

Université de Mons  
Faculté Des Sciences

---

## **Projet de Génie Logiciel**

### **S-INFO-015**

---

*Professeur :*

Tom MENS

*Assistants :*

Jeremy DUBRULLE

Gauvain DEVILLEZ

Sébastien BONTE

*Auteurs :*

Augustin HOUBA

Arnaud MOREAU

Cyril MOREAU

François VION



Année académique 2021-2022

# Table des matières

<b>1</b>	<b>Partie commune</b>	<b>3</b>
1.1	Introduction et choix de conception des diagrammes de conception UML . . . . .	4
1.1.1	Use case diagrams . . . . .	4
1.1.2	Interaction overview diagrams . . . . .	5
1.1.3	Class diagrams . . . . .	7
1.1.4	Sequence diagrams . . . . .	8
1.2	Diagrammes de conception UML : application client . . . . .	9
1.2.1	Use case diagram . . . . .	9
1.2.2	Interaction overview diagram . . . . .	10
1.2.3	Class diagram . . . . .	12
1.2.4	Sequence diagram . . . . .	13
1.3	Diagrammes de conception UML : application institution . . . . .	14
1.3.1	Use case diagram . . . . .	14
1.3.2	Interaction overview diagram . . . . .	15
1.3.3	Class diagram . . . . .	17
1.3.4	Sequence diagram . . . . .	18
1.4	Modèle de données . . . . .	19
1.5	Design du REST API . . . . .	21
1.6	Maquette de l'interface utilisateur : application client . . . . .	22
1.7	Maquette de l'interface utilisateur : application institution . . . . .	28
<b>2</b>	<b>Extension A : Gestion de cartes - Augustin HOUBA</b>	<b>33</b>
2.1	Diagrammes de conception UML : application client . . . . .	34
2.1.1	Use case diagram . . . . .	35
2.1.2	Interaction overview diagram . . . . .	36
2.1.3	Class diagram . . . . .	37
2.1.4	Sequence diagram . . . . .	38
2.2	Diagrammes de conception UML : application institution . . . . .	39
2.2.1	Use case diagram . . . . .	39
2.2.2	Interaction overview diagram . . . . .	40
2.2.3	Class diagram . . . . .	41
2.2.4	Sequence diagram . . . . .	42

<b>3</b>	<b>Extension B : Gestion de devises et virements internationaux - Cyril MOREAU</b>	<b>43</b>
3.1	Diagrammes de conception UML : application client . . . . .	44
3.1.1	Use case diagram . . . . .	44
3.1.2	Interaction overview diagram . . . . .	45
3.1.3	Class diagram . . . . .	46
3.1.4	Sequence diagram . . . . .	47
3.2	Diagrammes de conception UML : application institution . . . . .	50
3.2.1	Use case diagram . . . . .	50
3.2.2	Interaction overview diagram . . . . .	51
3.2.3	Class diagram . . . . .	52
3.2.4	Sequence diagram . . . . .	53
3.3	Modèle de données . . . . .	56
3.4	Maquette de l'interface de l'application client . . . . .	57
3.5	Maquette de l'interface de l'application pour une institution . . . . .	58
<b>4</b>	<b>Extension C : Gestion des contrats d'assurance - François VION</b>	<b>59</b>
4.1	Diagrammes de conception UML : application client . . . . .	60
4.1.1	Use case diagram . . . . .	60
4.1.2	Interaction overview diagram . . . . .	61
4.1.3	Class diagram . . . . .	62
4.1.4	Sequence diagram . . . . .	63
4.2	Diagrammes de conception UML : application institution . . . . .	64
4.2.1	Use case diagram . . . . .	64
4.2.2	Interaction overview diagram . . . . .	65
4.2.3	Class diagram . . . . .	66
4.2.4	Sequence diagram . . . . .	67
<b>5</b>	<b>Extension D : Paiements et gestion de fraudes - Arnaud MOREAU</b>	<b>68</b>
5.1	Diagrammes de conception UML : application client . . . . .	69
5.1.1	Use case diagram . . . . .	69
5.1.2	Interaction overview diagram . . . . .	77
5.1.3	Class diagram . . . . .	79
5.1.4	Sequence diagram . . . . .	80
5.2	Diagrammes de conception UML : application institution . . . . .	88
5.2.1	Use case diagram . . . . .	88
5.2.2	Interaction overview diagram . . . . .	90
5.2.3	Class diagram . . . . .	91
5.2.4	Sequence diagram . . . . .	92
5.3	Modèle de données . . . . .	93
5.4	Maquette de l'interface utilisateur . . . . .	94
5.5	Maquette de l'interface institution . . . . .	96

# **Chapitre 1**

## **Partie commune**

## **1.1 Introduction et choix de conception des diagrammes de conception UML**

### **1.1.1 Use case diagrams**

### 1.1.2 Interaction overview diagrams

En se basant sur le *use case diagram*, il est possible d'établir un *interaction overview diagram*, décrivant plutôt la structure et le *flow* général d'exécution du programme. Ce diagramme ne va pas dans les détails de chaque cas d'utilisation (cette tâche est réservée à chacun des *sequence diagrams* associés à chaque *use case*) mais décrit plutôt le chemin que l'utilisateur peut suivre durant l'exécution de l'application, en faisant usage des fonctionnalités mises à disposition. Par exemple, au démarrage de l'application, l'utilisateur peut se connecter, afficher la liste de ses portefeuilles, afficher l'historique d'un compte et l'exporter au format JSON, puis sélectionner un nouveau portefeuille et le désactiver, et enfin se déconnecter de l'application avant de fermer cette-dernière. Il est possible de suivre ce chemin d'exécution dans l'*interaction overview diagram*, et si l'on souhaite obtenir plus de détails sur le fonctionnement de l'exportation de l'historique, on peut suivre la référence au *sequence diagram* associé.

Deux *interaction overview diagrams* ont été réalisés. Le premier décrit l'application client, tandis que le second décrit l'application institution financière. Les deux applications sont chacune divisées en deux parties, à savoir :

1. Application client
  - (a) L'authentification ;
  - (b) L'accès à l'application en tant que client.
2. Application institution
  - (a) L'authentification ;
  - (b) L'accès à l'application en tant qu'institution.

Comme on peut le voir, chacune des applications utilise le système d'authentification avec les fonctionnalités de connexion, création de compte, etc. Ensuite, chaque application aura des fonctionnalités bien distinctes.

*Évidemment, si l'on crée un compte client, il ne faudra pas entrer les mêmes informations que pour un compte institution, mais le flux d'exécution reste le même.*

**Authentification** Cette première partie est commune aux deux applications. Elle est accessible à tous, même aux personnes qui ne sont pas enregistrées dans la base de données ou pour qui l'application n'est pas destinée. Cette partie ne communique avec la base de données que lors de l'inscription, afin de vérifier si, par exemple, l'adresse email n'est pas déjà enregistrée dans cette-dernière, ou afin d'enregistrer un nouvel utilisateur. La seule table à laquelle l'application accèdera durant l'exécution de cette partie est la table *Users*, ce qui permet d'empêcher d'accéder aux tables contenant des informations sensibles sans être connecté à son compte.

Lors de l'ouverture de l'application, si aucune connexion n'a été enregistrée (c'est-à-dire si on accède à l'application sans s'être précédemment connecté), l'utilisateur arrive sur la fenêtre d'authentification. Depuis celle-ci, il peut effectuer plusieurs actions :

1. Changer la langue ;
2. Se connecter ;
3. Créer un compte.

Quitter l'application est considéré comme un comportement trivial.

**Choix de conception** Les deux diagrammes se basent sur deux points centraux, représentés par deux grands diamants : *Authentication* et *Accès à l'application en tant que client/institution*. L'*accès à l'application en tant que client/institution* est ce qui différencie le diagramme de l'application client de celui de l'application institution. Chaque application se base sur le point central qui lui correspond, à partir duquel l'utilisateur peut prendre toutes les décisions qu'il désire afin d'utiliser à sa guise les différentes fonctionnalités mises à sa disposition. Cette structure a été choisie puisqu'elle concorde avec la structure qui sera implémentée graphiquement : un menu principal à partir duquel l'utilisateur peut accéder à différentes catégories, fonctionnalités... Typiquement, une flèche de *Control flow* quittant le diamant *Accès à l'application en tant que client* correspondra au clic d'un des boutons du menu principal (voir *User Interface diagrams*), mis à part la fermeture de l'application. Par exemple, pour l'application client, *Afficher les notifications* sera atteint si l'utilisateur décide d'afficher ses notifications via le bouton disponible sur l'écran principal de l'application. Un autre exemple pour l'application institution serait *Accéder à la liste des demandes*. Ce point du flot d'exécution serait atteint si l'utilisateur cliquait sur le bouton lui permettant d'accéder aux demandes faites au près de l'institution.

### **1.1.3 Class diagrams**



#### **1.1.4 Sequence diagrams**

## 1.2 Diagrammes de conception UML : application client

### 1.2.1 Use case diagram

L'énoncé étant globalement assez clair sur les fonctionnalités demandées, très peu de décisions importantes ont dû être prises pour les diagrammes de cas d'utilisation. Généralement tout faisait partie des concepts basiques d'une application (Ex : Réinitialisation du mot de passe) ou alors était clairement spécifié dans l'énoncé. Le use case *Introduire une demande* est une généralisation de toutes les demandes qu'un utilisateur peut envoyer (Demande d'accès à ses comptes, demande de virements,...). La fonctionnalité n'a pas été plus détaillée dans ce diagramme pour plus de clarté. Les descriptions semi-structurées de chaque cas d'utilisations de ce diagramme sont dans un document PDF séparé afin de garantir une certaine clarté.

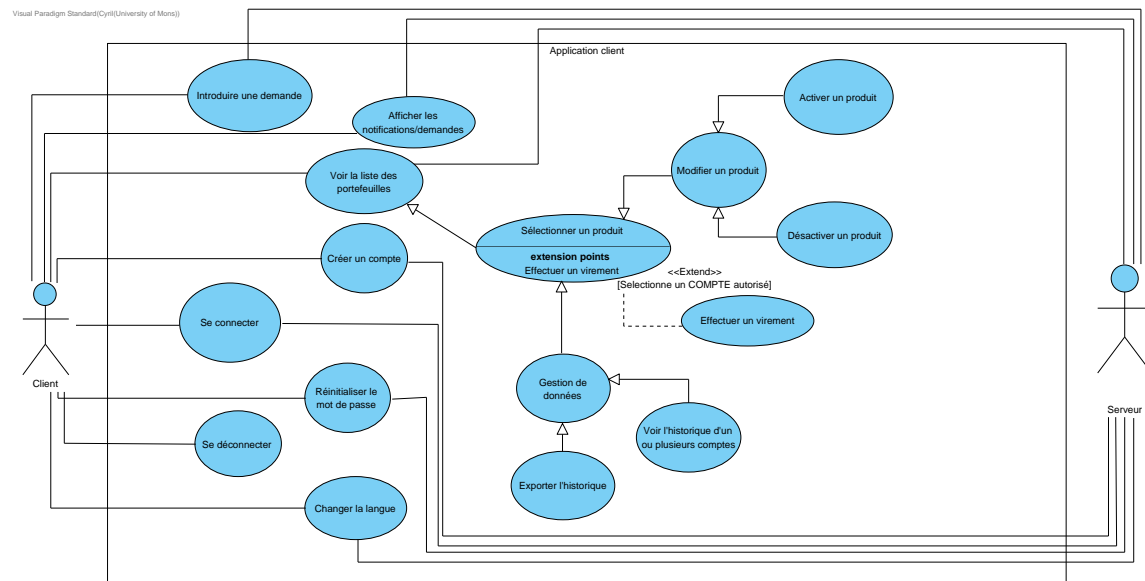


FIGURE 1.1 – Diagramme de case d'utilisation de l'application client.

## 1.2.2 Interaction overview diagram

**Changer la langue** L'utilisateur peut accéder à une interface lui permettant de changer sa langue préférée. Cela mettra à jour sa langue préférée dans la base de données et adaptera la langue de l'interface. Un menu déroulant affichera chaque fichier situé dans le répertoire dédié dont le nom respecte la forme *LL.json*, à savoir deux lettres déterminant la langue (par exemple *fr, en, de, nl, es...*) et au format JSON.

**Se connecter** L'utilisateur qui désire se connecter doit posséder un compte (pré-condition triviale). En entrant son nom d'utilisateur ou adresse email ainsi que son mot de passe, il peut s'identifier et accéder à son compte et ses portefeuilles.

**Créer un compte** L'utilisateur qui ne possède pas de compte et souhaite en créer un doit passer par l'interface de création de compte. Plusieurs informations sont requises : un nom, un prénom, un numéro national au format correct, une adresse email non utilisée et au format correct, un nom d'utilisateur libre, un mot de passe et sa confirmation, ainsi qu'une langue favorite. En confirmant qu'il a enregistré son mot de passe dans une base de donnée locale et sécurisée (comme un gestionnaire de mot de passe), nous encourageons l'utilisateur à protéger son mot de passe. Si toutes les informations sont remplies correctement, il pourra créer un compte et l'utiliser pour se connecter à l'application.

**Accès à application client, en tant que client** La seconde partie n'est accessible qu'en se connectant avec les identifiants d'un compte enregistré dans la base de données. Lors de la connexion, l'utilisateur arrive sur l'écran principal, et peut, depuis celui-ci, accéder à toutes les fonctionnalités. Après s'être connecté, le client peut effectuer plusieurs actions :

1. Accéder à la liste de ses produits financiers
2. Introduire une demande

*Quitter l'application, Changer la langue et Se déconnecter sont considéré comme des comportements triviaux.*

**Accéder à la liste de ses produits financiers** La liste des produits financiers de l'utilisateur s'affiche à l'écran, et les fonctionnalités sont accessibles si l'utilisateur sélectionne un produit.

**Sélectionner un produit** Lorsque l'utilisateur a sélectionné un produit, de nouvelles fonctionnalités deviennent utilisables :

1. Modifier un portefeuille ;
2. Effectuer un virement ;
3. Voir l'historique d'un compte.

**Modifier un portefeuille** La modification du portefeuille permet à l'utilisateur de basculer (*toggle*) l'état du portefeuille sélectionner. Si le portefeuille est activé, l'utilisateur pourra le désactiver, et si le portefeuille est désactivé, l'utilisateur pourra le ré-activer. Après avoir basculé l'état du portefeuille, l'utilisateur est envoyé sur l'écran précédent, à savoir la sélection d'un portefeuille.

**Effectuer un virement** Si l'utilisateur souhaite effectuer un virement, il en a la possibilité en accédant à cette étape du flot d'exécution. Il faut entrer un montant et l'IBAN du destinataire ; le destinataire, le message et la date sont optionnels.



### **1.2.3 Class diagram**

#### **1.2.4 Sequence diagram**

## 1.3 Diagrammes de conception UML : application institution

### 1.3.1 Use case diagram

Contrairement à l'application présentée précédemment, cette application est utilisée par une institution financière et, comme pour l'application 1, très peu de décision ont été prise car tout était assez clair dans l'énoncé. Notons cependant que le cas d'utilisation *Gérer les demandes* permet à l'utilisateur d'approuver ou de refuser une demande mais il devra effectuer les demandes manuellement (par exemple si la demande est d'ajouter un produit financier, il devra l'ajouter manuellement et ensuite accepter la demande.)

La description semi-détaillée de chaque cas d'utilisation de ce diagramme est également dans un PDF attaché à ce rapport.

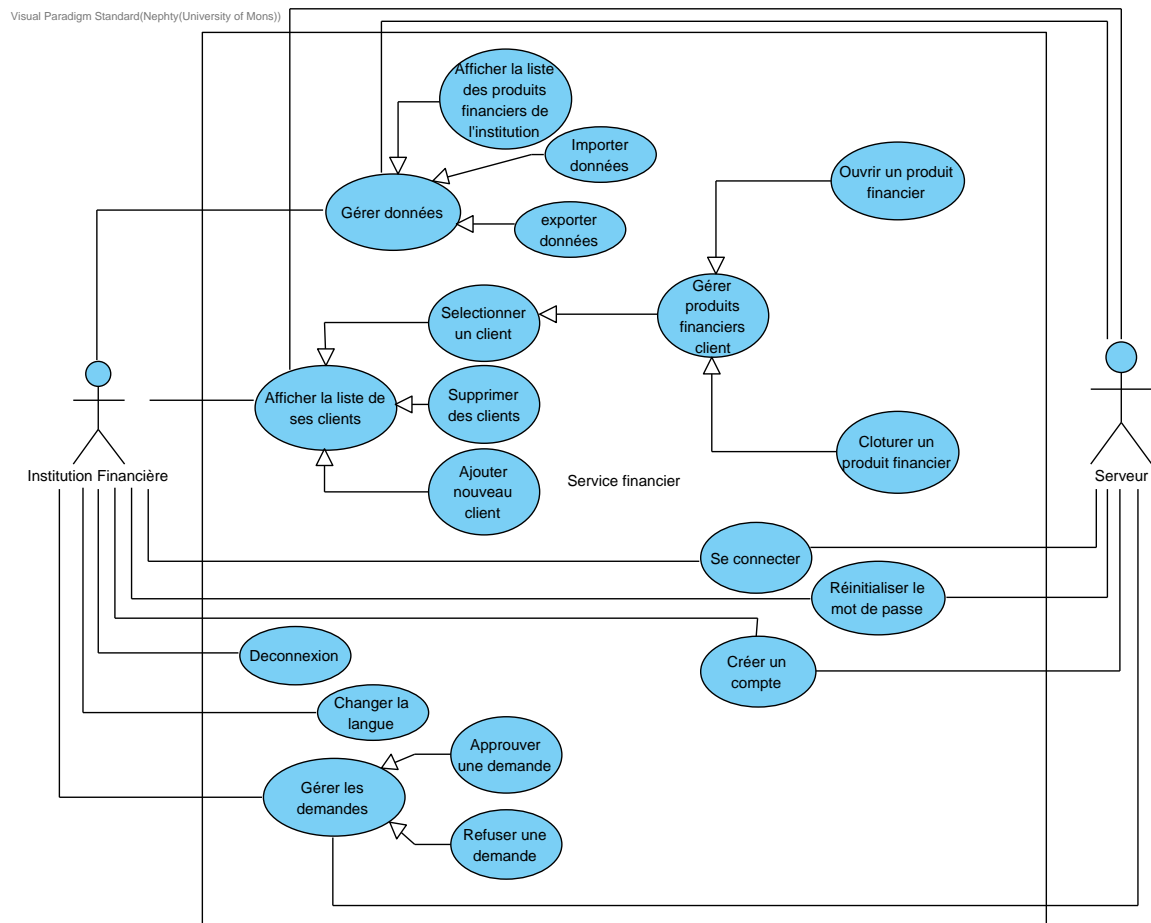


FIGURE 1.3 – Diagramme de case d'utilisation de l'application pour un service financier.

