



**GOLDEN BANK**

# **Gestion des comptes bancaires personnels**

**Document d'analyse**

**Réalisé par :**

**Rh a SAADE**

**Fady ZAKHARIA**

# Sommaire

<b>A - <u>INTRODUCTION</u></b>	<b>3</b>
<b>B - <u>MOBILE BANKING</u></b>	<b>3</b>
B.1 - DESCRIPTION :	3
B.2 - LES BESOINS DU CLIENT :	3
B.3 - LES PRINCIPALES FONCTIONNALITES :	3
<b>C - <u>LES INTERVENANTS DE L'APPLICATION</u></b>	<b>4</b>
<b>C - <u>LES TECHNOLOGIES UTILISEES</u></b>	<b>4</b>
<b>D - <u>LES FONCTIONNALITES DU SYSTEME</u></b>	<b>5</b>
<b>E - <u>CAS D'UTILISATION</u></b>	<b>10</b>
E.1 - LOGIN :	10
E.2 - TRANSFER DE COMPTE A COMPTE :	11
E.3 - MY PROFILE :	14
E.4 - ACCOUNTS :	15
E.5 - ATM/BRANCHES :	17
E.6 - PRODUCTS :	19
<b>F - <u>LE MODELE DES DONNEES</u></b>	<b>25</b>
F.1 - LE DIAGRAMME DE CLASSES	25
F.1.a - Classe Client	26
F.1.b - Classe Account	26
F.1.c - Classe Transaction	26
F.1.d - Classe Client Card	26
F.1.e - Classe Currency	27
F.1.f - Classe BankBranche	27
F.1.g - Classe ATMBranche	27
F.1.h - Classe Products	27
F.1.i - Classe Products_Loans	28
F.1.j - Classe Products_Cards	28
F.2 - LE DIAGRAMME DE BASE DES DONNEES	29
<b>G - <u>CONCLUSION</u></b>	<b>30</b>

# **A - Introduction**

Le secteur bancaire étant en pleine expansion, chaque banque cherche à optimiser sa relation avec les clients en leur rendant l'accès à leurs comptes plus facile.

Notre projet consiste à améliorer et faciliter l'accès du client à son compte grâce à une application mobile lui permettant d'effectuer des transferts entre compte et compte, de consulter le solde de chaque compte et d'exécuter plusieurs autres actions sans aller à la banque.

## **B - Mobile Banking**

### **|Description :**

L'application contiendra plusieurs fonctionnalités que la banque offre dans ces agences.

### **|Les besoins du client :**

Les clients cherchent de nos jours à avoir toute les informations à portée de main.

### **|Les principales fonctionnalités :**

L'application permettra à l'utilisateur de :

1. Consulter son/ses compte(s).
2. Effectuer un transfert de compte à un autre compte.
3. Consulter les 10 dernières transactions effectuées sur son compte.
4. Repérer les différentes branches de banque et ATM.

## C - Les intervenants de l'application.

**Client:** Ce sont les clients de la banque ayant effectué l'enregistrement à l'application.

**Public :** Ce sont d'abord les personnes qui n'ont pas de compte, et ensuite les personnes qui ne se sont pas enregistrées dans l'application.

**Administrateur :** Il est capable d'ajouter et de mettre à jour un client, de créer et d'annuler un compte ainsi que les branches.

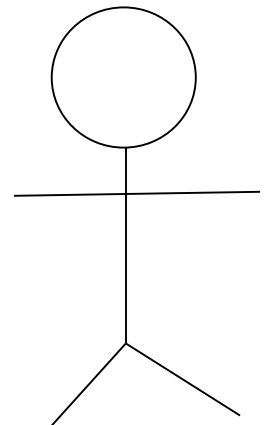
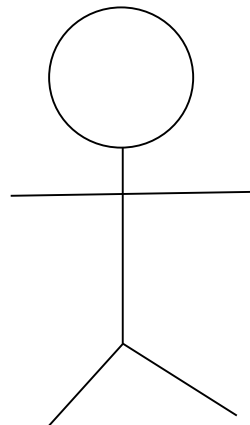
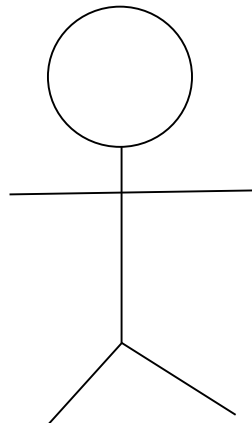
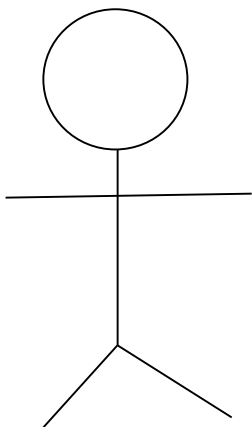
**Système :** Il est responsable de la gestion des données dans la base de données.

