir qu'il y a un souci...)
- Le bouton de déconnexion, qui devra être réactif sur une plus large zone que le simple texte

### 3.2.4 Menu vertical droit

C'est un menu vertical qui permet d'accéder aux différentes vues de l'application.

Un logo représentatif et un texte se situe en dessous de chaque bouton, pour faciliter l'apprentissage et ensuite l'utilisation de l'interface.

Quand on est sur une vue qui correspond au bouton, le bouton en question est plus foncé que les autres pour simuler un « enfoncement ».

### 3.2.5 Control Box Options

C'est ici qu'est contrôlée la vue du Tableau De Bord, on peut y choisir son agence, ou plusieurs agences afin de faire des comparaisons, la période de comparaison, la caractéristique à représenter par tous les graphes (volume/€/%) et un bouton qui permet d'appliquer les options.

Les différents éléments ont différentes utilités :

- Choix de(s) l'agence(s) : Une liste déroulante avec des CheckBox. On aurait pu disposer les checkbox directement dans la group box des Options, cependant comme le but avant tout ici est d'afficher les stats de son agence, il n'est pas judicieux de prendre de l'espace pour rien, il vaut mieux en laisser pour les autres widgets. L'accès aux autres agences est rendu plus difficile (2 clics au lieu d'un) pour laisser de la place pour quelque chose de plus utile.
- La période de comparaison : L'utilisateur doit pouvoir comparer les chiffres de n'importe quelle période avec n'importe quelle période. Au lieu de le laisser choisir deux dates dans des widgets DateEdit, j'ai préféré lui laisser choisir une seule date, et une durée. Ce qui revient exactement au même, mais c'est beaucoup plus pratique à l'utilisation dans notre cas. Ça permet de rapidement switcher entre comparaison de 3 mois, 6 mois ou 1 an. Cette façon de faire n'étant pas commune, le mot « Début » permet normalement à l'utilisateur de comprendre le fonctionnement de cet élément.
- Changer le type de tous les graphes est plus pratique que les faire un par un.
- Le bouton d'application des options : le bouton est par défaut désactivé, dès qu'il y a un changement dans les options il devient fonctionnel. Et son activation le rend désactivé de nouveau. On aurait pu croire qu'il est de trop, et qu'il suffirait de mettre à jour les graphes dès qu'il y a un changement dans les options, mais cela paraîtrait bizarre à l'utilisateur que ses graphes changent à chaque option alors qu'il n'a pas fini de toutes les cocher/activer/désactiver.

### 3.2.6 Widgets

Il y a plusieurs widgets « ensemble de graphes ». Du plus précis au plus global. Cependant ce qu'il y a d'intéressant ce sont les informations complémentaires :

- Pourcentage sur chaque barre affichant l'évolution de la période sélectionnée par rapport à la période de référence avec une couleur rouge ou verte adaptée. Élément que j'estime important, car souvent, les agences ont des objectifs d'augmentation de leur chiffre d'affaires par trimestre.
  - Une CheckBox pour chaque ensemble de barres pour imprimer, ou pour pré-cocher pour la vue Stat, certains éléments.
  - Des boutons pour changer le type de graphique : le bouton « volume » et le bouton « € » sont tous deux checkables et mutuellement exclusifs. Le bouton « des deux autres » permet de représenter le pourcentage de chaque version par rapport au total de la famille avec des diagrammes camembert.
-