### 1.3.2 Interaction overview diagram

**Accès à application institution financière, en tant qu'institution financière** La seconde partie n'est accessible qu'en se connectant avec les identifiants d'un compte enregistré dans la base de données. Lors de la connexion, l'utilisateur (le responsable de l'institution, l'économe, le gestionnaire...) arrive sur l'écran principal, et peut, depuis celui-ci, accéder à toutes les fonctionnalités. Après s'être connecté, l'utilisateur peut effectuer plusieurs actions :

1. Afficher la liste des clients de l'institution ;
2. Gérer les données de l'institution ;
3. Accéder à la liste des demandes faites auprès de l'institution.

Quitter l'application, Changer la langue et Se déconnecter sont considéré comme des comportements triviaux.

**Afficher la liste des clients de l'institution** L'utilisateur peut accéder à la liste des clients et peut ajouter un client, exporter les données de ses clients ou en sélectionner un afin d'accéder aux détails.

**Sélectionner un client** Après avoir sélectionné un client, l'utilisateur peut supprimer le client de la base de données, clôturer un compte ou en ouvrir un nouveau (de n'importe quel type). Ces trois comportements sont considérés comme triviaux.

**Ajouter un client** L'utilisateur peut ajouter un nouveau client en spécifiant les informations nécessaires.

**Exporter les données de ses clients (implémente le tri et la recherche)** Lorsque l'utilisateur effectue une recherche ou un tri de ses clients, une liste différente de clients sera affichée à l'écran. Il pourra alors exporter cette liste "personnalisée" au format JSON ou CSV, qui contiendra les informations à disposition de la banque à propos de ses clients.

**Gérer les données de l'institution** Grâce à cette fonctionnalité, l'utilisateur connecté peut importer des données ou exporter toutes les données de tous les clients.

**Importer des données** L'import de données peut servir, par exemple, à restaurer une liste de client dans la base de données pour la mettre à disposition de la banque via l'interface graphique. L'utilisateur peut choisir un fichier au format JSON qui sera lisible par l'application.

**Exporter toutes les données de tous les clients** L'export de toutes les données des clients permet d'exporter toutes les données en un clic ; c'est un moyen alternatif, plus rapide, mais qui ne permet pas le tri ou la recherche de mots clés.

**Accéder à la liste des demandes** L'utilisateur peut, finalement, accéder aux demandes, qui sont de deux types : demandes de permission de transferts et demande d'ouverture de compte. En sélectionnant l'une ou l'autre catégorie, il pourra approuver ou refuser une demande.





### **1.3.3 Class diagram**

#### **1.3.4 Sequence diagram**

## Visual Paradigm Standard(Cyril University of Morav)



Avec ce modèle de donnée, nous avons essayé de couvrir au maximum les besoins de l'application tout en gardant un schéma assez compréhensible. Les deux entités *USER* et *BANKS* sont les entités centrale du schéma, elle permettent de gérer les données des utilisateurs ainsi que des banques. Chaque user est identifié grâce à son numéro de registre national et chaque banque est identifiée grâce à son numéro SWIFT.

Plusieurs entités sont utilisées pour les notification. Chaque notification est identifiée par un *NotificationID* qui permet de connaître tous les informations concernant cette notification (par exemple le type de notification ou l'état de celle-ci). Deux entités sont créée, permettant de différencier les notifications des clients et les notifications des banques.

19

le nouveau montant du compte après la transaction. Nous avons décidé de calculer les montants en ajoutant ou diminuant le montant des transactions au montant actuel du compte. Cela prendra effectivement plus de temps à calculer cependant cela permet de rendre la table TransactionLog plus concise et permet aussi d'éviter certaines erreurs.

## **1.5 Design du REST API**

## 1.6 Maquette de l'interface utilisateur : application client

### Fenêtres disponibles

- Auth ;
- Sign in ;
- Sign up ;
- Change password ;
- Change language ;
- Main screen ;
- Financial products ;
- Toggle product ;
- Enter PIN to confirm ;
- Details ;
- Transaction history ;
- Visualize history ;
- Export history ;
- Transfer ;
- Requests ;
- Request transfer permission ;
- Request new portfolio ;
- Requests status ;
- Notifications.

### Fenêtre *Auth*

**Accès** : en ouvrant l'application sans être connecté, en cliquant sur les boutons *Back* des fenêtres *Sign in* et *Sign up* ou en cliquant sur le bouton *Sign out* de la fenêtre *Main screen*.

**Contenu** : les boutons *Sign in*, *Sign up* et *Language*.

- Le bouton *Sign in* : envoie l'utilisateur sur la fenêtre *Sign in* ;
- Le bouton *Sign up* : envoie l'utilisateur sur la fenêtre *Sign up* ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language*.

### Fenêtre *Sign in*

**Accès** : en cliquant sur le bouton *Sign in* de la fenêtre *Auth* ou en créant un compte sur la fenêtre *Sign up*.

**Contenu** : les boutons *Back*, *Sign in* et *Language*, le champ de texte *Username or email...* et le champ de mot de passe *Password...* .

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Auth* ;
- Bouton *Sign in* : si les identifiants sont corrects, connecte l'utilisateur et l'envoie sur la fenêtre *Main screen*, sinon, affiche une erreur ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le champ de texte *Username or email...* : permet à l'utilisateur d'entrer son nom d'utilisateur ou son email ;
- Le champ de mot de passe *Password...* : permet à l'utilisateur d'entrer son mot de passe de manière discrète.

### Fenêtre *Sign up*

**Accès** : en cliquant sur le bouton *Sign up* de la fenêtre *Auth*.

**Contenu** : les boutons *Back*, *Sign up* et *Language*, les champs de texte *Last name...*, *First name...*, *NRN...*, *Email address...*, *Username...*, les champs de mot de passe *Password...* et *Confirm password...*, le label *Favourite language*, le menu déroulant *Language* et la case à cocher *Confirmation* .

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Auth* ;
- Bouton *Sign up* : si l'adresse email est au format correct, l'adresse email et le nom d'utilisateur ne sont pas déjà pris par un autre utilisateur, le numéro national est correctement formaté, le mot de passe est confirmé et la case à cocher est cochée, enregistre un nouveau profil dans la base de données et envoie l'utilisateur sur la fenêtre *Sign in* avec le champ de texte *Username or email...* pré-complété ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le champ de texte *Last name...* : permet à l'utilisateur d'entrer son nom ;
- Le champ de texte *First name...* : permet à l'utilisateur d'entrer son prénom ;
- Le champ de texte *NRN...* : permet à l'utilisateur d'entrer son numéro national ;
- Le champ de texte *Email address...* : permet à l'utilisateur d'entrer son adresse email ;
- Le champ de texte *Username...* : permet à l'utilisateur d'entrer un nom d'utilisateur ;
- Le champ de mot de passe *Password...* : permet à l'utilisateur d'entrer un mot de passe de manière discrète ;
- Le champ de mot de passe *Confirm password...* : permet à l'utilisateur d'entrer la confirmation du mot de passe de manière discrète ;
- Le menu déroulant *Language* : permet à l'utilisateur de sélectionner sa langue préférée ;
- La case à cocher *Confirmation* : demande à l'utilisateur de confirmer qu'il a sauvegardé son mot de passe de manière locale, sécurisée et permanente.

#### **Fenêtre *Change password***

**Accès** : en cliquant sur le bouton *Change password* de la fenêtre *Main screen*.

**Contenu** : les boutons *Back*, *Change password* et *Language* et les champs de mot de passe *Current password*, *New password* et *Confirm new password* et la case à cocher *Confirmation*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- Le bouton *Change password* : change le mot de passe lié à l'utilisateur si la case à cocher est cochée, que l'ancien mot de passe est correct et que la confirmation du nouveau mot de passe est correcte ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le champ de mot de passe *Current password* : permet à l'utilisateur d'entrer son mot de passe actuel de manière discrète ;
- Le champ de mot de passe *New password* : permet à l'utilisateur d'entrer le nouveau mot de passe désiré de manière discrète ;
- Le champ de mot de passe *Confirm new password* : permet à l'utilisateur d'entrer la confirmation du nouveau mot de passe de manière discrète ;
- La case à cocher *Confirmation* : demande à l'utilisateur de confirmer qu'il a mis à jour son mot de passe précédemment enregistré de manière locale, sécurisée et permanente.

#### **Fenêtre *Change language***

**Accès** : en cliquant sur le bouton *Language* de la fenêtre *Auth*, *Sign in*, *Sign up*, *Reset password* ou *Main screen*.

**Contenu** : les boutons *Back*, *Add...* et *Confirm*, le label *Choose a language* et la liste *Available languages*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre précédente ;



- Le bouton *Add...* : permet de sélectionner un fichier JSON correspondant à une langue afin de l'ajouter à la liste de langues disponibles ;
- Le bouton *Confirm* : envoie l'utilisateur sur la fenêtre *précente* et modifie la langue de l'application par la nouvelle langue sélectionnée ;
- La liste *Available languages* : contient une liste de toutes les langues disponibles et permet à l'utilisateur d'en sélectionner une.

### **Fenêtre *Main screen***

**Accès** : en se connectant à son compte de la fenêtre *Sign in*, en cliquant sur le bouton *Back* de la fenêtre *Financial products*, en cliquant sur les boutons *Change password* ou *Back* de la fenêtre *Change password*, en ayant entré trois fois un code PIN incorrect via la fenêtre *Enter PIN to confirm*, en cliquant sur le bouton *Back* de la fenêtre *Financial products* ou en cliquant sur le bouton *Back* de la fenêtre *Requests*.

**Contenu** : les boutons *Sign out*, *Language*, *Financial products*, *Requests* et *Change password*.

- Le bouton *Sign out* : envoie l'utilisateur sur la fenêtre *Auth* et déconnecte l'utilisateur ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le bouton *Financial products* : envoie l'utilisateur sur la fenêtre *Financial products* ;
- Le bouton *Requests* : envoie l'utilisateur sur la fenêtre *Requests* ;
- Le bouton *Change password* : envoie l'utilisateur sur la fenêtre *Change password*.

### **Fenêtre *Financial products***

**Accès** : en cliquant sur le bouton *Financial products* de la fenêtre *Main screen* ou en cliquant sur le bouton *Back* des fenêtres *Toggle product* et *Details*.

**Contenu** : les boutons *Back*, *Edit* et *Details*, le label *Choose a product* et la liste *Products*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- La liste *Products* : affiche la liste des produits financiers associés au compte (c'est-à-dire ceux dont l'utilisateur est titulaire ou co-titulaire) ;
- Le bouton *Edit* : envoie l'utilisateur sur la fenêtre *Toggle product* ;
- Le bouton *Details* : envoie l'utilisateur sur la fenêtre *Details*.

### **Fenêtre *Toggle product***

**Accès** : en cliquant sur le bouton *Edit* de la fenêtre *Financial products*.

**Contenu** : les boutons *Back* et *Toggle* et le label *Toggle selected product*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Financial product* ;
- Le bouton *Toggle* : envoie l'utilisateur sur la fenêtre *Enter PIN to confirm* et active/désactive le produit financier sélectionné selon son état actuel si la réponse de la fenêtre *Enter PIN to confirm* est positive (c'est-à-dire si le code PIN est correct).

### **Fenêtre *Enter PIN to confirm***

**Accès** : en cliquant sur le bouton *Toggle* de la fenêtre *Toggle product* ou sur le bouton *Confirm* de la fenêtre *Transfer* si le virement est effectué.

**Contenu** : les boutons *Back* et *Confirm*, un bouton par chiffre allant de 0 à 9, un bouton *Delete* et le label *Entered code*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *précédente* ;
- Le bouton *Confirm* : renvoie à la fenêtre précédente avec une réponse positive si le code entré est bon, affiche une erreur si le code entré est mauvais et que l'utilisateur a fait moins de trois tentatives

erronées, ou renvoie à la fenêtre *Main screen* et bloque le compte si l'utilisateur a fait trois tentatives erronées ;

- Les boutons allant de 0 à 9 : entre le chiffre correspondant et modifie le label *PIN* ;
- Le bouton *Delete* (<) : supprime la dernière entrée du code PIN.

### **Fenêtre *Details***

**Accès** : en cliquant sur le boutons *Details* de la fenêtre *Financial products* ou en cliquant sur le bouton *Back* de la fenêtre *Transaction history*.

**Contenu** : Les boutons *Back*, *Transfer* et *History*, les labels *Choose an account* et *Product* et la liste *Accounts*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Financial products* ;
- Le bouton *Transfer* : envoie l'utilisateur sur la fenêtre *Transfer* ;
- Le bouton *History* : envoie l'utilisateur sur la fenêtre *Transaction history* ;
- La liste *Accounts* : affiche les comptes associés au produit financier précédemment sélectionné.

### **Fenêtre *Transaction history***

**Accès** : en cliquant sur le bouton *History* de la fenêtre *Details* ou en cliquant sur le bouton *Back* de la fenêtre *Export...*, en cliquant sur le bouton *Back* de la fenêtre *Visualize*, en cliquant sur les boutons *Export to JSON format* ou *Export to CSV format* de la fenêtre *Export history*.

**Contenu** : les boutons *Back*, *Visualize* et *Export history*, les labels *Account name* et *Account ID* et la liste *Transactions*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Details* ;
- Le bouton *Visualize* : envoie l'utilisateur sur la fenêtre *Visualize history*
- Le bouton *Export history* : envoie l'utilisateur sur la fenêtre *Export history* ;
- La liste *Transactions* : contient les transactions passées.

### **Fenêtre *Visualize history***

**Accès** : en cliquant sur le bouton *Visualize* de la fenêtre *Transaction history*.

**Contenu** : les boutons *Back*, *Export...*, *Add* et *Remove*, les labels *Product name*, *Product ID*, *Time scale* et *Add/remove account*, les menus déroulants *Time scale* et *Accounts* et la zone d'affiche des graphiques.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Transaction history* ;
- Le bouton *Export* : envoie l'utilisateur sur la fenêtre *Export history* et lui permet d'exporter l'historique avec une échelle de temps différente ;
- Le bouton *Add* : ajoute le produit sélectionné à visualiser ;
- Le bouton *Remove* : retire le produit sélectionné visualisé ;
- Le menu déroulant *Time scale* : permet à l'utilisateur de choisir une échelle de temps à utiliser pour les graphiques ;
- le menu déroulant *Accounts* : permet à l'utilisateur de sélectionner un produit à ajouter/enlever de la visualisation.

### **Fenêtre *Export history***

**Accès** : en cliquant sur le bouton *Export...* de la fenêtre *Transaction history* ou en cliquant sur le bouton *Export...* de la fenêtre *Visualize history*.

**Contenu** : les boutons *Back*, *Choose path...*, *Export to JSON format* et *Export to CSV format* et les labels *Export location* et *Selected path*

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Transaction history* ;
- Le bouton *Choose path...* : ouvre l'explorateur de fichier afin de déterminer le chemin de destination ;
- Le bouton *Export to JSON format* : exporte l'historique au format JSON ;
- Le bouton *Export to CSV format* : exporte l'historique au format CSV.

### **Fenêtre *Transfer***

**Accès** : en cliquant sur le bouton *Transfer* de la fenêtre *Financial products*.

**Contenu** : les boutons *Back* et *Confirm*, les labels *Amount*, *Recipient*, *IBAN*, *Message* et *Date* et les champs de texte *Amount*, *Recipient*, *IBAN*, *Message* et *Date*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Financial products* ;
- Le bouton *Confirm* : envoie l'utilisateur sur la fenêtre *Enter PIN to confirm* si le montant, l'IBAN et la date sont correctement formatés, et effectue le virement si la réponse de la fenêtre *Enter PIN to confirm* est positive ;
- Le champ de texte *Amount* : permet à l'utilisateur d'entrer la somme à envoyer ;
- Le champ de texte *Recipient* : permet à l'utilisateur d'entrer le destinataire ;
- Le champ de texte *IBAN* : permet à l'utilisateur d'entrer l'IBAN du destinataire ;
- Le champ de texte *Message* : permet à l'utilisateur d'entrer la communication liée au virement ;
- Le champ de texte *Date* : permet à l'utilisateur d'entrer la date de planification du virement.

### **Fenêtre *Requests***

**Accès** : en cliquant sur le bouton *Request* de la fenêtre *Main screen*, en cliquant sur le bouton *Back* de la fenêtre *Request transfer permission* ou en cliquant sur le bouton *Back* de la fenêtre *Request new portfolio*.

**Contenu** : les boutons *Back*, *Transfer permission*, *New portfolio* et *Requests status*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- Le bouton *Transfer permission* : envoie l'utilisateur sur la fenêtre *Request transfer permission* ;
- Le bouton *New portfolio* : envoie l'utilisateur sur la fenêtre *Request new portfolio* ;
- Le bouton *Requests status* : envoie l'utilisateur sur la fenêtre *Requests status*.

### **Fenêtre *Request transfer permission***

**Accès** : en cliquant sur le bouton *Transfer permission* de la fenêtre *Requests*.

**Contenu** : les boutons *Back* et *Send request*, le label *Select portfolio* et le menu déroulant *Portfolio*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Requests* ;
- Le bouton *Send request* : affiche une confirmation que la demande a bien été envoyée si un portefeuille a été sélectionné ;
- Le menu déroulant *Portfolio* : contient les portefeuilles qui ne possèdent pas déjà la permission d'effectuer des virements.

### **Fenêtre *Request new portfolio***

**Accès** : en cliquant sur le bouton *New portfolio* de la fenêtre *Requests*.

**Contenu** : les boutons *Back* et *Send request*, le label *Select SWIFT* et le menu déroulant *SWIFT*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Requests* ;
- Le bouton *Send request* : affiche une confirmation que la demande a bien été envoyée si un code SIWFT a été sélectionné ;
- Le menu déroulant *SWIFT* : contient les codes SWIFT existants dans la base de donnée.

#### **Fenêtre *Requests status***

**Accès** : en cliquant sur le bouton *Requests status* de la fenêtre *Requests*.

**Contenu** : Le bouton *back* et la liste *Requests*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Requests* ;
- La liste *Requests* : affiche toutes les requêtes effectuées ainsi que leur status (validée, refusée, en attente).

#### **Fenêtre *Notifications***

**Accès** : en cliquant sur le bouton *Notifications* de la fenêtre *Main screen*.

**Contenu** : le bouton *Back* et la liste *Notifications*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- La liste *Notifications* : contient les notifications de l'utilisateur.