Système

Administrateur

Client

Public

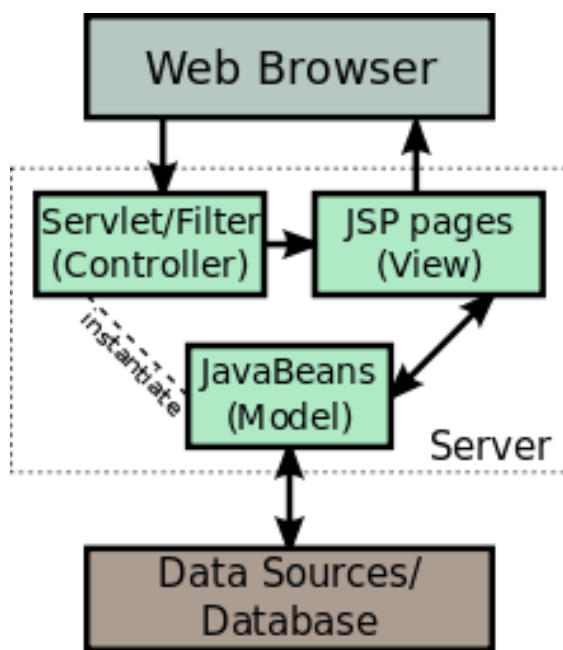


# D - Technologies utilisées

- 1- Back office : Netbeans 8.1 : Principe de programmation Servlet et JSP
- 2- Database : PostgreSQL
- 3- Glassfish server 4.1
- 4- Front office : Android Studio

## Le principe de programmation de Servlet + JSP :

- a- Ne jamais autoriser les utilisateurs à accéder directement au page JSP.
- b- JSP est uniquement considéré comme le lieu d'affichage de l'interface.
- c- Servlet agit en tant que contrôleur des flux de l'application et du traitement logique du programme.

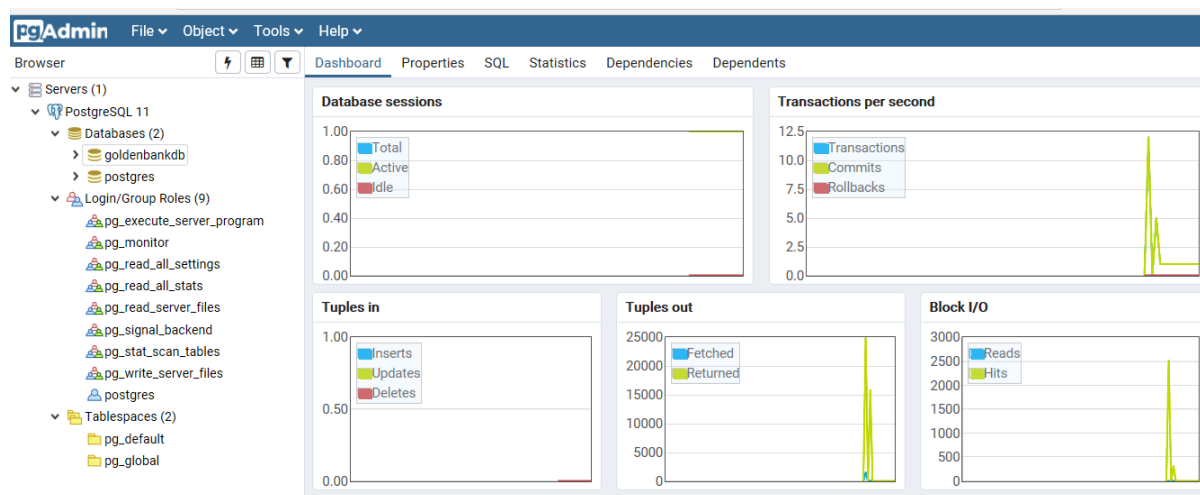


Lorsque l'utilisateur fait un « request » à un servlet, il en supprime les exigences, telles que l'insertion et la mise à jour, puis leur transmission à la page JSP pour afficher les données. Ainsi, chaque servlet a 0 ou plusieurs pages JSP correspondantes (en général, il ne faut que 1).

## Configuration du serveur PostgreSQL

Ouvrez les services Windows et vérifiez si le serveur PostgreSQL est en cours d'exécution. Le service s'appelle postgresql-x64-11. S'il est en cours d'exécution, ouvrez pgAdmin 4 - il devrait y avoir un shortcut sous le menu de PostgreSQL 10.