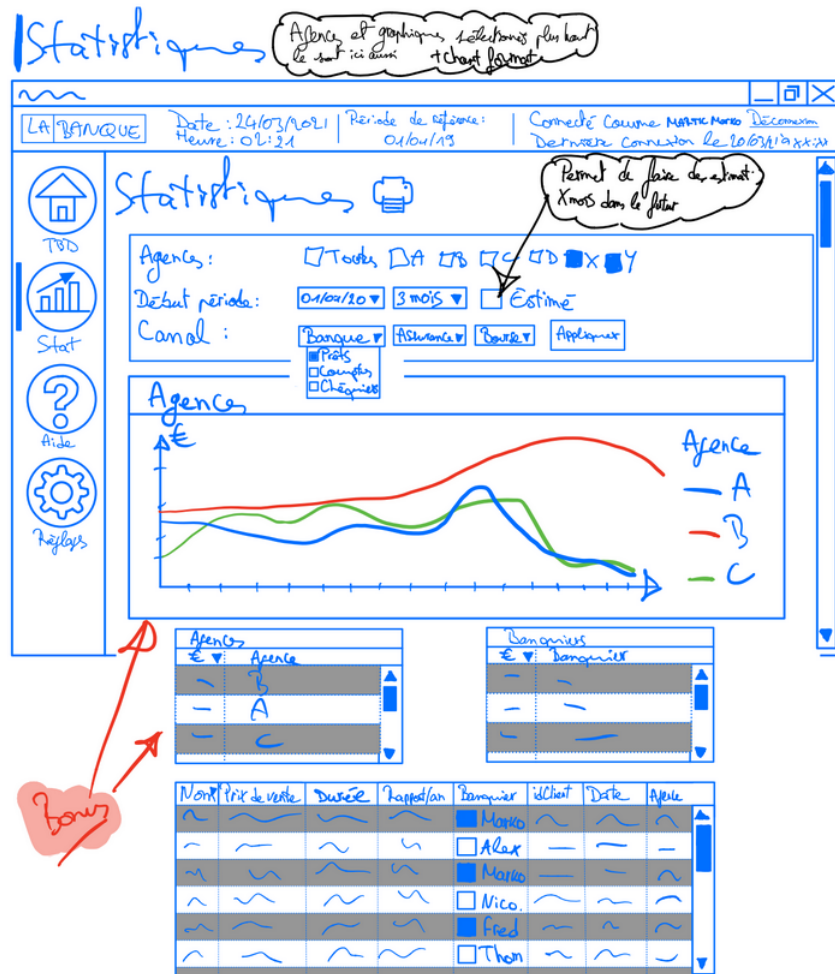
**3.2.7 A Améliorer (après coup)**

On voit que les boutons de choix de type de graphiques sous chaque ensemble de graphes ne sont pas de même taille pour le graphe Global que pour le graphe µProduits bancaires, le design serait un peu à changer en ce sens mais ce n'est pas le plus urgent.

**3.3 Stat (Statistiques)**

Le but ici est de pouvoir comparer toute agence ainsi que tout banquier.

## 3.3.1 Wireframe



L'idée ici est de pouvoir comparer toute agence, ainsi que tout banquier.

Figure 3.3: wireframe : stat

## 3.3.2 Control Box

Ici les agences sont directement disposées en CheckBox, il est aussi possible de faire des projections dans le futur en cochant la case « Estimé », et il est possible de sélectionner tout produit de tout canal.