**Illustration du diagramme** Par souci de clareté, ce diagramme est disponible en annexe du rapport.

## 1.7 Maquette de l'interface utilisateur : application institution

### Fenêtres disponibles

- Auth ;
- Sign in ;
- Sign up ;
- Change password ;
- Change language ;
- Main screen ;
- Client ;
- Details ;
- Export data ;
- Create client account ;
- Requests ;
- Transfer permission requests ;
- Portfolio requests ;
- Manage data ;
- Import data.

### Fenêtre *Auth*

**Accès** : en ouvrant l'application sans être connecté, en cliquant sur les boutons *Back* des fenêtres *Sign in* et *Sign up* ou en cliquant sur le bouton *Sign out* de la fenêtre *Main screen*.

**Contenu** : les boutons *Sign in*, *Sign up* et *Language*.

- Le bouton *Sign in* : envoie l'utilisateur sur la fenêtre *Sign in* ;
- Le bouton *Sign up* : envoie l'utilisateur sur la fenêtre *Sign up* ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language*.

### Fenêtre *Sign in*

**Accès** : en cliquant sur le bouton *Sign in* de la fenêtre *Auth* ou en créant un compte sur la fenêtre *Sign up*.

**Contenu** : les boutons *Back*, *Sign in* et *Language*, le champ de texte *SWIFT...* et le champ de mot de passe *Password...*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Auth* ;
- Bouton *Sign in* : si les identifiants sont corrects, connecte l'utilisateur et l'envoie sur la fenêtre *Main screen*, sinon, affiche une erreur ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le champ de texte *SWIFT...* : permet à l'utilisateur d'entrer le code SWIFT de la banque ;
- Le champ de mot de passe *Password...* : permet à l'utilisateur d'entrer le mot de passe du compte de la banque de manière discrète.

### Fenêtre *Sign up*

**Accès** : en cliquant sur le bouton *Sign up* de la fenêtre *Auth*, les champs de texte *SWIFT...*, *City...*, *Country...*, *Name...* et *Password...*, les champs de mot de passe *Password...* et *Confirm password...*, le label *Favorite language*, le menu déroulant *Language* et la case à cocher *Confirmation*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Auth* ;

- Bouton *Sign up* : si le code SWIFT est au format correct et disponible, les champs de texte *City...*, *Country...* et *Name...* sont remplis, le mot de passe est confirmé, la langue favorite est sélectionnée et la case est cochée, enregistre un nouveau profil dans la base de données et envoie l'utilisateur sur la fenêtre *Sign in* avec le champ de texte *SWIFT...* pré-complété ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le champ de texte *SWIFT...* : permet à l'utilisateur d'entrer le code SWIFT de la banque ;
- Le champ de texte *City...* : permet à l'utilisateur d'entrer la ville dans laquelle la banque est basée ;
- Le champ de texte *Country...* : permet à l'utilisateur d'entrer le pays dans lequel la banque est basée ;
- Le champ de texte *Name...* : permet à l'utilisateur d'entrer le nom de la banque ;
- Le champ de mot de passe *Password...* : permet à l'utilisateur d'entrer un mot de passe de manière discrète ;
- Le champ de mot de passe *Confirm password...* : permet à l'utilisateur d'entrer la confirmation du mot de passe de manière discrète ;
- Le menu déroulant *Language* : permet à l'utilisateur de sélectionner sa langue préférée ;
- La case à cocher *Confirmation* : demande à l'utilisateur de confirmer qu'il a sauvegardé son mot de passe de manière locale, sécurisée et permanente.

### **Fenêtre *Change password***

**Accès** : en cliquant sur le bouton *Change password* de la fenêtre *Main screen*.

**Contenu** : les boutons *Back*, *Change password* et *Language* et les champs de mot de passe *Current password*, *New password* et *Confirm new password* et la case à cocher *Confirmation*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- Le bouton *Change password* : change le mot de passe lié à l'utilisateur si la case à cocher est cochée, que l'ancien mot de passe est correct et que la confirmation du nouveau mot de passe est correcte ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le champ de mot de passe *Current password* : permet à l'utilisateur d'entrer son mot de passe actuel de manière discrète ;
- Le champ de mot de passe *New password* : permet à l'utilisateur d'entrer le nouveau mot de passe désiré de manière discrète ;
- Le champ de mot de passe *Confirm new password* : permet à l'utilisateur d'entrer la confirmation du nouveau mot de passe de manière discrète ;
- La case à cocher *Confirmation* : demande à l'utilisateur de confirmer qu'il a mis à jour son mot de passe précédemment enregistré de manière locale, sécurisée et permanente.

### **Fenêtre *Change language***

**Accès** : en cliquant sur le bouton *Language* de la fenêtre *Auth*, *Sign in*, *Sign up*, *Reset password* ou *Main screen*.

**Contenu** : les boutons *Back*, *Add...* et *Confirm*, le label *Choose a language* et la liste *Available languages*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre précédente ;
- Le bouton *Add...* : permet de sélectionner un fichier JSON correspondant à une langue afin de l'ajouter à la liste de langues disponibles ;
- Le bouton *Confirm* : envoie l'utilisateur sur la fenêtre précédente et modifie la langue de l'application par la nouvelle langue sélectionnée ;
- La liste *Available languages* : contient une liste de toutes les langues disponibles et permet à l'utilisateur d'en sélectionner une.

### **Fenêtre Main screen**

**Accès** : en se connectant à son compte de la fenêtre *Sign in*, en cliquant sur le bouton *Back* de la fenêtre *Clients*, en cliquant sur les boutons *Change password* ou *Back* de la fenêtre *Change password*, en cliquant sur le bouton *Back* de la fenêtre *Financial products* ou en cliquant sur le bouton *Back* de la fenêtre *Manage data*.

**Contenu** : les boutons *Sign out*, *Language*, *Clients*, *Manage requests*, *Manage data* et *Change password*.

- Le bouton *Sign out* : envoie l'utilisateur sur la fenêtre *Auth* et déconnecte l'utilisateur ;
- Le bouton *Language* : envoie l'utilisateur sur la fenêtre *Change language* ;
- Le bouton *Clients* : envoie l'utilisateur sur la fenêtre *Clients* ;
- Le bouton *Manage requests* : envoie l'utilisateur sur la fenêtre *Requests* ;
- Le bouton *Manage data* : envoie l'utilisateur sur la fenêtre *Manage data* ;
- Le bouton *Change password* : envoie l'utilisateur sur la fenêtre *Change password*.

### **Fenêtre Clients**

**Accès** : en cliquant sur le bouton *Clients* de la fenêtre *Main screen*, en cliquant sur le bouton *Back* de la fenêtre *Details*, en cliquant sur le bouton *Back*, *Export to JSON format* ou *Export to CSV format* de la fenêtre *Export data* ou en cliquant sur le bouton *Back* de la fenêtre *Add client*.

**Contenu** : les boutons *Back*, *Search*, *Export data...*, *Add client* et *Details*, les labels *Choose a client*, *Search name or ID*, *Sort by*, le champ de texte *Enter search...*, le menu déroulant *Sort* et la liste *Clients*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- La liste *Clients* : affiche la liste des clients associés à l'institution (c'est-à-dire ceux étant titulaires ou co-titulaires d'un compte dans cette institution) ;
- Le bouton *Export data...* : envoie l'utilisateur sur la fenêtre *Export data* ;
- Le bouton *Add client* : envoie l'utilisateur sur la fenêtre *Add client* ;
- Le bouton *Details* : envoie l'utilisateur sur la fenêtre *Details* ;
- Le bouton *Search* : effectue une recherche en fonction des critères entrés par l'utilisateur ;
- Le champ de texte *Enter search...* : permet à l'utilisateur d'entrer sa recherche ;
- Le menu déroulant *Sort* : permet à l'utilisateur de choisir un critère de tri des résultats.

### **Fenêtre Add client**

**Accès** : en cliquant sur le bouton *Add client* de la fenêtre *Clients*.

**Contenu** : les boutons *Back* et *Add client*, le label *Client NRN* et le champ de texte *NRN...*

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Clients* ;
- Le bouton *Add client* : ajoute un client dans la table des clients de la banque dans la base de données ;
- Le champ de texte *NRN...* : permet à l'utilisateur d'entrer le numéro de registre national du client.

### **Fenêtre Details**

**Accès** : en cliquant sur le bouton *Details* de la fenêtre *Clients*, en cliquant sur les boutons *Back*, *Export to JSON format* ou *Export to CSV format* de la fenêtre *Export data* ou en cliquant sur le bouton *Back* de la fenêtre *Create account*.

**Contenu** : les boutons *Back*, *Remove client*, *Search*, *Export data...*, *Create account* et *Close account*, les labels *Choose an account*, *Client*, *Search IBAN* et *Sort by*, le champ de texte *Enter search...*, le menu déroulant *Sort* et la liste *Accounts*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Clients* ;
- Le bouton *Remove client* : enlève le client de la banque ;
- Le bouton *Search* : effectue une recherche en fonction des critères entrés par l'utilisateur ;

- Le bouton *Export data...* : envoie l'utilisateur sur la fenêtre *Export data* ;
- Le bouton *Create account* : envoie l'utilisateur sur la fenêtre *Create account* ;
- Le bouton *Close account* : supprime le compte de ce client ;
- Le champ de texte *Enter search...* : permet à l'utilisateur d'entrer sa recherche ;
- Le menu déroulant *Sort* : permet à l'utilisateur de choisir un critère de tri des résultats.

#### **Fenêtre *Export data***

**Accès** : en cliquant sur le bouton *Export data...* de la fenêtre *Clients* ou *Details* ou en cliquant sur le bouton *Export all client data* de la fenêtre *Manage data*.

**Contenu** : les boutons *Back*, *Choose path...*, *Export to JSON format* et *Export to CSV format* et les labels *Export location* et *Selected path*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *précédente* ;
- Le bouton *Choose path...* : ouvre l'explorateur de fichier afin de déterminer le chemin de destination ;
- Le bouton *Export to JSON format* : exporte l'historique au format JSON ;
- Le bouton *Export to CSV format* : exporte l'historique au format CSV.

#### **Fenêtre *Create client account***

**Accès** : en cliquant sur le bouton *Create account* de la fenêtre *Details*.

**Contenu** : les boutons *Back* et *Create account*, les labels *Account type*, *Client name*, *IBAN* et *Client ID* et le menu déroulant *Type*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Details* ;
- Le bouton *Create account* : envoie l'utilisateur sur la fenêtre *Details* et crée le compte ;
- Le menu déroulant *Type* : permet à l'utilisateur d'entrer le type de compte à créer.

#### **Fenêtre *Requests***

**Accès** : en cliquant sur le bouton *Manage requests* de la fenêtre *Main screen*, en cliquant sur le bouton *Back* de la fenêtre *Transfer permission requests* ou en cliquant sur le bouton *Back* de la fenêtre *Portfolio requests*.

**Contenu** : les boutons *Back*, *Transfer permission requests* et *Portfolio requests*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- Le bouton *Transfer permission requests* : envoie l'utilisateur sur la fenêtre *Transfer permission requests* ;
- Le bouton *Portfolio requests* : envoie l'utilisateur sur la fenêtre *Portfolio requests*.

#### **Fenêtre *Transfer permission requests***

**Accès** : en cliquant sur le bouton *Transfer permission requests* de la fenêtre *Requests*.

**Contenu** : les boutons *Back*, *Deny* et *Approve*, le label *Choose a request* et la liste *Requests*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Requests* ;
- Le bouton *Deny* : refuse la requête sélectionnée ;
- Le bouton *Approve* : accepte la requête sélectionnée ;
- La liste *Requests* : liste les requêtes des utilisateurs.

#### **Fenêtre *Portfolio requests***

**Accès** : en cliquant sur le bouton *Portfolio requests* de la fenêtre *Requests*.

**Contenu** : les boutons *Back*, *Deny* et *Approve*, le label *Choose a request* et la liste *Requests*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Requests* ;



- Le bouton *Deny* : refuse la requête sélectionnée ;
- Le bouton *Approve* : accepte la requête sélectionnée ;
- La liste *Requests* : liste les requêtes des utilisateurs.

#### **Fenêtre *Manage data***

**Accès** : en cliquant sur le bouton *Manage data* de la fenêtre *Main screen* ou en cliquant sur le bouton *Back* de la fenêtre *Import data*.

**Contenu** : les boutons *Back*, *Import data* et *Export all client data*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- Le bouton *Import data* : envoie l'utilisateur sur la fenêtre *Import data* ;
- Le bouton *Export all client data* : envoie l'utilisateur sur la fenêtre *Export data*.

#### **Fenêtre *Import data***

**Accès** : en cliquant sur le bouton *Import data* de la fenêtre *Manage data*.

**Contenu** : les boutons *Back*, *Choose file...* et *Import file* et les labels *Choose file* et *Selected file*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Manage data* ;
- le bouton *Choose file...* : ouvre l'explorateur de fichier afin de déterminer le fichier à importer ;
- Le bouton *Import file* : importe les données des fichiers.

**Illustration du diagramme** Par souci de clarté, ce diagramme est disponible en annexe du rapport.

## **Chapitre 2**

### **Extension A : Gestion de cartes - Augustin HOUBA**

## **2.1 Diagrammes de conception UML : application client**

### **2.1.1 Use case diagram**

### **2.1.2 Interaction overview diagram**

### **2.1.3 Class diagram**

#### **2.1.4 Sequence diagram**

## **2.2 Diagrammes de conception UML : application institution**

### **2.2.1 Use case diagram**



### **2.2.2 Interaction overview diagram**

### **2.2.3 Class diagram**

#### **2.2.4 Sequence diagram**

## **Chapitre 3**

### **Extension B : Gestion de devises et virements internationaux - Cyril MOREAU**



## Visual Paradigm Standard(Cyrt)(University of Marse)



45

### 3.1.3 Class diagram

Visual Paradigm Standard(CyrtUniversity of Mors)

Logic

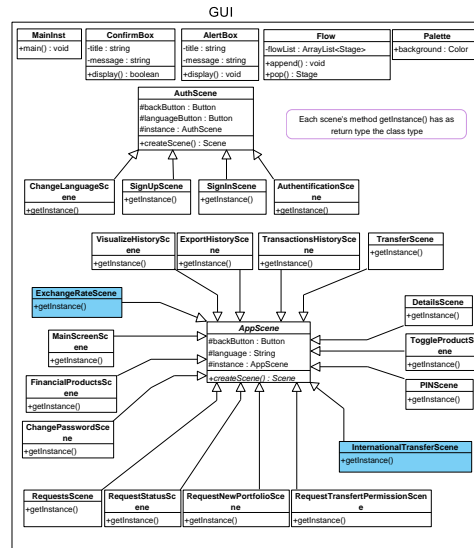
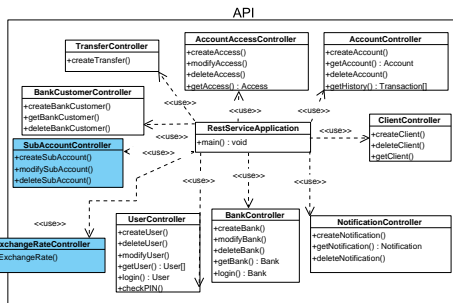
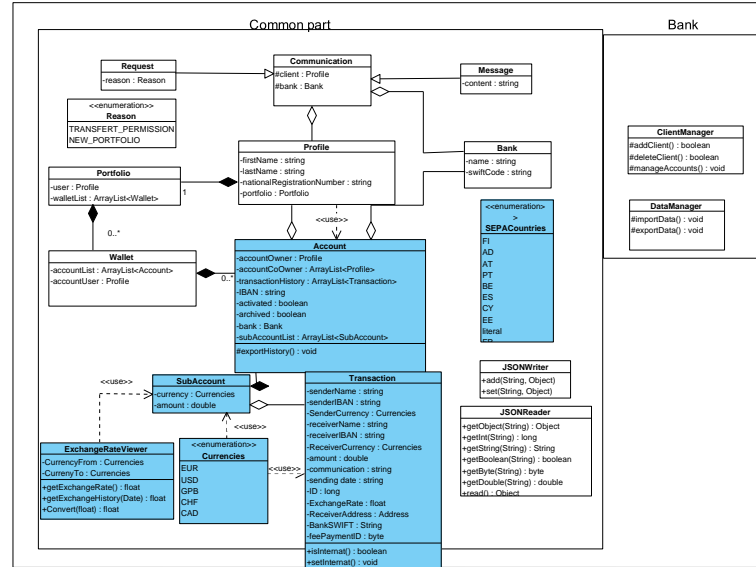