Une fois que vous êtes connecté à pgAdmin 4, vous devriez voir quelque chose comme ci-dessous :



Pour créer une nouvelle base de données, importer le fichier database.sql

(<https://github.com/banque-projetc2/GoldenBankAdmin>)

## Implémentation de la connexion JDBC Java PostgreSQL :

Maintenant que votre serveur de base de données est prêt, il est temps de vous y connecter. Lancez le projet Netbeans et ajouter une nouvelle librairie (postgresql-42.2.2.jar)

Ajouter les informations relatives à la connexion dans 'glassfish-ressource.xml' comme suit :

```
<resources>
<jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connect
<property name="serverName" value="localhost"/>
<property name="portNumber" value="5432"/>
<property name="databaseName" value="goldenbankdb"/>
<property name="User" value="postgres"/>
<property name="Password" value="rheasaade"/>
<property name="URL" value="jdbc:postgresql://localhost:5432/goldenbankdb"/>
<property name="driverClass" value="org.postgresql.Driver"/>
</jdbc-connection-pool>
<jdbc-resource enabled="true" jndi-name="goldenbankdb" object-type="user" pool-name="post-gre-s
</resources>
```

## Front office – Android studio :

Pour se connecter aux web services du back office, on a utilisé la librairie Volley de Android

Volley est une bibliothèque HTTP qui facilite et accélère la mise en réseau des applications Android. Il gère le traitement et la mise en cache des requêtes réseau et permet aux développeurs de gagner un temps précieux à l'écriture répétitive du même code d'appel / réseau du même réseau.



# E - Les fonctionnalités du système

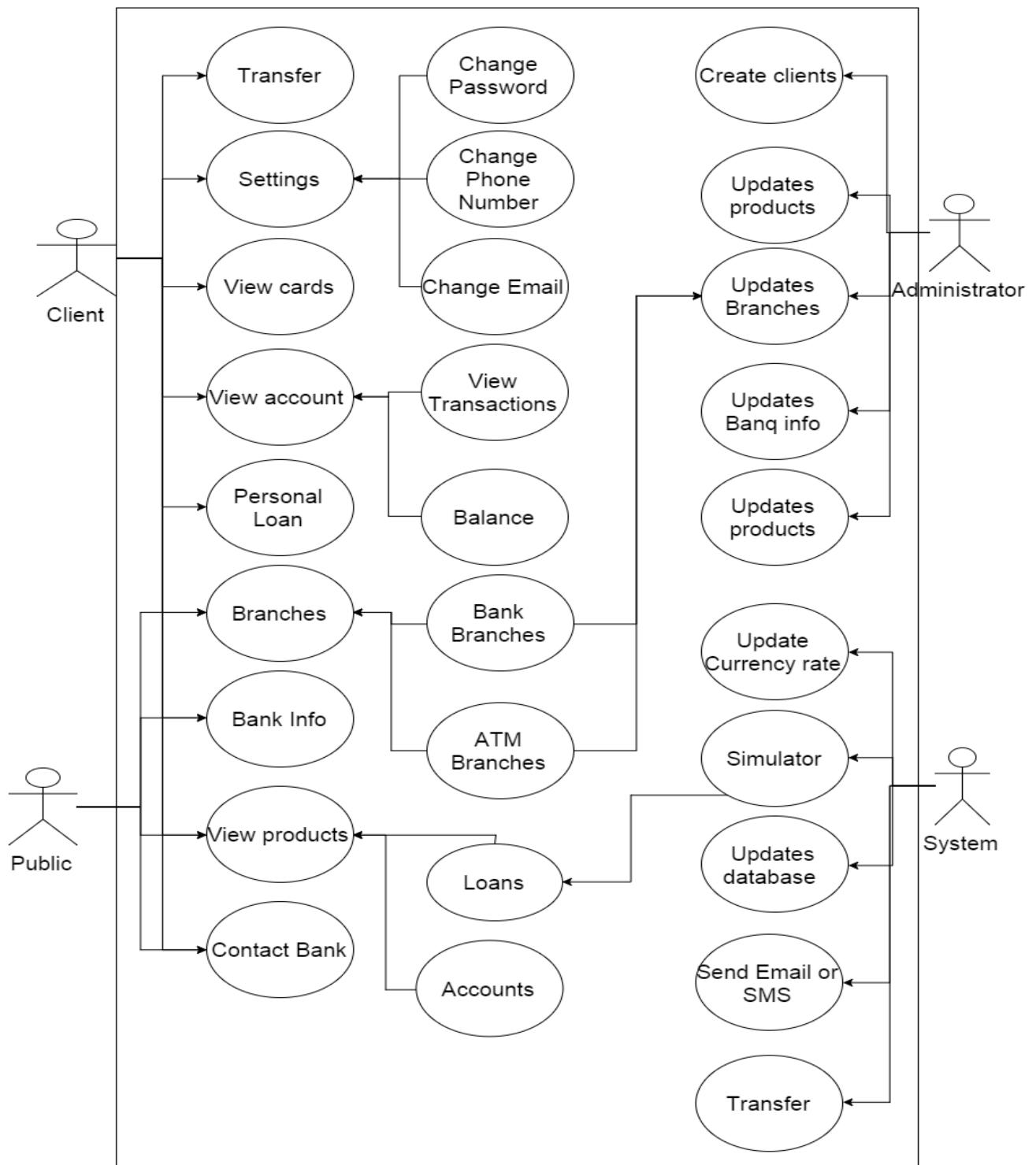
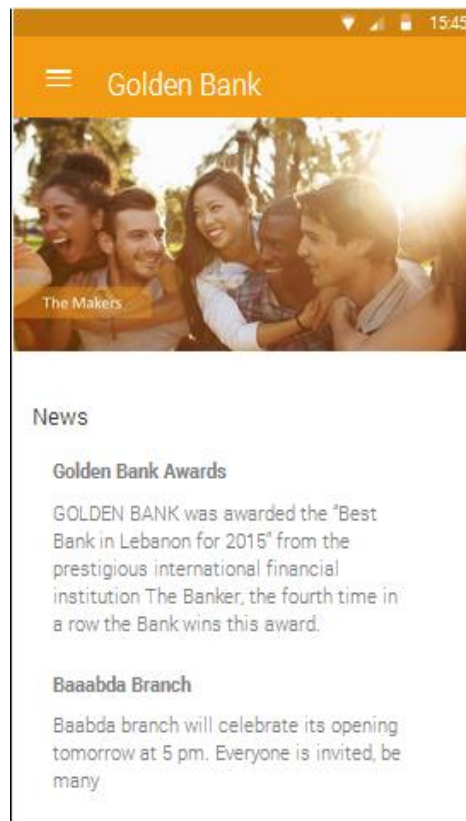