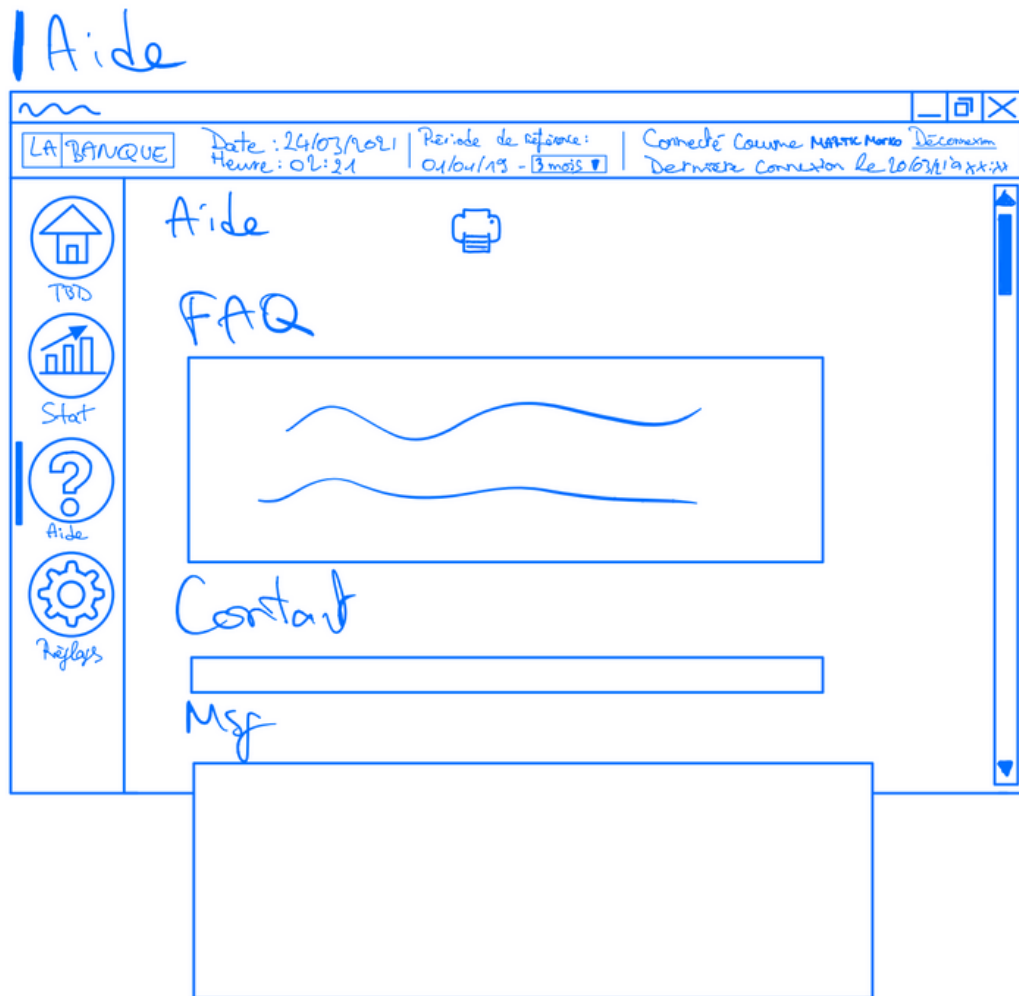
**3.3.3 Widgets**

Il y a un widget listant tous les banquiers des agences sélectionnées ainsi qu'un widget listant les ventes de toutes les banques.

**3.4 Aide & Réglages**

Parties pas vraiment développées, ni réalisées, mais obligatoires pour l'app finale tout de même donc présentes.

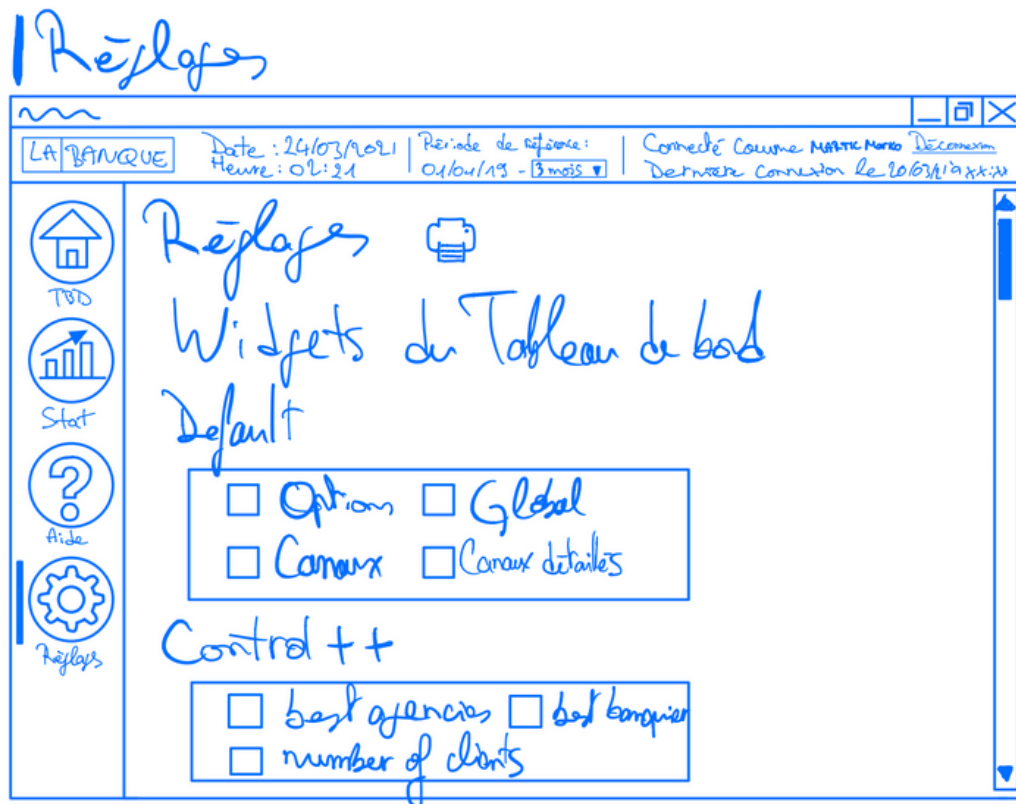
## 3.4.1 Wireframes



ici il s'agit surtout de donner  
la possibilité aux utilisateurs de contacter  
le dev. pour des améliorations/questions.

Figure 3.4: wireframe : aide





Les utilisateurs peuvent vouloir personnaliser leur tableau de bord, pour des usages spécifiques.