FIGURE 3.3 – Diagramme de classe de l'application client de l'extension 2

### 3.1.4 Sequence diagram

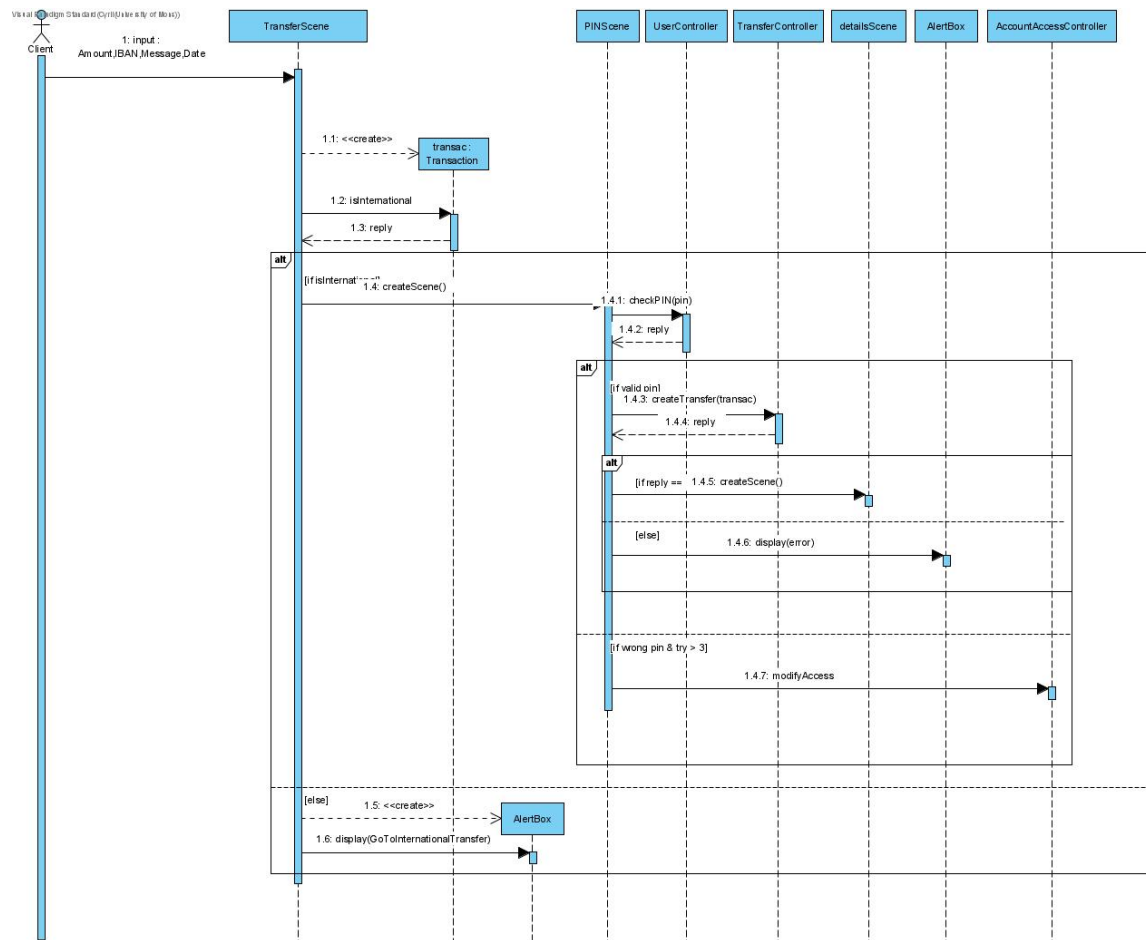


FIGURE 3.4 – Effectuer un virement



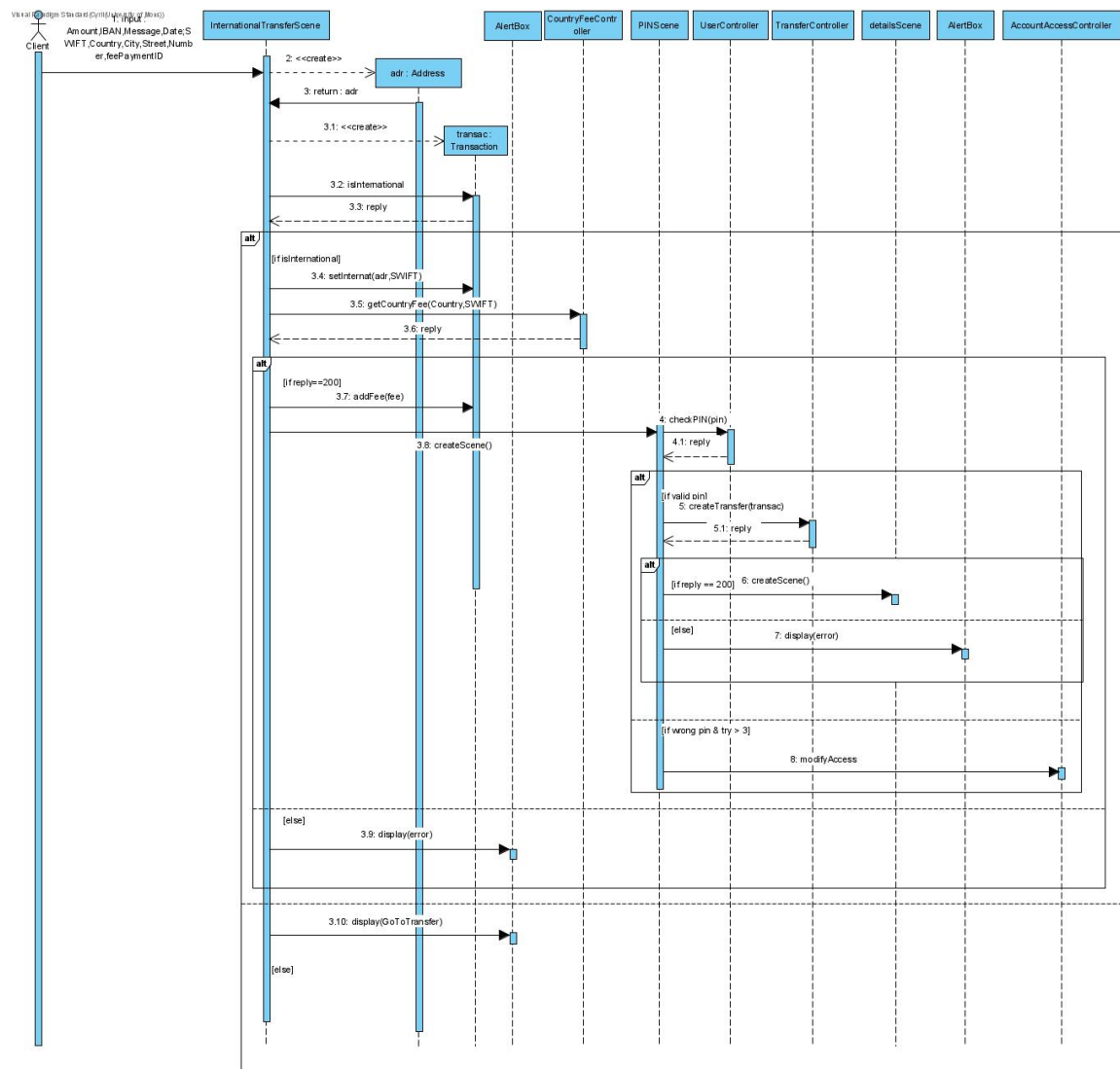


FIGURE 3.5 – Virement international

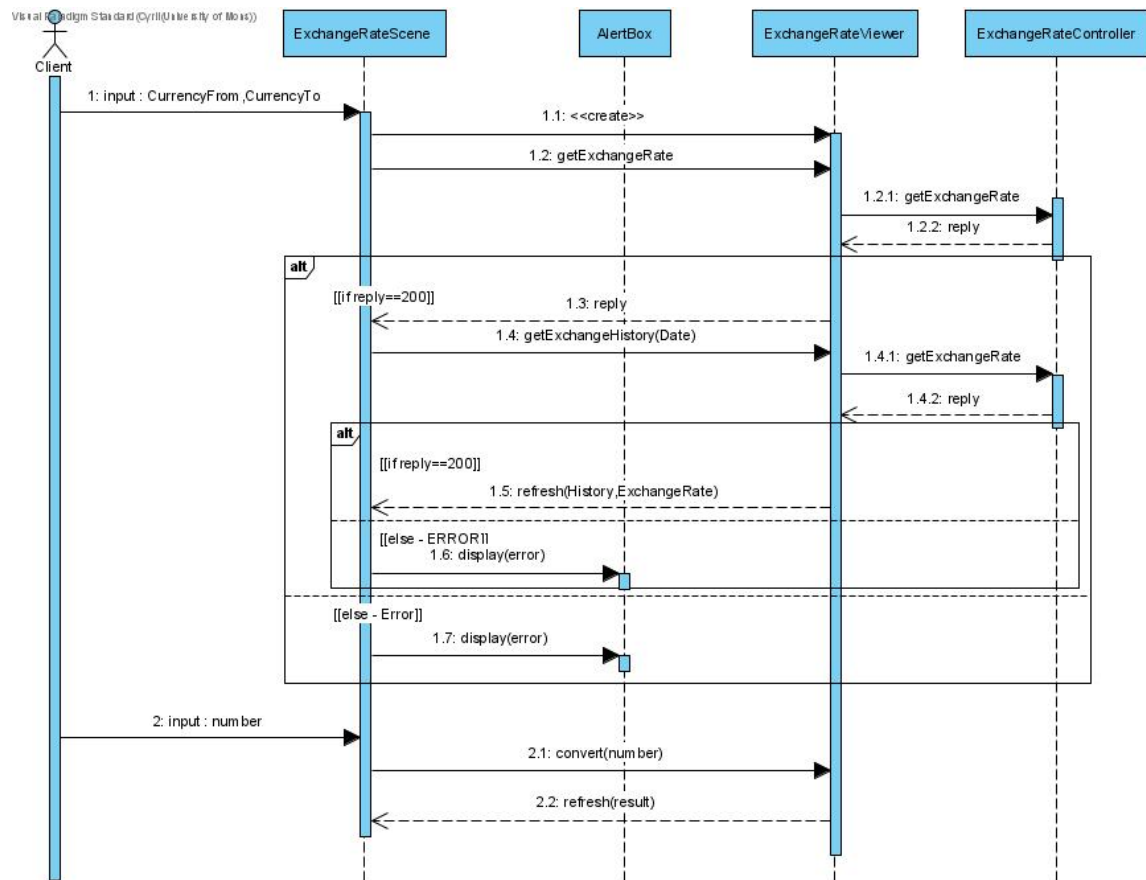


FIGURE 3.6 – Voir taux conversion

## 3.2 Diagrammes de conception UML : application institution

### 3.2.1 Use case diagram

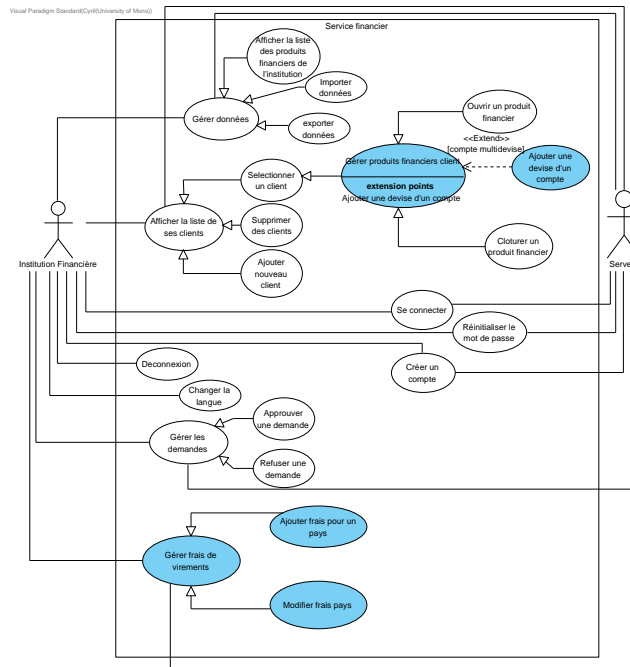


FIGURE 3.7 – Diagramme de case d'utilisation de l'application service financier de l'extension 2.

Ici, peu de use case ont dû être ajouté. Seul une extension du use case *Gérer produits financiers client* qui n'est disponible uniquement pour les comptes multidevises. Des cas d'utilisation ont également été ajouté afin de permettre aux institution d'ajouter des frais aux pays ne se trouvent pas en zone SEPA, si les pays ne sont pas dans cette liste, le virement ne peut pas être effectué. Elle permet d'ajouter une devise au compte. Ici également, la description semi-structurée des nouveaux use case se trouve dans un PDF attaché à ce rapport.

### 3.2.2 Interaction overview diagram

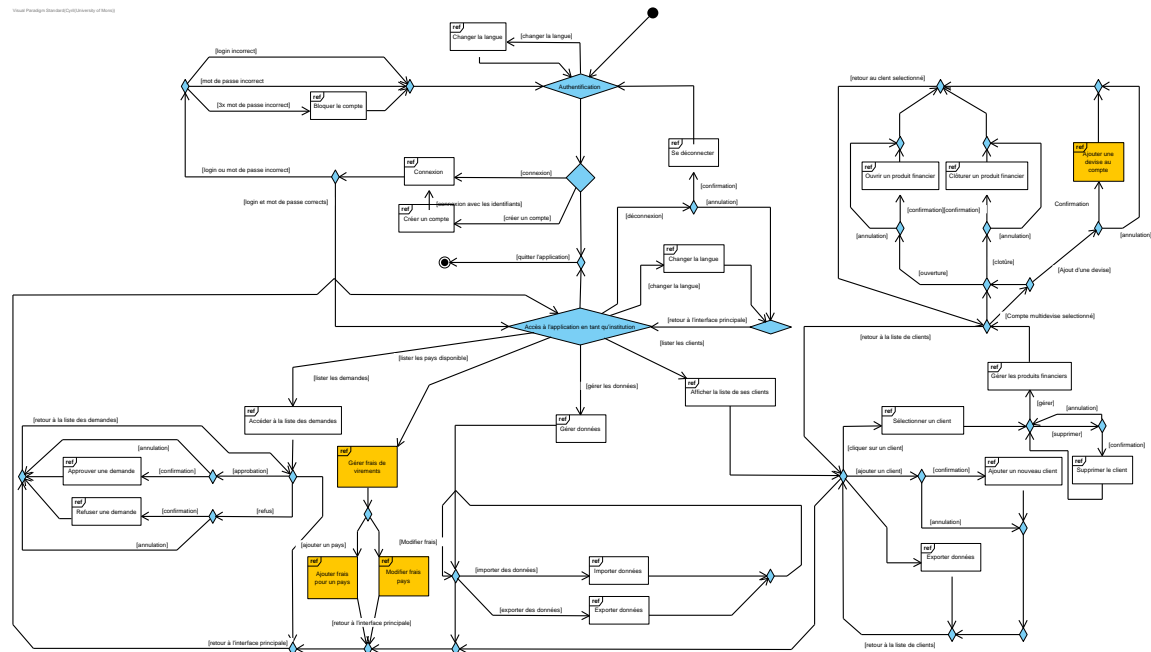


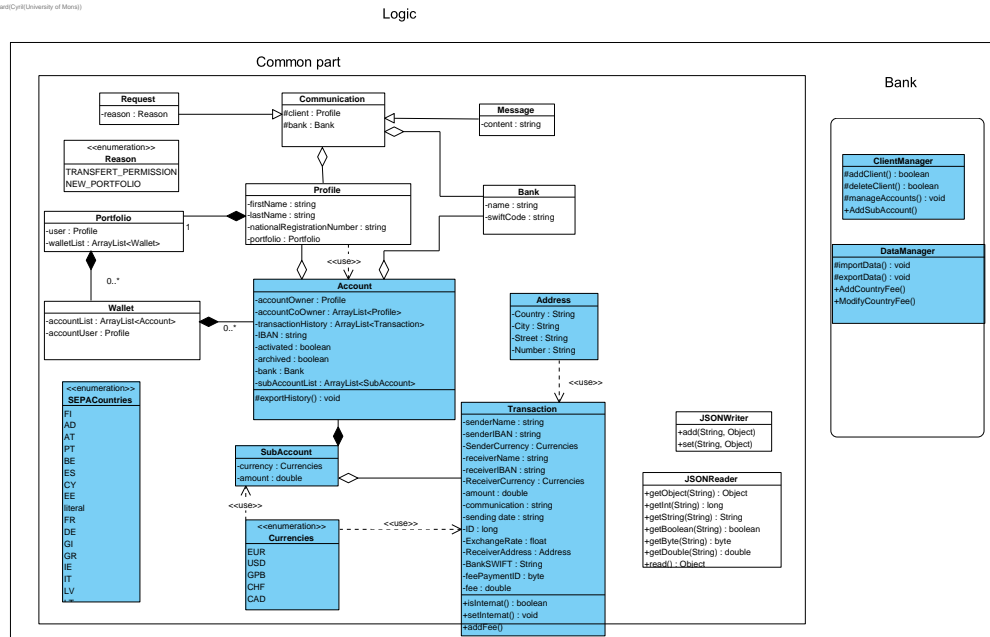
FIGURE 3.8 – Interaction overview diagram de l’application service financier de l’extension 2.

Dans ce diagramme, plusieurs choses ont été ajoutées :

Tout d’abords, lorsque l’institution souhaite gérer un produit financier, si celui-ci est un compte multi-devise, elle a la possibilité d’ajouter une devise. Ensuite, l’institution a maintenant la possibilité, après s’être authentifié, de gérer les pays vers lesquels elle autorise d’effectuer des virements (pays hors zone SEPA). Elle peut ajouter un pays dans la liste et modifier les frais lorsqu’un client effectue un virement vers un compte d’une banque située dans un de ces pays.

### 3.2.3 Class diagram

Visual Paradigm Standard(CyrilUniversity of Mons)



GUI

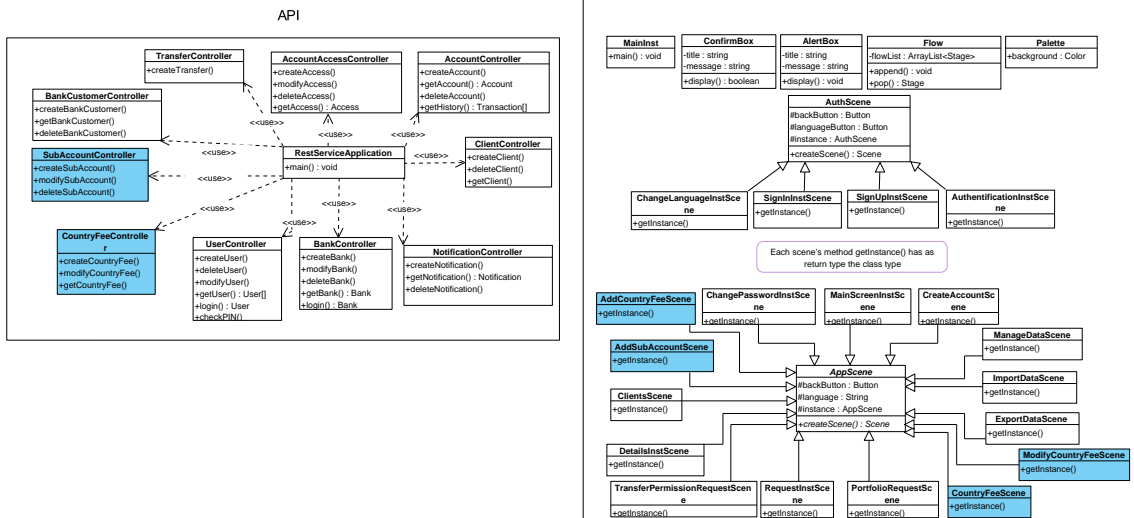


FIGURE 3.9 – Diagramme de classe de l'application service financier de l'extension 2