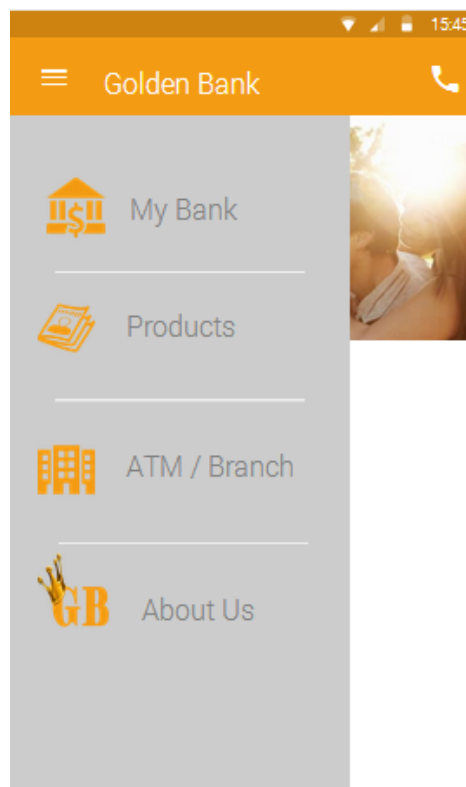
Figure 2: Le diagramme des cas d'utilisation



## Page Principale (Home Page) :



## Menu:



# F - Cas d'utilisation :

## Login :

**Objectif :** Accès au compte personnel.

**Acteurs :** Client, Système.

**Pré conditions :** Client doit être déjà enregistré dans la banque et a reçu son username et mot de passe par email/SMS.

### Scénario nominal :

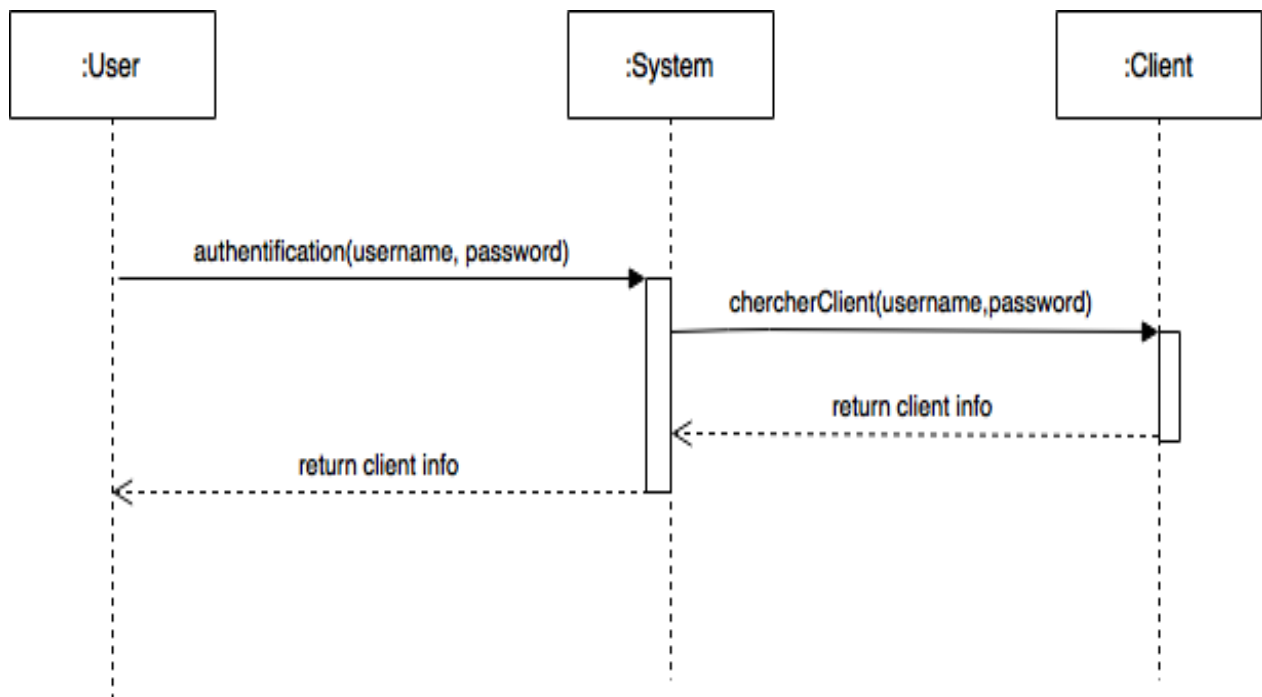
1-Le client écrit son username et mot de passe.