Figure 3.5: wireframe : réglages

## Chapter 4

# Phase 3 : Photoshop + Charte Graphique

Ici il était question de trouver une identité graphique à La Banque, notamment :

- Le logo
- Couleurs du darkmode (Les couleurs du mode « light » seront celles par défaut de Qt)

### 4.1 Logo + Dark Mode

La Banque fait penser à l'or, c'est une couleur assez prestige dans l'inconscient collectif donc dans le thème de l'application. J'ai choisi une dominante jaune qui tire vers le marron avec un look un peu prestige.



Figure 4.1: logo La Banque

Grâce au cercle chromatique, j'ai choisi des couleurs complémentaires ainsi que quelques variantes pour faire la palette de couleurs, que voici :

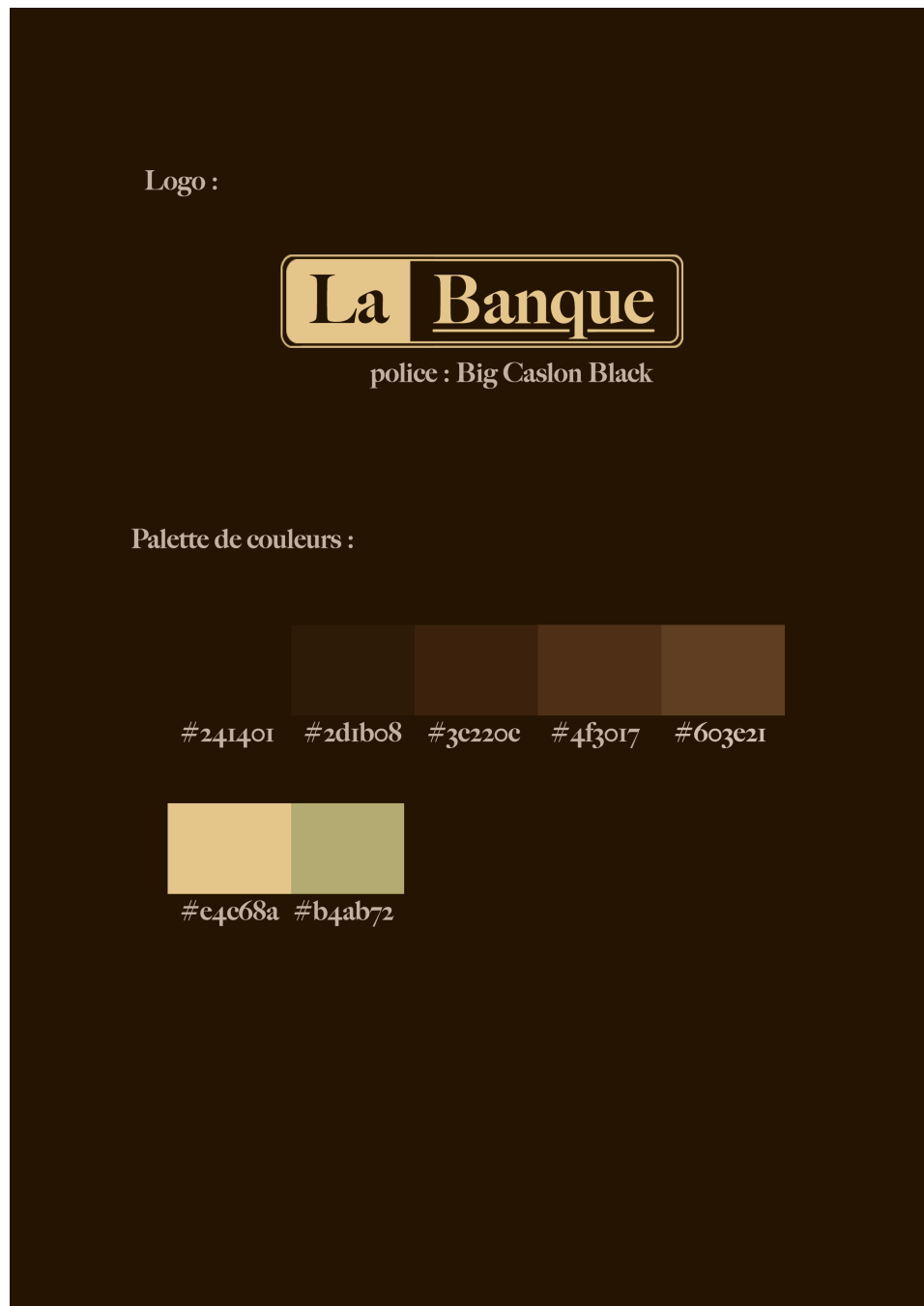


Figure 4.2: charte graphique

## 4.2 Choix des couleurs

De manière générale pour les couleurs du mode sombre, elles vont de la plus sombre à la plus claire, de l'arrière plan au premier plan.