### 3.2.4 Sequence diagram

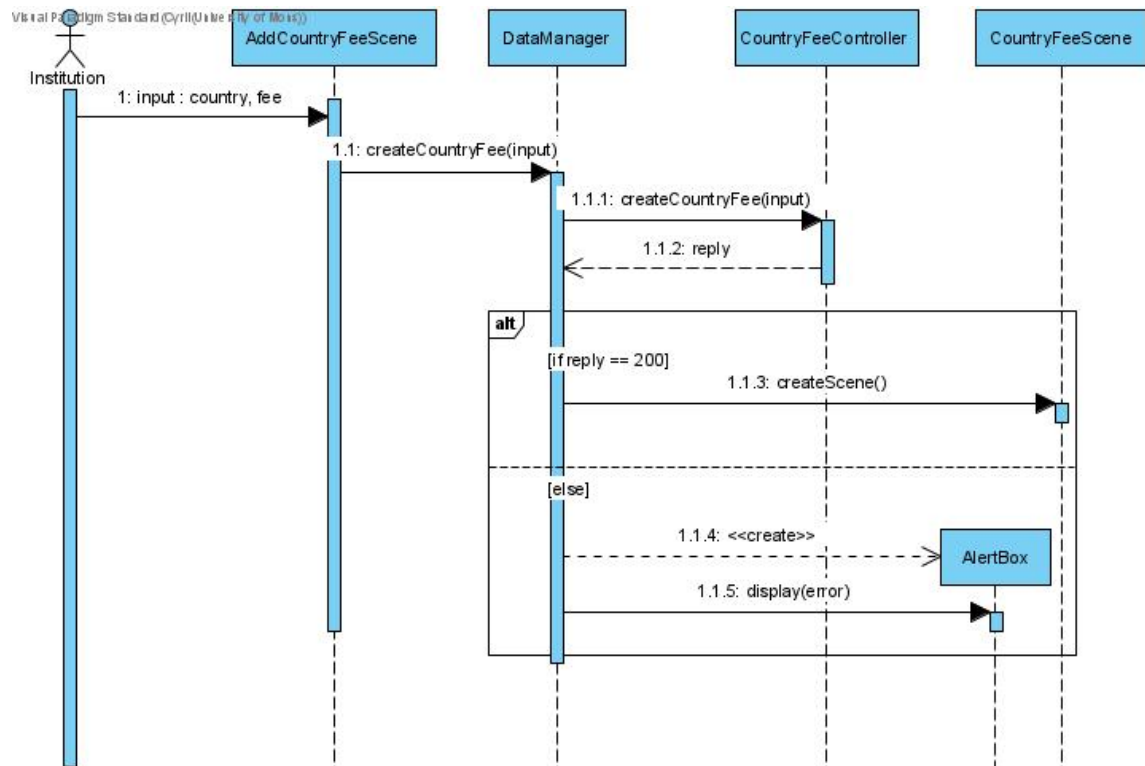


FIGURE 3.10 – Ajouter frais pour un pays

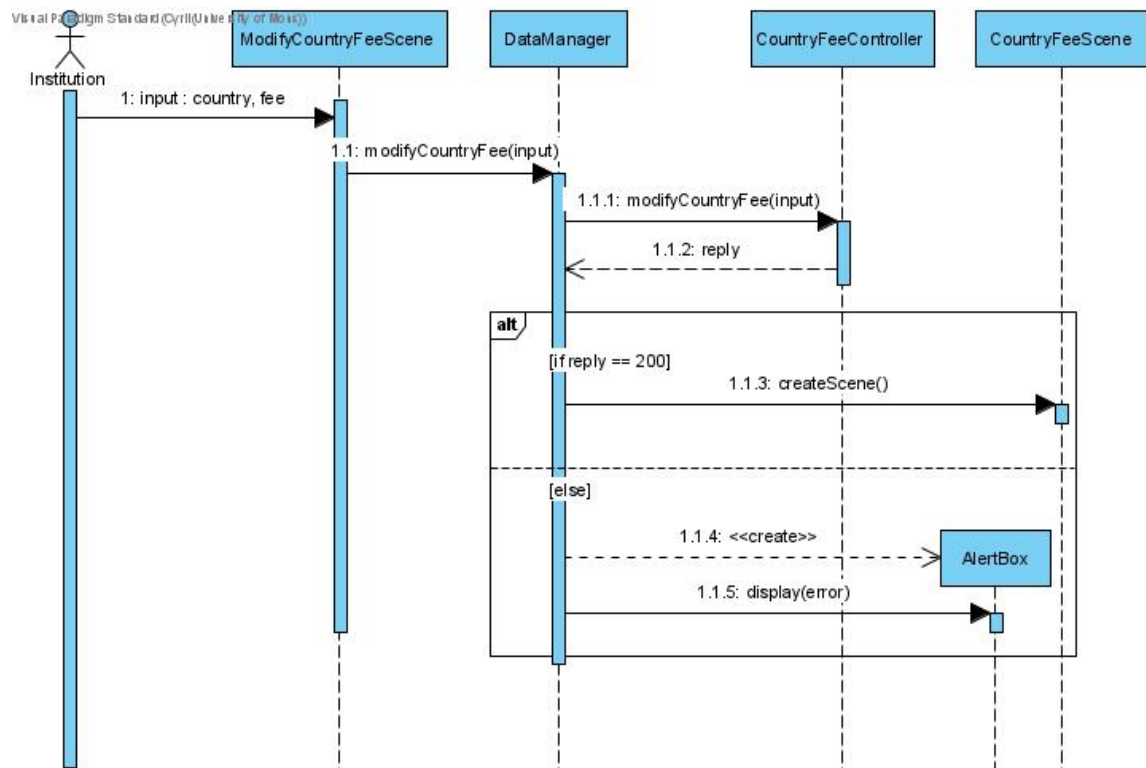


FIGURE 3.11 – Modifier frais pour un pays

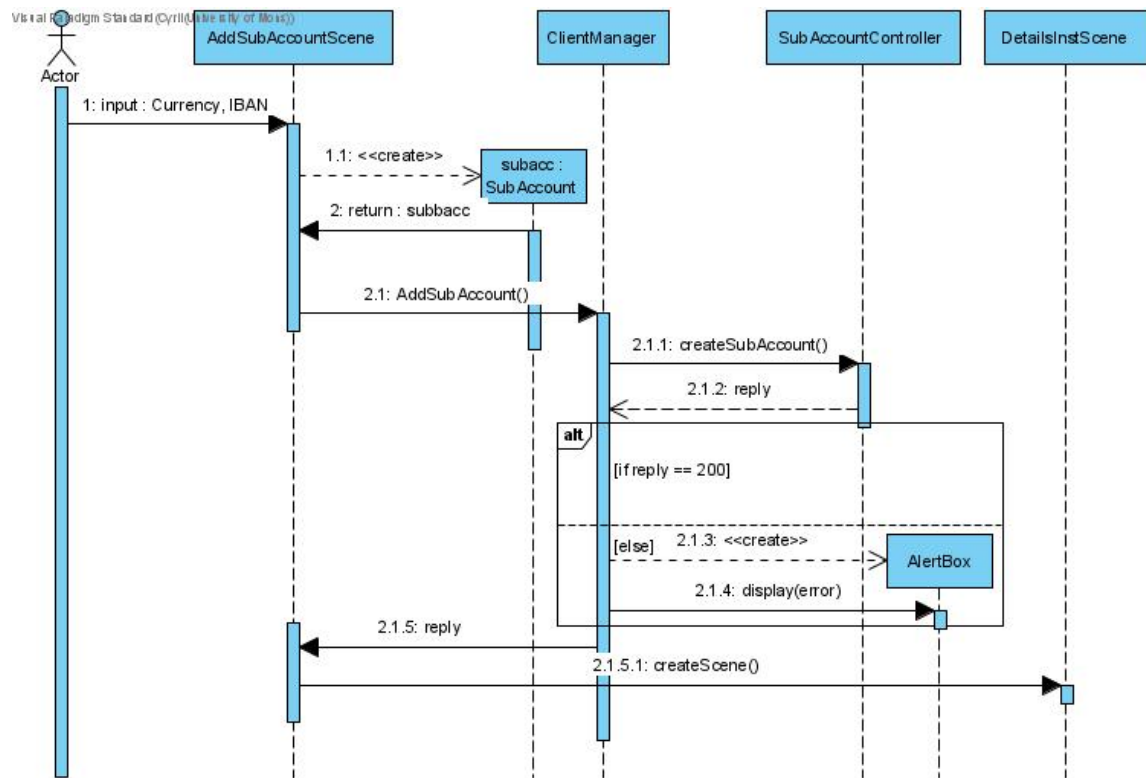


FIGURE 3.12 – Ajouter une devise pour un compte



### 3.3 Modèle de données

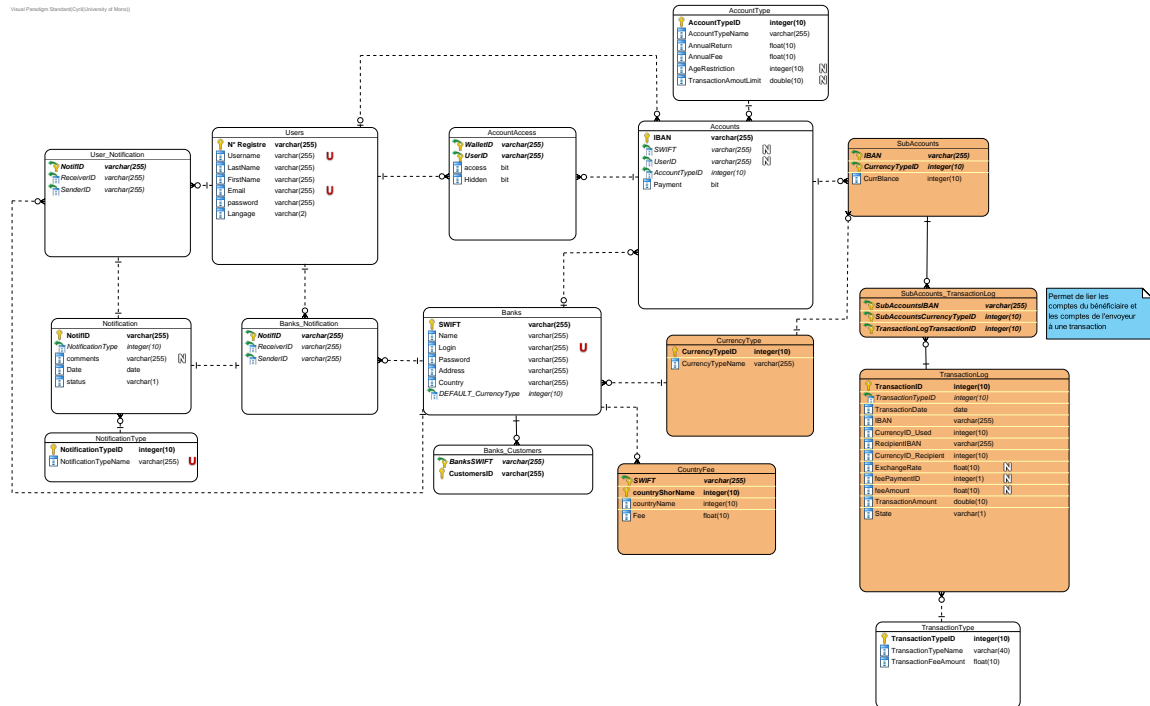


FIGURE 3.13 – Diagramme d'entité relation de l'extension 2.

Pour cette extension, de nombreux changements ont été effectués au niveau du diagramme d'entité relation.

Tout d'abord, chaque compte contient plusieurs subcomptes qui sont différencier par le type de devise. Ce qui permet de créer des comptes multi-devises. Si un compte n'est pas multi-devise, il ne possèdera qu'un seul subaccount.

L'ajout de ces SubAccount implique quelques changement de l'entité *TransactionLog* car il faut effectuer le virement depuis un subaccount vers un autre subaccount. C'est pour cela que les deux attributs *CurrencyI\_Used* et *CurrencyID\_Recipient* ont été ajouté car ils permettent de décrire de quel devise a été envoyé l'argent et vers quelle devise. Un attribut est également ajouté afin de connaître le taux de conversion entre les deux devise au moment de la transaction.

Une entité *CountryFee* a également été ajoutée afin de connaître les frais de virement internationaux pour chaque banque. Si un utilisateur souhaite effectuer un virement vers un pays qui n'est pas dans la liste de sa banque, le virement ne peut pas être effectué.

### **3.4 Maquette de l'interface de l'application client**

### **3.5 Maquette de l'interface de l'application pour une institution**

## **Chapitre 4**

### **Extension C : Gestion des contrats d'assurance - François VION**

## **4.1 Diagrammes de conception UML : application client**

### **4.1.1 Use case diagram**

### **4.1.2 Interaction overview diagram**

### **4.1.3 Class diagram**

#### **4.1.4 Sequence diagram**



## **4.2 Diagrammes de conception UML : application institution**

### **4.2.1 Use case diagram**

### **4.2.2 Interaction overview diagram**

### **4.2.3 Class diagram**

#### **4.2.4 Sequence diagram**

## **Chapitre 5**

### **Extension D : Paiements et gestion de fraudes - Arnaud MOREAU**

## 5.1 Diagrammes de conception UML : application client

### 5.1.1 Use case diagram

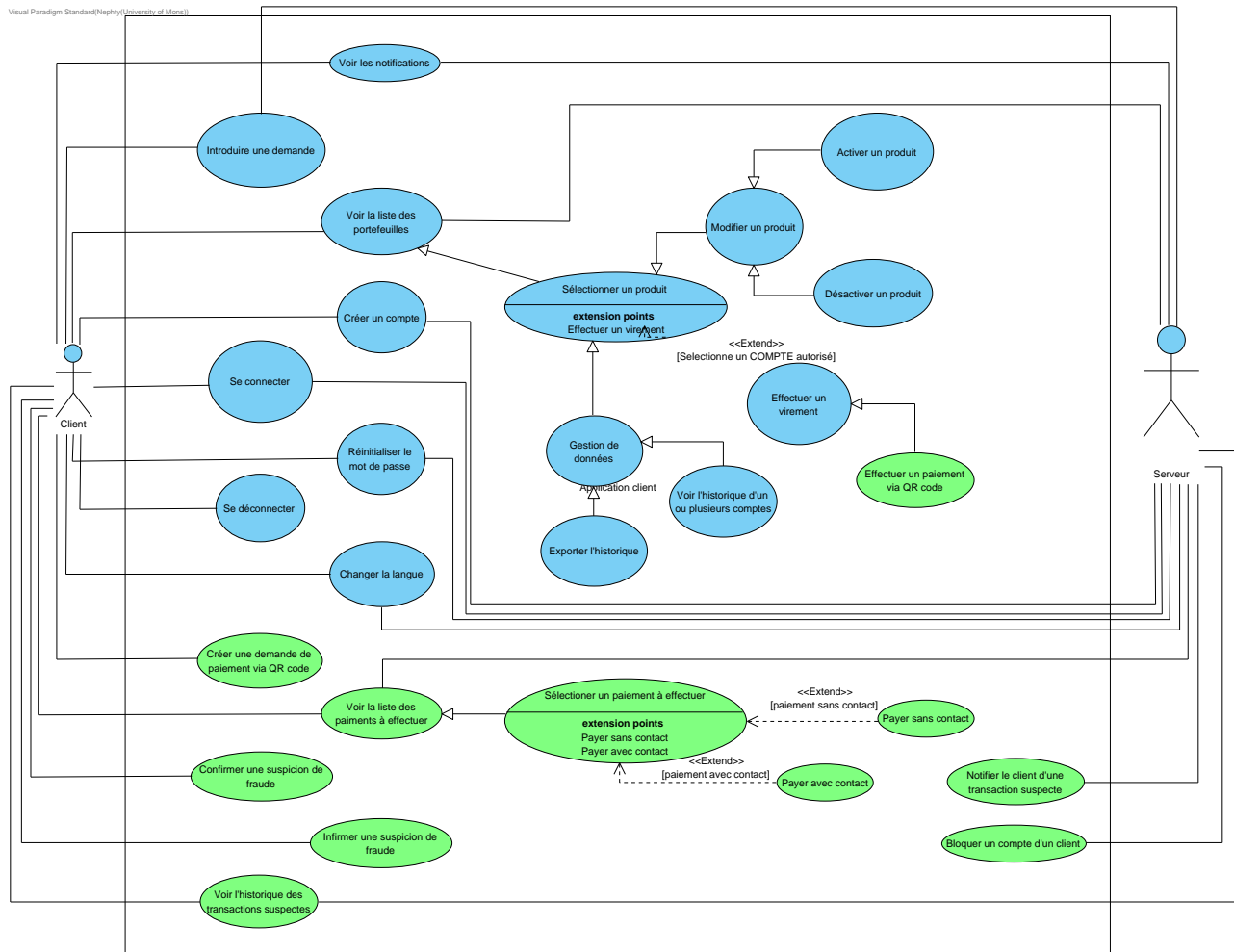


FIGURE 5.1 – Nouveau diagramme de cas d'utilisation de l'application client

Acteur	Client, serveur
Description	Permet à l'utilisateur de créer un QR code qui pourra plus tard être scanné afin d'effectuer le paiement défini
Préconditions	L'utilisateur est authentifié
Postconditions	L'utilisateur possède un QR code sous forme d'image qui permet de réaliser le paiement qu'il a décrit
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur arrive sur la fenêtre ;</li> <li>2. L'utilisateur entre un montant (chiffre à virgule) et éventuellement un message ;</li> <li>3. L'utilisateur choisi un chemin de destination pour l'export ;</li> <li>4. L'utilisateur clique sur le bouton <i>Generate</i> et une image contenant le QR code est créée dans le dossier spécifié ;</li> <li>5. Une transaction ayant pour état "en attente" est enregistrée dans la base de donnée.</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. L'utilisateur entre un montant que l'on ne peut pas convertir en <i>double</i> (car il y a des lettres par exemple) : un message d'erreur est affiché est l'export n'a pas lieu.</li> </ol>
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Generate</i> de la fenêtre <i>QR codes</i>
Fréquence d'utilisation	Rare

TABLE 5.1 – Créer une demande de paiement via QR code

Acteur	Client, serveur
Description	Permet à l'utilisateur d'utiliser un QR code afin d'effectuer un paiement
Préconditions	L'utilisateur est authentifié et possède un QR code sous forme d'image
Postconditions	Le paiement a été effectué : le compte sélectionné est débité et le destinataire reçoit l'argent
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur arrive sur la fenêtre ;</li> <li>2. L'utilisateur sélectionne l'image contenant le QR code ;</li> <li>3. L'utilisateur choisi le compte à partir duquel débiter l'argent ;</li> <li>4. L'utilisateur clique sur le bouton <i>Pay</i> ;</li> <li>5. Un appel serveur demandant le montant disponible sur le compte sélectionné est réalisé afin de vérifier si l'utilisateur peut payer à partir du compte sélectionné ;</li> <li>6. L'utilisateur est invité à entrer son code PIN ;</li> <li>7. Une requête de modification de la transaction correspondante est envoyée au serveur afin de modifier son état de "en attente" à "confirmée". La transaction pour alors avoir eu lieu et prendre l'état "traitée".</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. L'utilisateur sélectionne une image ne contenant pas de QR code : un message d'erreur s'affiche, la sélection redevient vide (plus d'image sélectionnée) et l'utilisateur peut sélectionner une nouvelle image ;</li> <li>2. Le montant à payer est supérieur au montant contenu dans le compte sélectionné : un message d'erreur s'affiche, la sélection du menu déroulant est réinitialisée et l'utilisateur peut à nouveau sélectionner un compte ;</li> <li>3. Le code PIN est incorrect (moins de trois fois) : l'utilisateur est invité à réintroduire son code PIN ;</li> <li>4. Le code PIN est incorrect pour la troisième fois : un message d'erreur s'affiche et le compte de l'utilisateur est bloqué.</li> </ol>
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Read</i> de la fenêtre <i>QR codes</i>
Fréquence d'utilisation	Rare