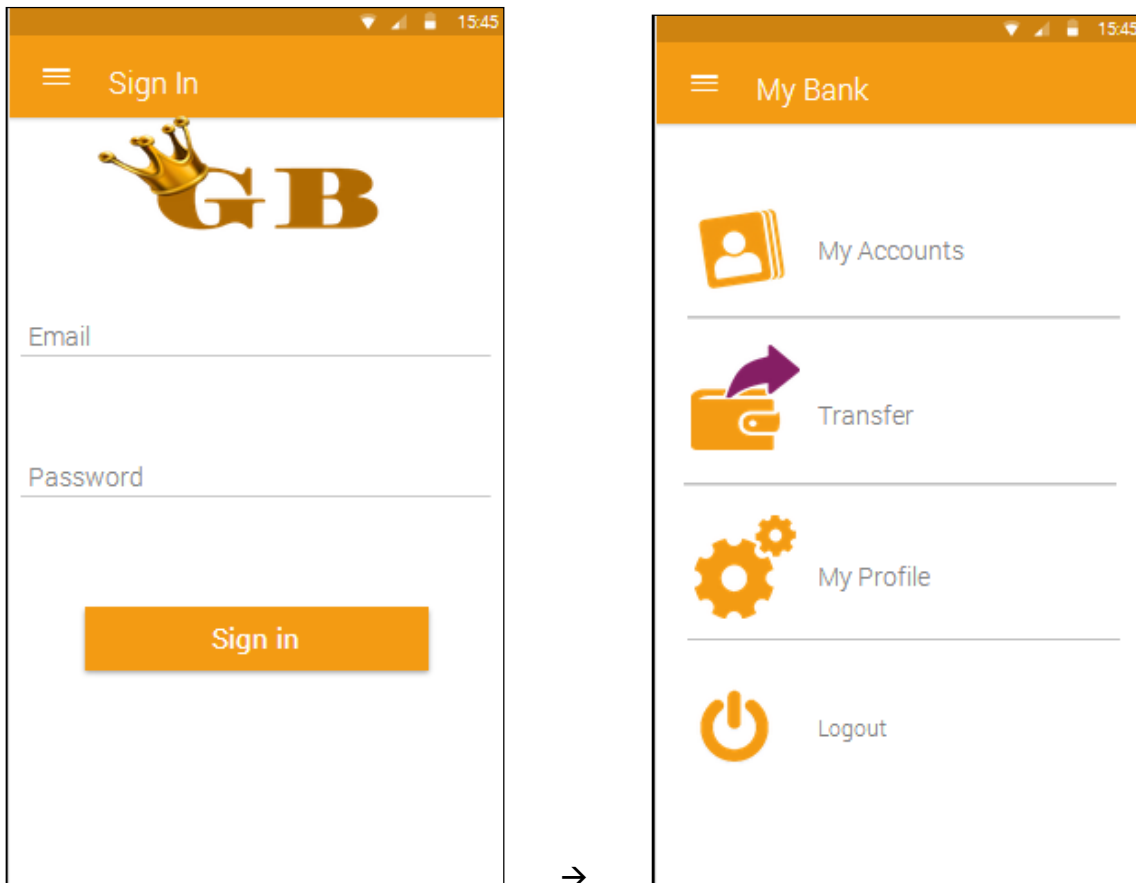
2- Le système le renvoi à la page « MyBank ».

### Scenario d'exception :

1-a- L'utilisateur inconnu ou mot de passe erroné, aller vers 1.