Quant aux couleurs des boutons, ils ont leur couleur normale, qui s'éclaircit au survol (le bouton ressort), et une fois que le bouton est pressé sa couleur devient plus terne (pour simuler un enfoncement).

## 4.3 Couleurs HSL

Pour avoir les variations plus ou moins claires d'une couleur, le plus simple que j'ai trouvé était de prendre la couleur en question en mode HSL (hue - saturation - light), et d'augmenter/diminuer la troisième valeur.

## 4.4 Thème QT

Une feuille de style a été choisie depuis ce site web : <https://qss-stock.devsecstudio.com/templates.php> puis modifiée pour correspondre à la charte graphique.

## Chapter 5

# Phase 4 : Programmation

Cette phase est la moins aboutie, car pas très utile pour l'IHM, néanmoins, il y a un minimum de fait.

### 5.1 Graphes

Pour les graphes j'ai utilisé le package externe « QCustomPlot » car il permet d'exporter les graphes sous forme d'image (sera utile pour imprimer un document), mais aussi d'annoter les barres tel que je les ai dessinées dans les wireframes.

### 5.2 CheckBoxList

Dans la vue du Tableau de Bord, les agences sont cochées dans une liste déroulante. Ce widget n'existe pas tel quel dans Qt, je ne voulais pas changer pour ne pas tomber dans le techno-centrisme, j'ai essayé de l'implémenter sans grand succès, puis j'ai retrouvé un widget déjà réalisé sur github libre de droit au nom de « qcheckboxlist.h ».

## Chapter 6

# Remarques & Conclusion

Remarques :

- Beaucoup de temps a été dépensé sur des détails au niveau du back-end (notamment les graphes et les checkBoxList) pour coller au plus aux wireframes... Je voulais à tout prix éviter le techno-centrisme donc je persévérais jusqu'à trouver QCustomPlot
- J'ai entendu dire par des camarades que la responsivité était importante. Je ne l'ai pas du tout prise en compte, toutes les vues ont été faites pour passer au mieux sur tablette et ordinateurs. Mais évidemment, il aurait fallu faire du responsive, ou carrément repenser l'interface entièrement, car l'utilisation sur les deux appareils est différente.
- Dorénavant, éviter de paramétrer les widgets dans le Qt Creator directement. Il serait préférable de tout mettre dans le code, afin d'avoir une meilleure visibilité sur ce qu'il se passe, ainsi qu'une modification plus aisée.
- Le professeur conseillait une approche itérative et incrémentale, moi je m'approchais plus de l'itératif avec un peu d'incrémental par-ci par-là. Chose qui serait à corriger.

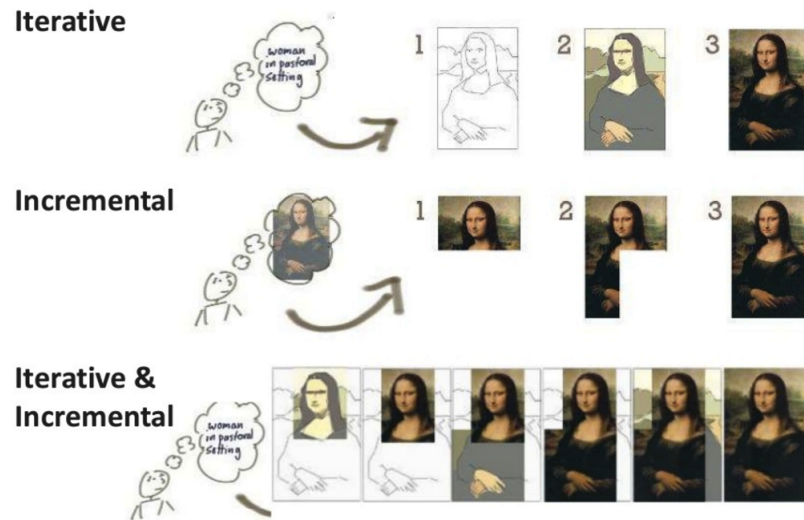


FIGURE 6.1 : itératif/incrémental

Conclusion :

- L'application est loin d'être finie, mais a une base saine et continuable. To be continued...