TABLE 5.2 – Effectuer un paiement via QR code



Acteur	Client, institution
Description	Permet à l'utilisateur de visualiser les paiements à effectuer situés dans le fichier JSON dédié
Préconditions	L'utilisateur est authentifié et possède une liste de paiements à effectuer au format JSON (qui peut être vide)
Postconditions	La liste de paiements est disponible en mémoire
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur arrive sur la fenêtre ;</li> <li>2. Si la liste de paiements n'est pas déjà disponible en mémoire, la lecture du fichier JSON dédié est effectuée ;</li> <li>3. La liste des paiements est affichée dans la liste.</li> </ol>
Scénario alternatif	Aucun
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Due payments</i> de la fenêtre <i>Main screen</i>
Fréquence d'utilisation	Moyenne

TABLE 5.3 – Voir la liste des paiements à effectuer

Acteur	Client
Description	Permet à l'utilisateur de sélectionner un paiement à effectuer et d'utiliser les boutons <i>Pay</i> et <i>Pay contactless</i> de la fenêtre <i>Due payments</i>
Préconditions	L'utilisateur est authentifié, possède une liste de paiements à effectuer au format JSON et en a sélectionné un (la liste lue dans le fichier JSON ne peut donc pas être vide)
Postconditions	Un paiement est sélectionné
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur est déjà en train de visualiser la liste de paiements ;</li> <li>2. L'utilisateur clique sur un paiement de la liste ;</li> <li>3. Un objet en mémoire est modifié pour savoir quel paiement est actuellement sélectionné par l'utilisateur.</li> </ol>
Scénario alternatif	Aucun
Trigger	Lorsque l'utilisateur clique sur un des paiements de la liste <i>Due payments</i> de la fenêtre <i>Due payments</i>
Fréquence d'utilisation	Élevée

TABLE 5.4 – Sélectionner un paiement à effectuer

Acteur	Client, institution
Description	Permet à l'utilisateur de réaliser le paiement précédemment sélectionner "sans contact"
Préconditions	L'utilisateur est authentifié, possède une liste de paiements à effectuer au format JSON, en a sélectionné un (la liste lue dans le fichier JSON ne peut donc pas être vide), a sélectionné un compte duquel débiter l'argent et a sélectionné le moyen de paiement "sans contact"
Postconditions	Le paiement a été effectué : le compte sélectionné est débité et le destinataire reçoit l'argent
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le bouton <i>Pay contactless</i> ;</li> <li>2. Un appel serveur est réalisé afin de savoir si effectuer cette transaction ne dépassera pas le seuil maximal imposé par la banque et si le montant d'argent disponible sur le compte sélectionné est suffisant ;</li> <li>3. Une nouvelle transaction est envoyée au serveur ;</li> <li>4. La transaction est enregistrée dans la base de données ;</li> <li>5. Un transfert d'argent est effectué.</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. Le montant à payer est supérieur au montant contenu dans le compte sélectionné : un message d'erreur s'affiche, la sélection du menu déroulant est réinitialisée et l'utilisateur peut à nouveau sélectionner une méthode de paiement ;</li> <li>2. Le seuil maximal d'argent transféré "sans contact" est dépassé : un message d'erreur s'affiche et invite le client à effectuer la transaction "avec contact".</li> </ol>
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Pay contactless</i> de la fenêtre <i>Due payments</i> en ayant sélectionné un paiement à effectuer
Fréquence d'utilisation	Moyenne

TABLE 5.5 – Payer sans contact

Acteur	Client, institution
Description	Permet à l'utilisateur de réaliser le paiement précédemment sélectionner "avec contact"
Préconditions	L'utilisateur est authentifié, possède une liste de paiements à effectuer au format JSON, en a sélectionné un et a sélectionné le moyen de paiement "avec contact"
Postconditions	Le paiement a été effectué : le compte sélectionné est débité et le destinataire reçoit l'argent
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le bouton <i>Pay</i> ;</li> <li>2. Un appel serveur demandant le montant disponible sur le compte sélectionné est réalisé afin de vérifier si l'utilisateur peut payer à partir du compte sélectionné ;</li> <li>3. L'utilisateur est invité à entrer son code PIN ;</li> <li>4. Une nouvelle transaction est envoyée au serveur ;</li> <li>5. La transaction est enregistrée dans la base de données ;</li> <li>6. Un transfert d'argent est effectué.</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. Le montant à payer est supérieur au montant contenu dans le compte sélectionné : un message d'erreur s'affiche, la sélection du menu déroulant est réinitialisée et l'utilisateur peut à nouveau sélectionner un compte ;</li> <li>2. Le code PIN est incorrect (moins de trois fois) : l'utilisateur est invité à réintroduire son code PIN ;</li> <li>3. Le code PIN est incorrect pour la troisième fois : un message d'erreur s'affiche et le compte de l'utilisateur est bloqué.</li> </ol>
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Pay contactless</i> de la fenêtre <i>Due payments</i> en ayant sélectionné un paiement à effectuer
Fréquence d'utilisation	Moyenne

TABLE 5.6 – Payer avec contact

Acteur	Client, serveur
Description	Permet au client de visualiser les transactions marquées comme suspectes dans la base de donnée
Préconditions	L'utilisateur est authentifié
Postconditions	La liste des transactions suspectes est disponible en mémoire
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur arrive sur la fenêtre ;</li> <li>2. Un appel serveur est effectué afin de récupérer les transactions suspectes ;</li> <li>3. La liste des transactions suspecte est affichée dans la liste.</li> </ol>
Scénario alternatif	Aucun
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Suspicious transactions</i> de la fenêtre <i>Main screen</i>
Fréquence d'utilisation	Moyenne

TABLE 5.7 – Voir l'historique des transactions suspectes

Acteur	Client, serveur
Description	Permet à l'utilisateur de confirmer qu'une transaction suspecte n'était pas intentionnelle
Préconditions	L'utilisateur est authentifié et a sélectionné une suspicion à valider
Postconditions	La suspicion est confirmée
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le bouton <i>Suspicious</i> ;</li> <li>2. La transaction est marquée comme suspecte dans la base de données ;</li> <li>3. Localement, on calcule le niveau de dangerosité qui détermine comment augmenter le niveau de sécurité du compte de l'utilisateur (typiquement, plus le niveau de sécurité est élevé, plus il faudra pouvoir confirmer son identité et plus il sera compliqué d'utiliser le compte de manière frauduleuse).</li> </ol>
Scénario alternatif	Aucun
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Suspicious</i> de la fenêtre <i>Suspicious transactions history</i>
Fréquence d'utilisation	Moyenne

TABLE 5.8 – Confirmer une suspicion de fraude

Acteur	Client, serveur
Description	Permet à l'utilisateur de confirmer qu'une transaction suspecte était intentionnelle
Préconditions	L'utilisateur est authentifié et a sélectionné une suspicion à infirmer
Postconditions	La suspicion est infirmée
Scénario principal	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le bouton <i>Intentional</i> ;</li> <li>2. La transaction est marquée comme non suspecte dans la base de données ;</li> <li>3. Localement, on calcule le niveau de dangerosité qui détermine comment diminuer le niveau de sécurité du compte de l'utilisateur (typiquement, plus le niveau de sécurité est faible, moins il faudra pouvoir confirmer son identité et plus il y aura de fonctionnalités "à risque" (comme les paiements sans contact) qui seront utilisables).</li> </ol>
Scénario alternatif	Aucun
Trigger	Lorsque l'utilisateur clique sur le bouton <i>Intentional</i> de la fenêtre <i>Suspicious transactions history</i>
Fréquence d'utilisation	Moyenne

TABLE 5.9 – Infirmer une suspicion de fraude

Acteur	Client, serveur
Description	Permet à l'utilisateur de recevoir une notification indiquant qu'une transaction suspecte a été effectuée
Préconditions	Une transaction a été réalisée
Postconditions	L'utilisateur est notifié et il a accès à la notification dans la fenêtre <i>Notifications</i>
Scénario principal	<ol style="list-style-type: none"> <li>1. Une transaction a lieu ;</li> <li>2. Le serveur calcule la suspicion de fraude de la transaction selon les règles de calcul établies et ce résultat dépasse un certain seuil ;</li> <li>3. Une notification est envoyée au client ;</li> <li>4. La transaction est marquée comme suspecte et l'utilisateur peut examiner cette suspicion, la confirmer ou l'infirmer.</li> </ol>
Scénario alternatif	<ol style="list-style-type: none"> <li>1. La transaction n'est pas suspecte : le reste de la procédure n'a pas lieu</li> </ol>
Trigger	Lorsqu'une transaction a lieu et est suspecte
Fréquence d'utilisation	Très élevée

TABLE 5.10 – Notifier le client d'une transaction suspecte

## 5.1.2 Interaction overview diagram

### Nouvelles références :

- Créer une demande de paiement via QR code ;
- Effectuer un paiement via QR code ;
- Voir l'historique des transactions suspectes ;
- Infirmer une suspicion de fraude ;
- Confirmer une suspicion de fraude ;
- Voir les paiements à effectuer ;
- Sélectionner un paiement à effectuer ;
- Payer avec contact ;
- Payer sans contact.

**Créer une demande de paiement via QR code** L'utilisateur a la possibilité de créer une image contenant un QR code, qui pourra transmettre des données (montant, destinataire et message) et qui pourra plus tard être utilisé par un autre utilisateur afin d'effectuer une transaction correspondant aux données.

**Effectuer un paiement via QR code** L'utilisateur peut aussi sélectionner une image dans ses fichiers locaux contenant un QR code créé par un autre utilisateur auparavant. Ce QR code pourra alors être lu et une transaction pourra avoir lieu.

**Voir l'historique des transactions suspectes** Le client peut visualiser l'ensemble des transactions qui auront été marquées comme suspectes dans la base de données. Par la suite, il pourra confirmer ou infirmer cette suspicion.

**Infirmer une suspicion de fraude** Le client ayant accédé aux transactions suspectes, il pourra infirmer cette suspicion et baisser le niveau de sécurité de son compte. Typiquement, plus ce niveau est élevé, plus l'utilisateur est restreint, afin d'empêcher toute utilisation frauduleuse de son compte.

**Confirmer une suspicion de fraude** Le client ayant accédé aux transactions suspectes, il pourra confirmer cette suspicion et augmenter le niveau de sécurité de son compte.

**Voir les paiements à effectuer** L'utilisateur peut accéder à une liste de paiements prévus, situés dans un fichier local au format JSON. Cette liste de paiements peut être réglée, moyennant un ou plusieurs paiements avec ou sans contact.

**Sélectionner un paiement à effectuer** L'utilisateur peut sélectionner un paiement parmi ceux proposés dans la liste, afin d'effectuer la transaction.

**Payer avec contact** Après avoir sélectionné le paiement qu'il souhaite effectuer, l'utilisateur a la possibilité de payer avec contact, ce qui signifie qu'il devra entrer son code PIN afin de confirmer le paiement.

**Payer sans contact** Après avoir sélectionner le paiement qu'il souhaite effectuer, l'utilisateur a la possibilité de payer sans contact, ce qui signifie que le paiement sera effectuer sans avoir besoin d'entrer le code PIN. Ce moyen de paiement sera soumis à plus de restrictions (notamment une limite, comme explicité pour l'*Interaction Overview Diagram* de l'application institution) afin d'éviter les utilisations frauduleuses.

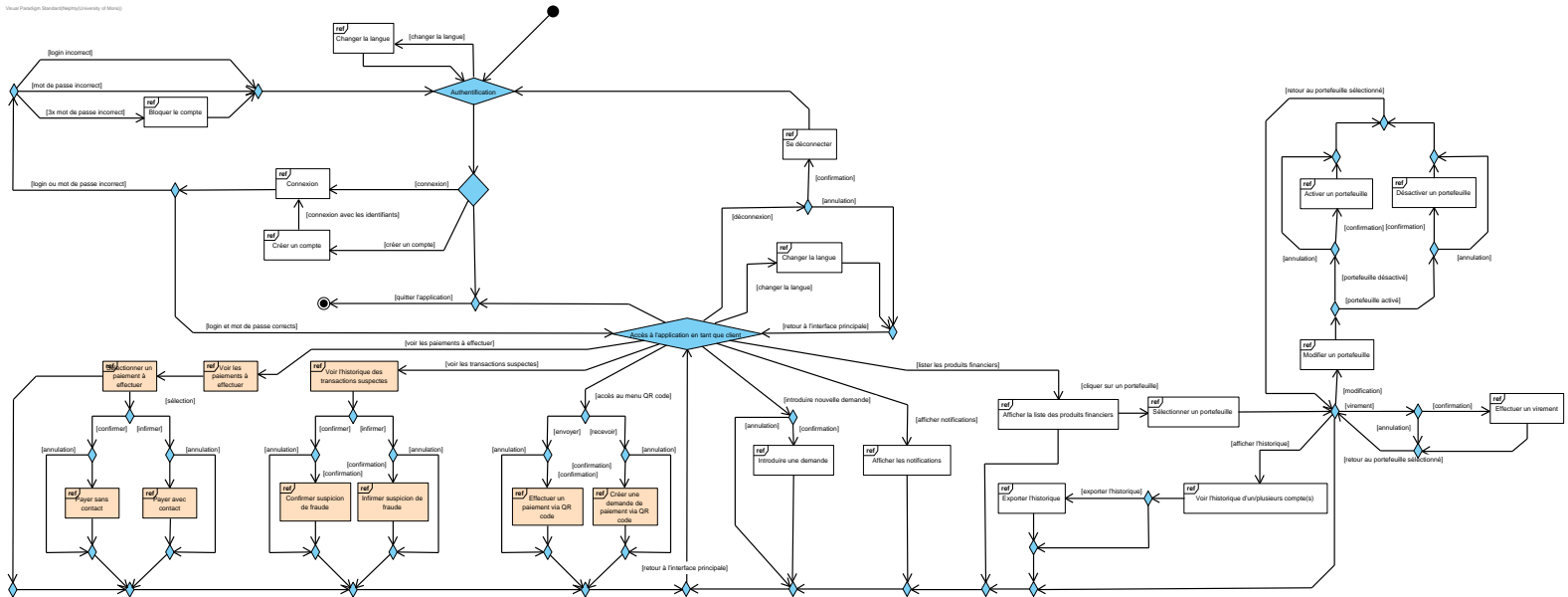


FIGURE 5.2 – Nouveau diagramme d'interaction de l'application client

### 5.1.3 Class diagram

**Structure** Le diagramme de classe de l'application client suit globalement la même structure que celui de la partie commune. Des ajouts ont été faits au GUI : de nouvelles fenêtres requises par les nouvelles fonctionnalités sont disponibles et respectent toujours le design pattern Singleton. Un utilitaire (classe *QRCode*) a été ajouté à la partie logique, ainsi qu'un ensemble d'attributs qui serviront notamment à la détection de fraude et à l'imposition de limites par l'institution financière. Trois énumérations ont aussi été ajoutées : *SEPACountries*, *Country* et *State*, servant respectivement à identifier les pays faisant partie de la zone SEPA, les pays à partir desquels on peut effectuer une transaction et l'état d'une transaction. Quelques méthodes ont été ajoutés à l'API afin d'obtenir certaines données plus facilement, sans devoir implémenter de nouvelles méthodes dans le code de l'application.