### Diagramme de séquence :





## **Transfer de Compte à Compte :**

**Objectif :** Transférer d'un compte à un autre.

**Acteurs :** Client, Système.

**Pré conditions :** Celui qui transfère doit être membre.

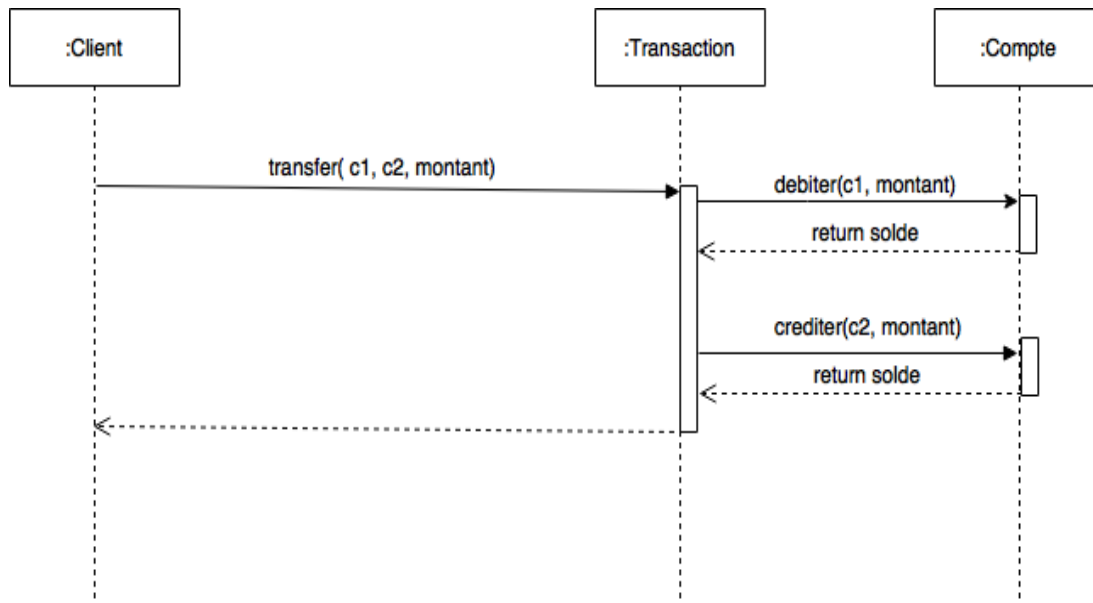
**Scénario nominal :**

- 1-Le Client s'authentifie
- 2-Le système affiche la première page.
- 3-Le client clique sur transfert.
- 4- Le système le renvoie à la page « Transfer ».
- 5-Le client sélectionne un de ces comptes (Drop Down List), le compte destination, et le montant, puis appuie sur transférer.
- 6- Le système effectue le transfert.

### Scenario d'exception :

- 1-a- L'utilisateur inconnu ou mot de passe erroné, aller vers 1.
- 5-a- Si compte de destination n'existe pas ou montant > solde, afficher4 message d'erreur puis aller vers 5.
- 5-b- Si différentes devise, calculer la différence puis aller vers 6.

### Diagramme de séquence :



The image shows a mobile application interface for a transfer screen. The top bar is orange with a menu icon (three horizontal lines) and the word "Transfer". Below the header, there are three input fields: "From" with a dropdown arrow, "To", and "Amount" with a "Currency" field next to it. At the bottom, there is an orange button labeled "Transfer".

#### Saisie par l'acteur

- Compte destinataire
- Montant
- Choisie de quel compte envoyer l'argent

#### Affiché par le système

- Message d'erreur
- Message de confirmation : que le transfert a été effectué.

#### Règles de gestion :

Les règles de gestion suivantes sont appliquées à la saisie de l'information.

référence	Explication
Numéro de compte	Le nombre de caractères doit être convenable

## My Profile :

**Objectif :** Voir ou modifier quelques éléments tels que mot de passe, adresse mail ou le numéro de téléphone.

**Acteurs :** Client, Système.

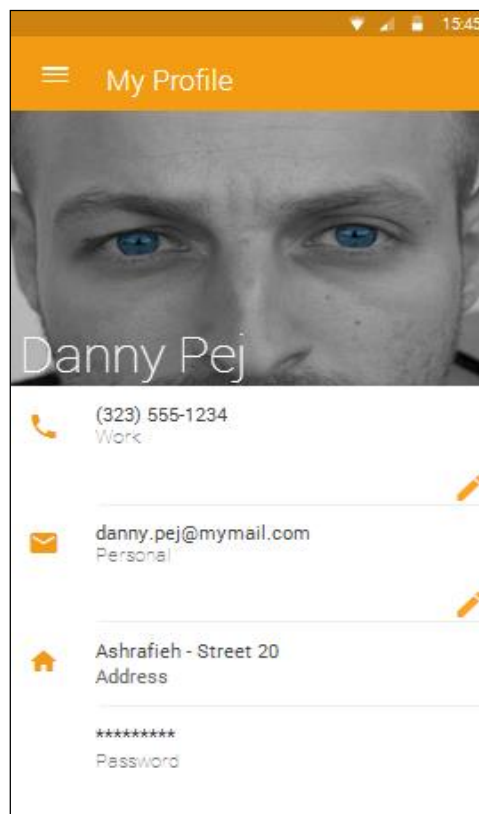
**Pré conditions :** L'utilisateur doit être membre et déjà authentifié.

### Scénario nominal :

- 1- Le client clique sur l'icône « My Profile ».
- 2- Le système le renvoie à la page.
- 3- Le client verra sa photo, son email, son numéro de téléphone, son adresse et son mot de passe. Il clique sur le crayon à coté de chacune pour modifier le champ correspondant.
- 4- Le système changera le champ cliqué sous forme de 'Edit Text' pour pouvoir le modifier.
- 5- Le client fera les modifications nécessaires et tape hors champ pour revenir à l'état principal.

### Scénario d'exception :

- 5-a- Email invalide, afficher un message d'erreur et revenir vers 3.
- 5-b- Numéro de téléphone invalide, afficher un message d'erreur et revenir vers 3.
- 5-c- Mot de passe < 6 chiffres, afficher un message d'erreur et revenir vers 3.



## Accounts :

**Objectif :** Le client va pouvoir consulter ces comptes et revoir les 10 dernières transactions effectuées sur chaque carte.

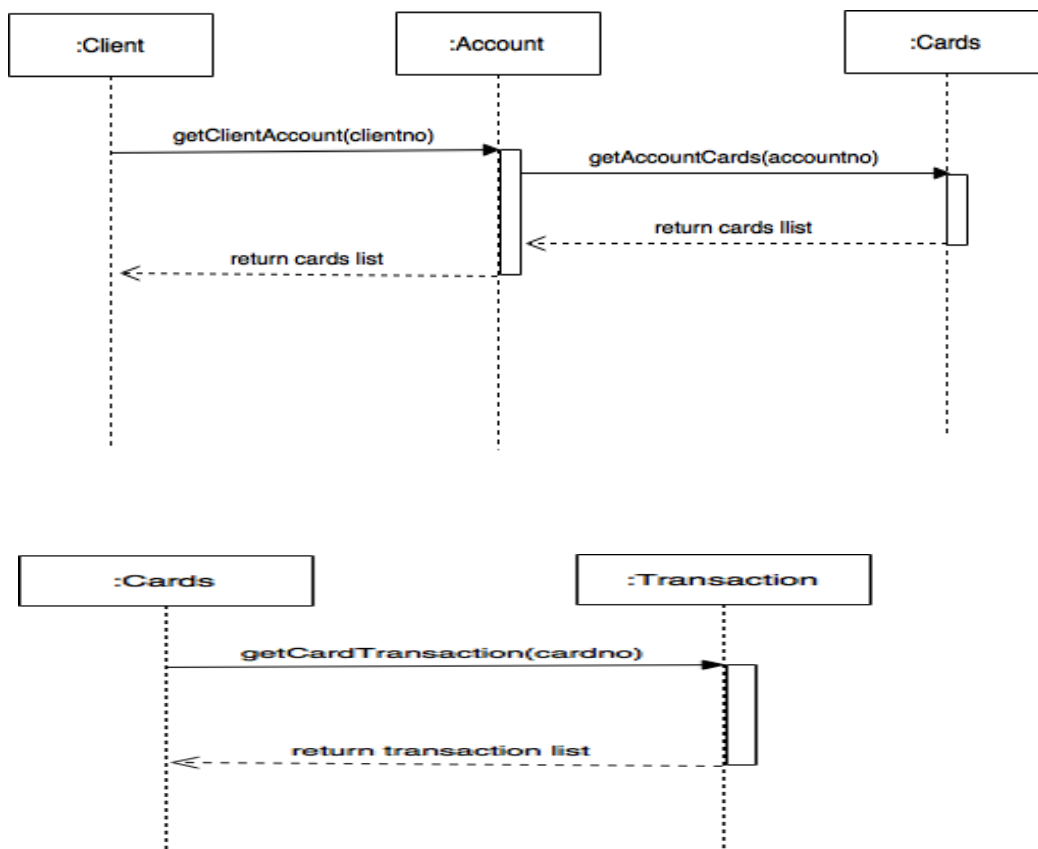
**Acteurs :** Client, Système.