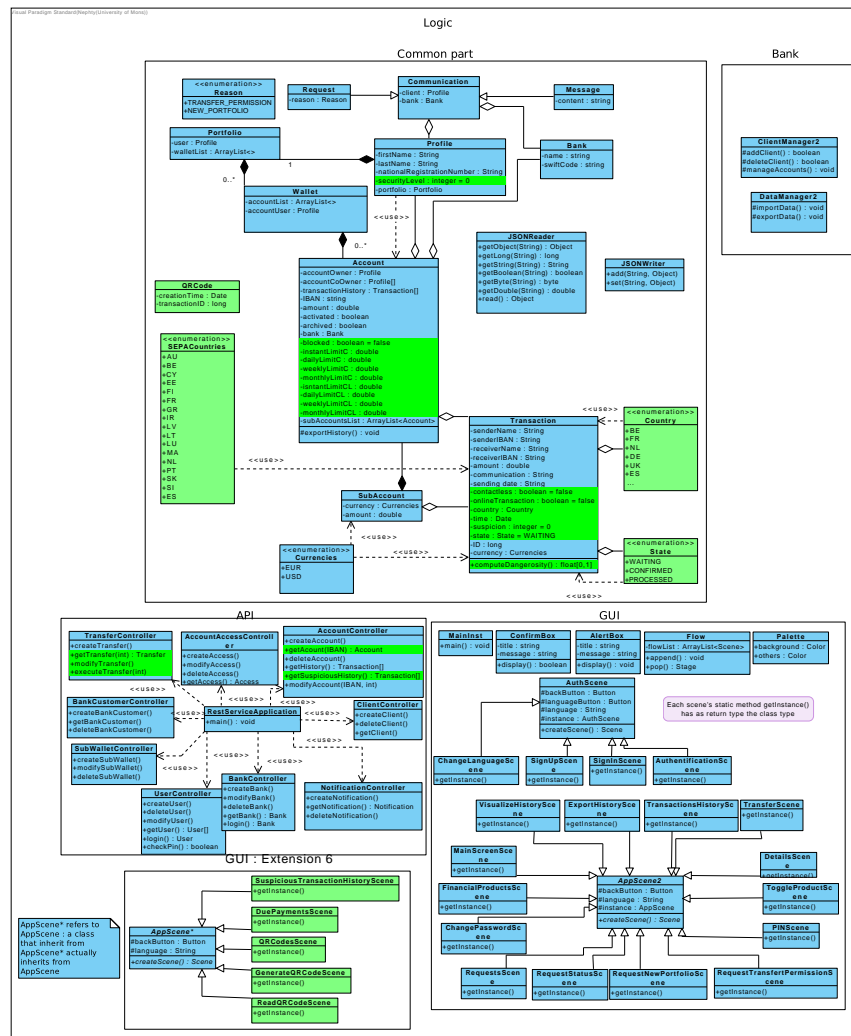


FIGURE 5.3 – Nouveau diagramme de classes de l'application client



### 5.1.4 Sequence diagram

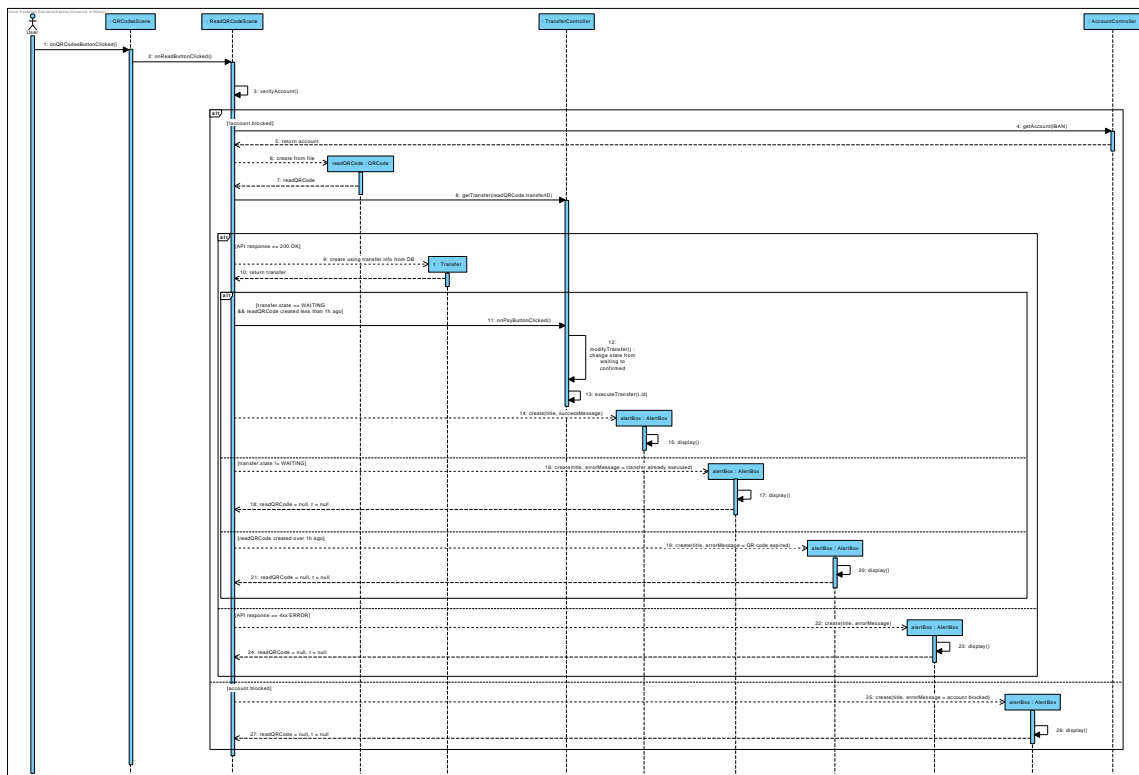


FIGURE 5.4 – Effectuer un paiement via QR code

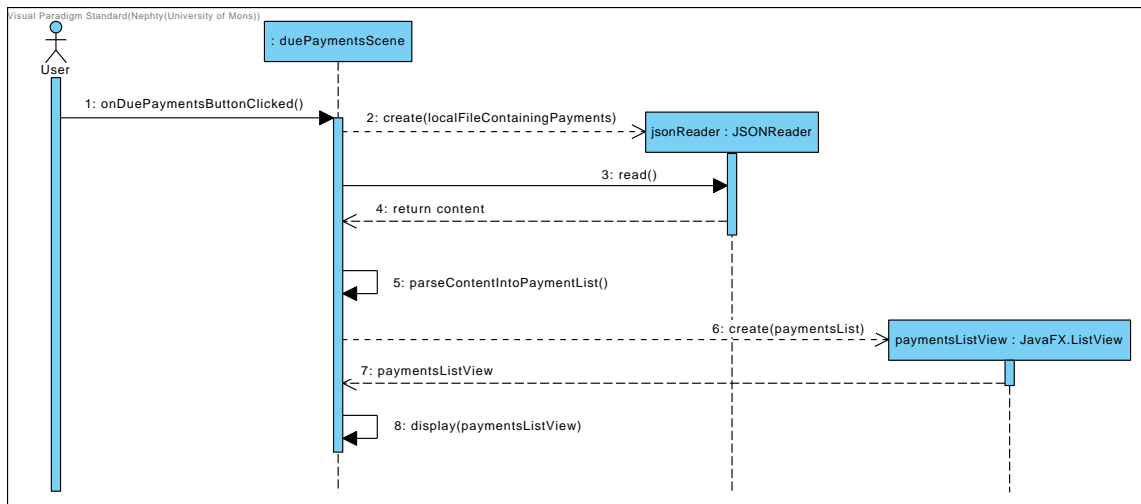


FIGURE 5.5 – Voir la liste des paiements à effectuer

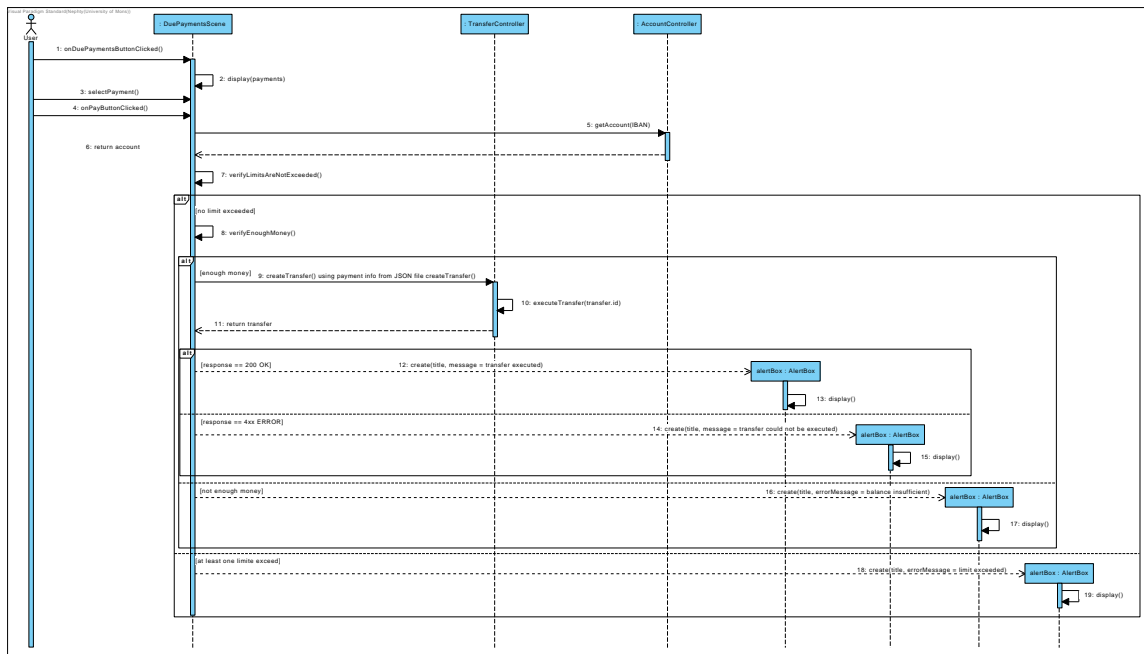


FIGURE 5.6 – Payer sans contact

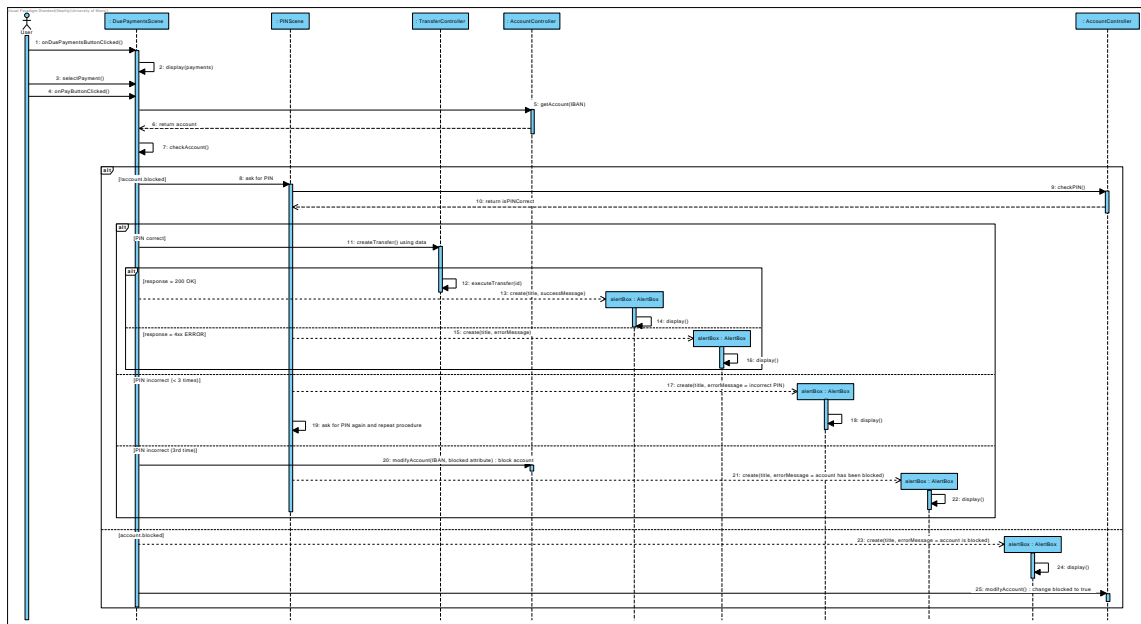


FIGURE 5.7 – Payer avec contact

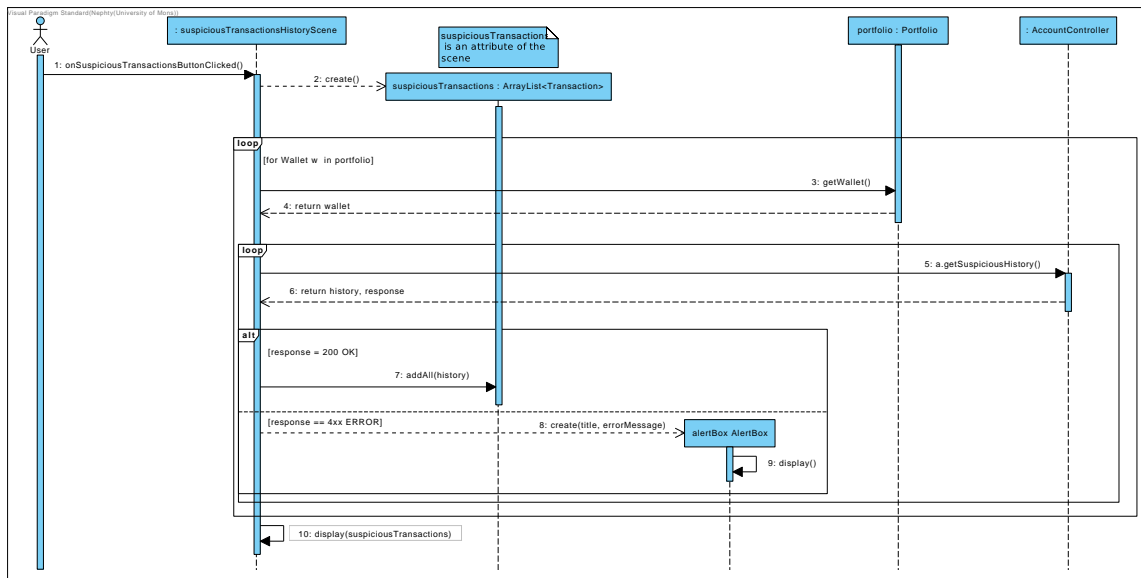


FIGURE 5.8 – Voir l’historique des transactions suspectes

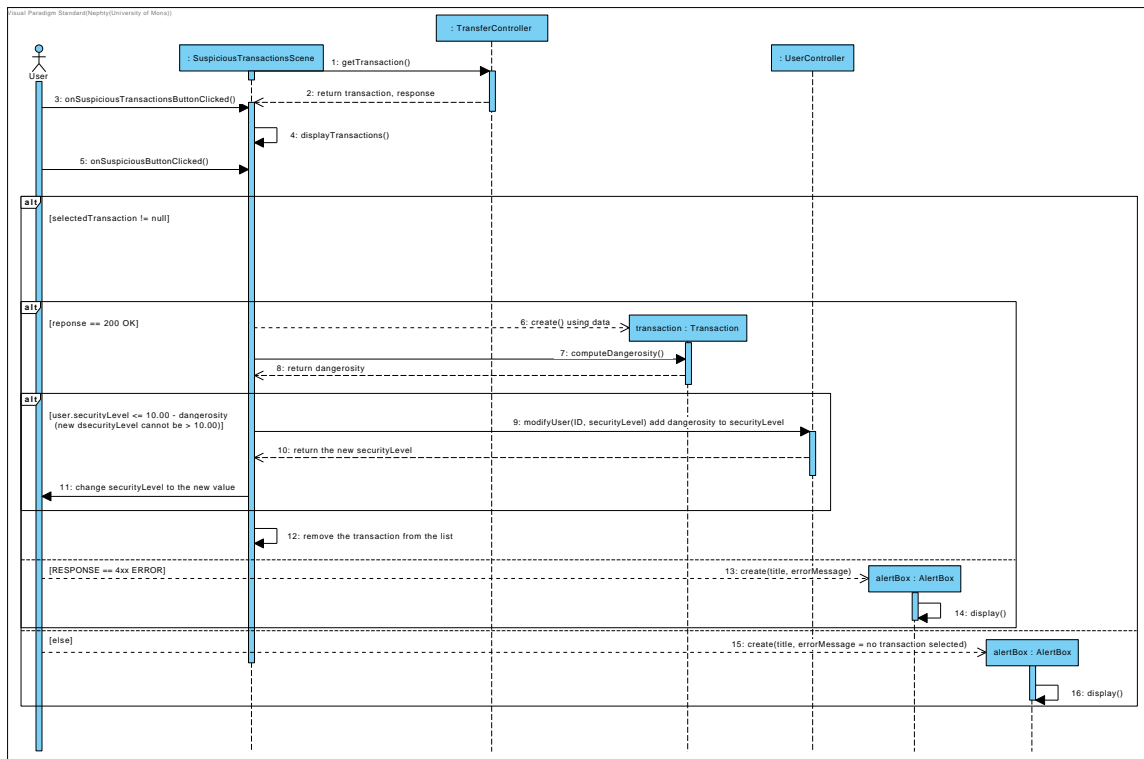


FIGURE 5.9 – Confirmer une suspicion de fraude

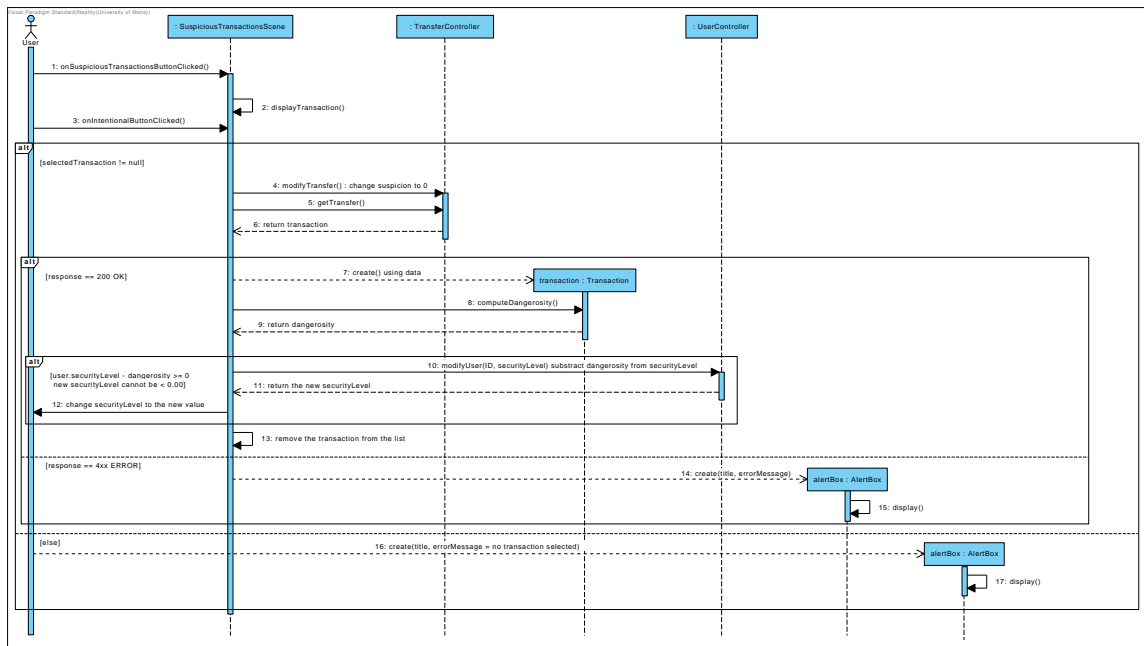


FIGURE 5.10 – Infirmer une suspicion de fraude

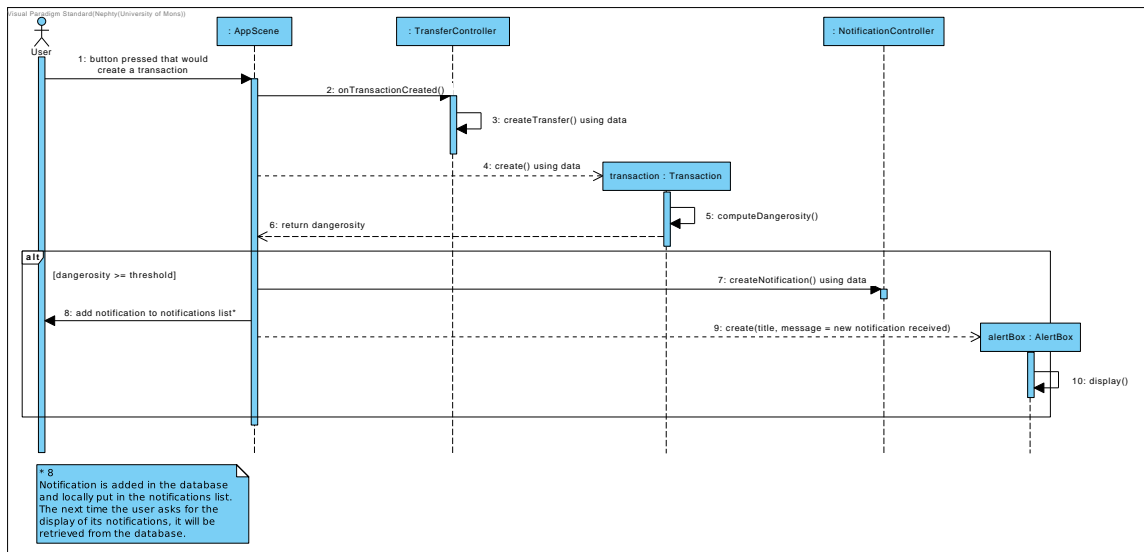
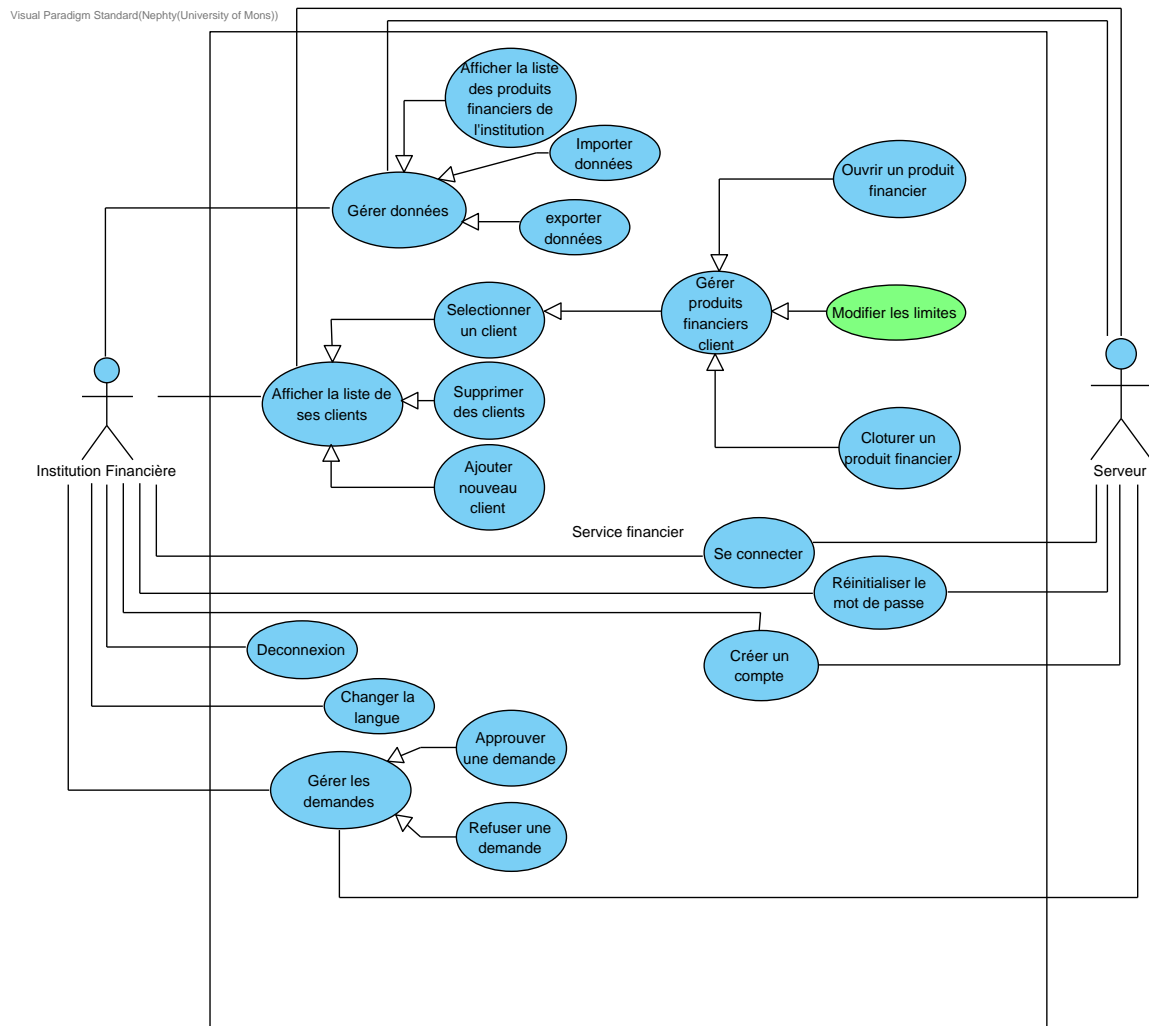


FIGURE 5.11 – Notifier le client d’une transaction suspecte



## 5.2 Diagrammes de conception UML : application institution

### 5.2.1 Use case diagram



Acteur	Serveur
Description	Permet de bloquer un compte de l'utilisateur
Préconditions	Une action (peu importe sa nature) ayant pour conséquence de bloquer le compte d'un utilisateur a eu lieu
Postconditions	Le compte est bloqué
Scénario principal	<ol style="list-style-type: none"> <li>1. Une notification est envoyée au client ;</li> <li>2. Le compte en question est désormais inutilisable.</li> </ol>
Scénario alternatif	Aucun
Trigger	Lorsqu'une action demandant à bloquer un compte a lieu
Fréquence d'utilisation	Très rare

TABLE 5.11 – Bloquer un compte d'un client

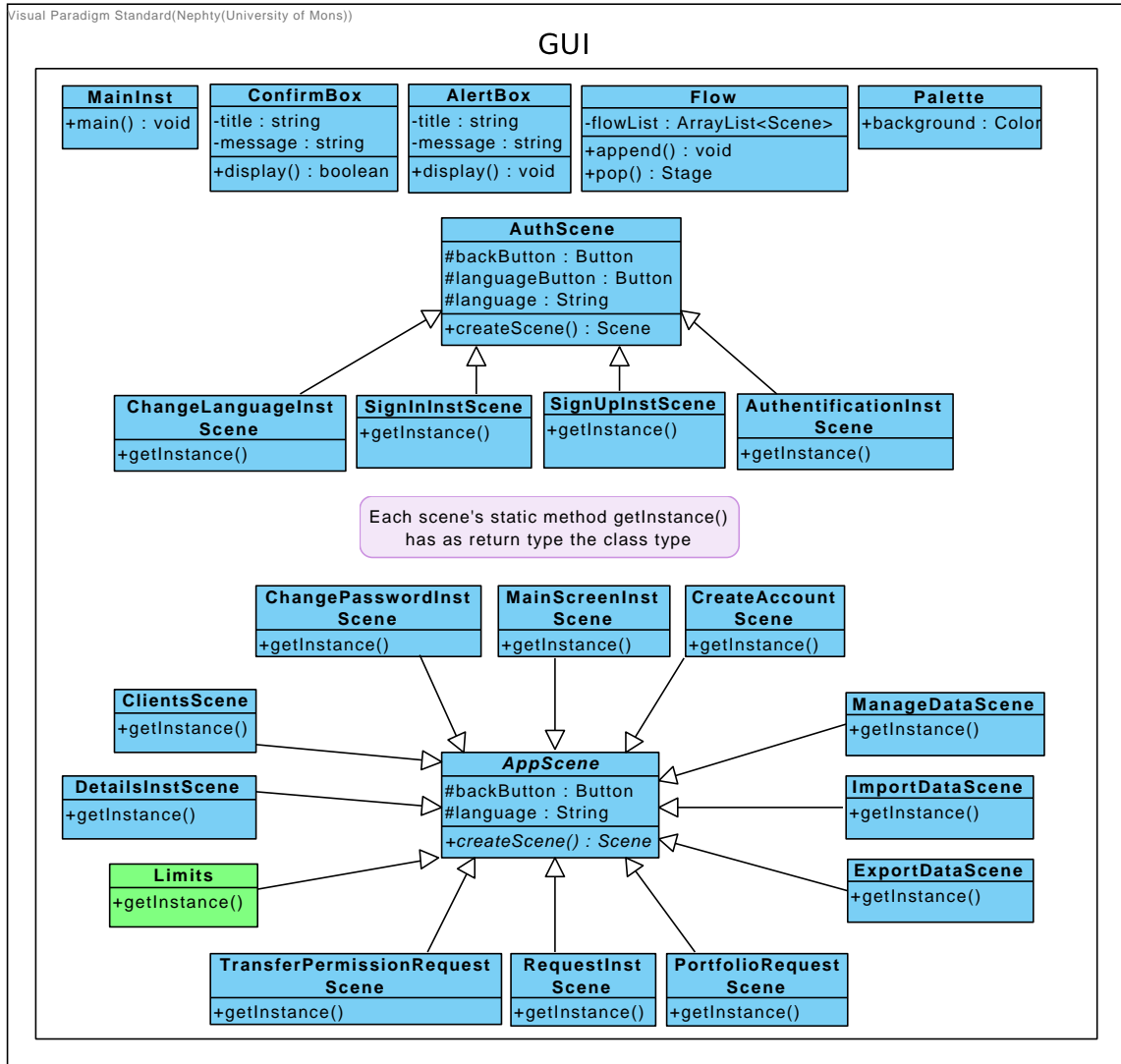
Acteur	Serveur
Description	Permet de bloquer un compte de l'utilisateur
Préconditions	Une action (peu importe sa nature) ayant pour conséquence de bloquer le compte d'un utilisateur a eu lieu
Postconditions	Le compte est bloqué
Scénario principal	<ol style="list-style-type: none"> <li>1. Une notification est envoyée au client ;</li> <li>2. Le compte en question est désormais inutilisable.</li> </ol>
Scénario alternatif	Aucun
Trigger	Lorsqu'une action demandant à bloquer un compte a lieu
Fréquence d'utilisation	Très rare

TABLE 5.12 – Modifier les limites



### 5.2.3 Class diagram

**Structure** Le diagramme de classe de l'institution est très légèrement modifié : une classe supplémentaire correspondant à une nouvelle fenêtre requise est ajoutée, toujours en respectant le design pattern Singleton. Pour rappel, la partie logique de l'application est commune avec celle de l'application client et peut être trouvée plus haut.



#### **5.2.4 Sequence diagram**

## 5.3 Modèle de données

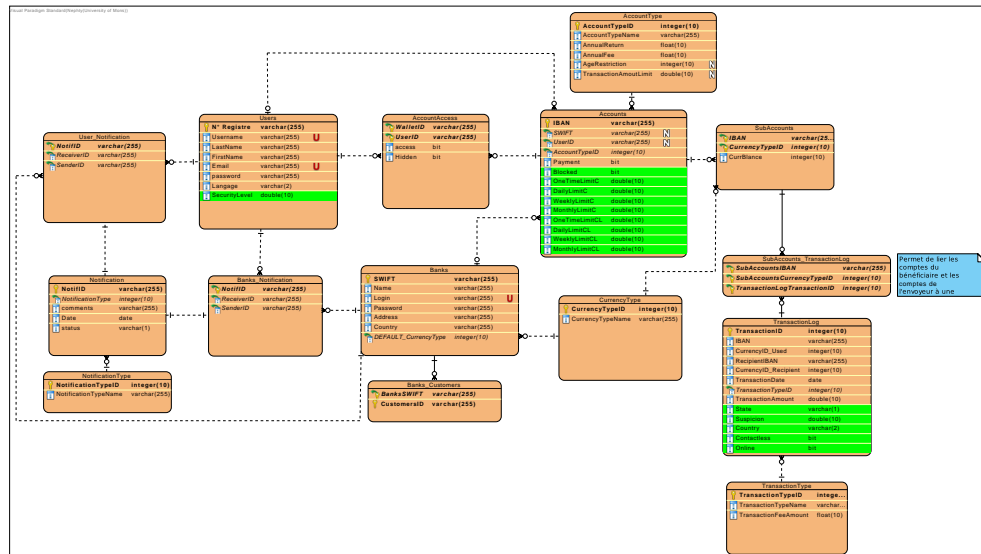
**Nouveaux attributs** La base de données possède certains attributs afin de contribuer au bon fonctionnement de cette extension.

**Table Users** Cette table possède un attribut *SecurityLevel* qui décrit le niveau de sécurité du compte de l'utilisateur. Typiquement, plus le niveau de sécurité sera élevé, plus l'utilisateur sera restreint ou devra confirmer son identité durant les transactions (par exemple, les limites de paiement sans contact seront plus basses, voire les paiements sans contact seront totalement bloqués).

**Table Account** Cette table possède un attribut *Blocked* qui indique si le compte est bloqué ou non, ainsi que huit attributs "limite" qui décrivent les limites instantanées, journalières, hebdomadaires et mensuelles des paiements avec et sans contact (limites distinctes).

**Table TransactionLog** Cette table possède cinq attributs : *State* qui décrit l'état d'une transaction (en attente, confirmée et traitée), *Suspicion* qui indique le niveau de suspicion de la transaction (une indication du taux de probabilité que cette transaction soit frauduleuse), *Country* qui indique le pays à partir duquel la transaction a été effectuée, *Contactless* qui indique si la transaction a été réalisée sans contact ou non et *Online* qui indique si la transaction a été réalisée dans un magasin en ligne.

**Table Notification** Cette table possède un attribut *Permanent* qui, s'il est à 1 (vrai), empêche de supprimer la notification de la liste des notifications. Grâce à cette stratégie, nous pouvons évaluer la dangerosité d'une transaction de manière locale (sans devoir évaluer la dangerosité d'une transaction chaque fois que le serveur en reçoit une), puis envoyer une notification au client à qui le compte à partir duquel la transaction a été réalisée appartenait. De cette manière, une personne usurpant le compte d'une autre personne et effectuant une transaction suspecte ne pourra pas supprimer la notification de la liste des notifications. La notification de suspicion sera supprimée lorsque l'utilisateur aura confirmé ou infirmé la suspicion.



## 5.4 Maquette de l'interface utilisateur

### Nouvelles fenêtres disponibles

- QR codes ;
- Generate QR code ;
- Read QR code ;
- Suspicious transactions history ;
- Due payments.

### Fenêtres modifiées

- Main screen.

#### Fenêtre *QR codes*

**Accès** : en cliquant sur le bouton *QR codes* de la fenêtre *Main screen*.

**Contenu** : les boutons *Back*, *Generate* et *Read*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- Le bouton *Generate* : envoie l'utilisateur sur la fenêtre *Generate QR code* ;
- Le bouton *Read* : envoie l'utilisateur sur la fenêtre *Read QR code*.

#### Fenêtre *Generate QR code*

**Accès** : en cliquant sur le bouton *Generate* de la fenêtre *QR code*.

**Contenu** : les boutons *Back*, *Choose path...* et *Generate*, les labels *Amount*, *Message*, *Export location* et *Selected path* et les champs de texte *Amount* et *Message*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *QR codes* ;
- Le bouton *Choose path...* : ouvre l'explorateur de fichier afin de déterminer le chemin de destination ;
- Le bouton *Generate* : génère un QR code si le montant est un nombre supérieur à zéro, l'IBAN est correctement formaté et la destination est choisie ;
- Le champ de texte *Amount* : permet à l'utilisateur d'entrer le montant ;
- Le champ de texte *Message* : permet à l'utilisateur d'entrer la communication.

#### Fenêtre *Read QR code*

**Accès** : en cliquant sur le bouton *Read* de la fenêtre *QR codes*.

**Contenu** : les boutons *Back*, *Choose file...* et *Pay*, les labels *Choose file*, *Selected file*, *Account* et *QR code state* et le menu déroulant *Account*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *QR codes* ;
- Le bouton *Choose file...* : ouvre l'explorateur de fichier afin de déterminer le fichier à lire ;
- Le bouton *Pay* : effectue le paiement si le QR code est valide ;
- Le menu déroulant *Account* : permet à l'utilisateur de sélectionner le compte à partir duquel retirer l'argent.

#### Fenêtre *Suspicious transactions history*

**Accès** : en cliquant sur le bouton *Suspicious transactions* de la fenêtre *Main screen*.

**Contenu** : les boutons *Back*, *Intentional* et *Suspicious*, le label *Choose a suspicion* et la liste *Suspicious transaction history*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;

- Le bouton *Intentional* : permet à l'utilisateur de marquer une transaction comme intentionnelle et d'infirmar la suspicion ;
- Le bouton *Suspicious* : permet à l'utilisateur de marquer une transaction comme suspecte et de confirmer la suspicion ;
- La liste *Suspicious transaction history* : contient tout l'historique des transactions marquées comme suspectes.

#### **Fenêtre *Due payments***

**Accès** : en cliquant sur le bouton *Due payments* de la fenêtre *Main screen*.

**Contenu** : les boutons *Back*, *Pay* et *Pay contactless*, les labels *Choose a payment* et *Choose an account*, le menu déroulant *Account* et la liste *Due payments*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Main screen* ;
- Le bouton *Pay* : permet à l'utilisateur de payer "avec contact" avec le compte sélectionné, c'est-à-dire entrer son compte PIN afin de valider la transaction ;
- Le bouton *Pay contactless* : permet à l'utilisateur de payer "sans contact" avec le compte sélectionné, c'est-à-dire valider la transaction sans entrer son code PIN, à condition que la limite ne soit pas dépassée ;
- Le menu déroulant *Account* : permet à l'utilisateur de sélectionner le compte à partir duquel effectuer le paiement ;
- La liste *Due payments* : affiche tous les paiements trouvés dans le fichier dédié et permet à l'utilisateur d'en sélectionner un à effectuer.

#### **Fenêtre *Main screen***

**Nouvel accès** : en cliquant sur le bouton *Back* des fenêtres *QR codes*, *Suspicious transactions history* ou *Due payments*.

**Nouveau contenu** : les boutons *Due payments*, *Suspicious transactions* et *QR codes*.

- Le bouton *Due payments* : envoie l'utilisateur sur la fenêtre *Due payments* ;
- Le bouton *Suspicious transactions* : envoie l'utilisateur sur la fenêtre *Suspicious transactions history* ;
- Le bouton *QR codes* : envoie l'utilisateur sur la fenêtre *QR codes*.

**Illustration du diagramme** Par souci de clarté, ce diagramme est disponible en annexe du rapport.



## 5.5 Maquette de l'interface institution

### Nouvelles fenêtres disponibles

- Limits.

### Fenêtres modifiées

- Details.

#### Fenêtre *Limits*

**Accès** : en cliquant sur le bouton *Edit limits* de la fenêtre *Details*.

**Contenu** : les boutons *Back* et *Set limit*, les labels *Client name*, *Account*, *Time scope*, *Current limit* et *New limit*, le champ de texte *New limit...* et le menu déroulant *Time scope*.

- Le bouton *Back* : envoie l'utilisateur sur la fenêtre *Details* ;
- Le bouton *Set limit* : modifie la limite maximale d'argent qu'il est possible de transférer à l'échelle de temps choisie dans le menu déroulant ;
- Le menu déroulant *Time scope* : permet de choisir l'échelle de temps pour laquelle il faut modifier la limite (at once, daily, weekly ou monthly).

#### Fenêtre *Details*

**Nouvel accès** : en cliquant sur le bouton *Back* de la fenêtre *Limits*.

**Nouveau contenu** : le bouton *Edit limits*.

- Le bouton *Edit limits* : envoie l'utilisateur sur la fenêtre *Limits*.

**Illustration du diagramme** Par souci de clareté, ce diagramme est disponible en annexe du rapport.