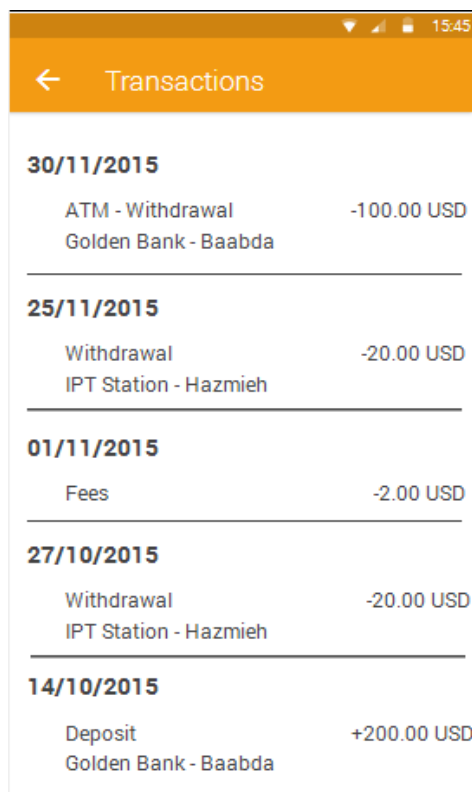
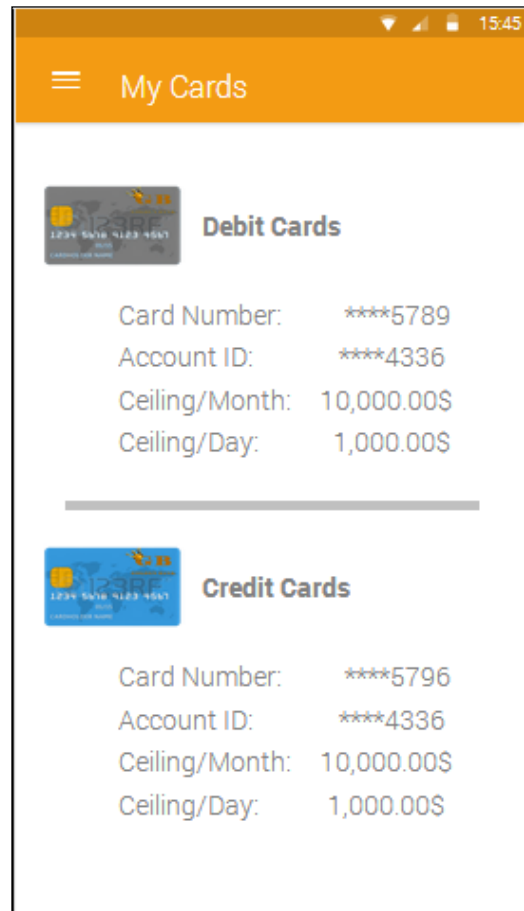
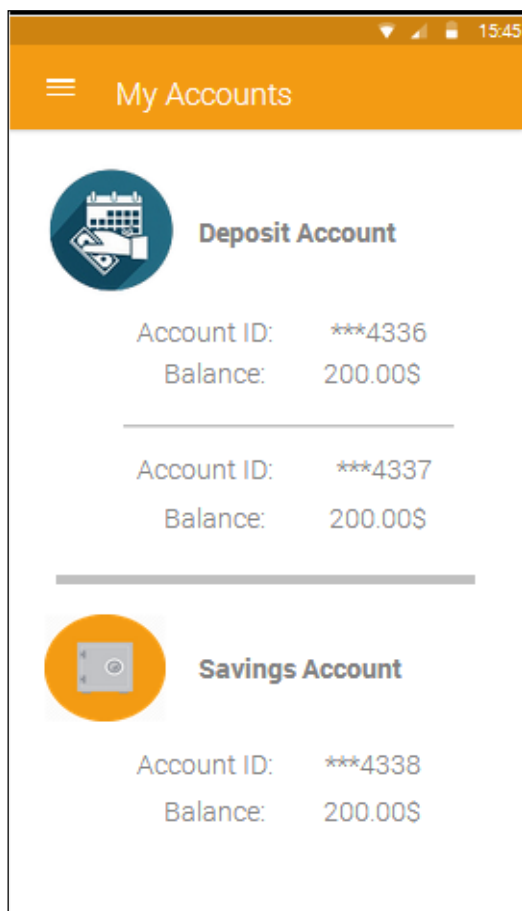
**Pré conditions :** L'utilisateur doit être membre et déjà authentifiée.

### Scénario nominal :

- 1- Le client clique sur l'icône « My Accounts ».
- 2- Le système le renvoie à la page « My Accounts », qui contiendra le solde de chaque compte avec son numéro.
- 3- Le client clique sur le compte dont il désire voir ces cartes.
- 4- Le système lui renvoie une page qui contient toutes les cartes attachées à ce compte.
- 5- Puis le client choisit une de ces cartes pour voir les 10 dernières transactions faites.

### Diagramme de séquence :







## ATM/Branches :

**Objectif :** L'utilisateur va pouvoir repérer l'ATM ou la branche la plus proche de là où il est.

**Acteurs :** Client, Public, Système.

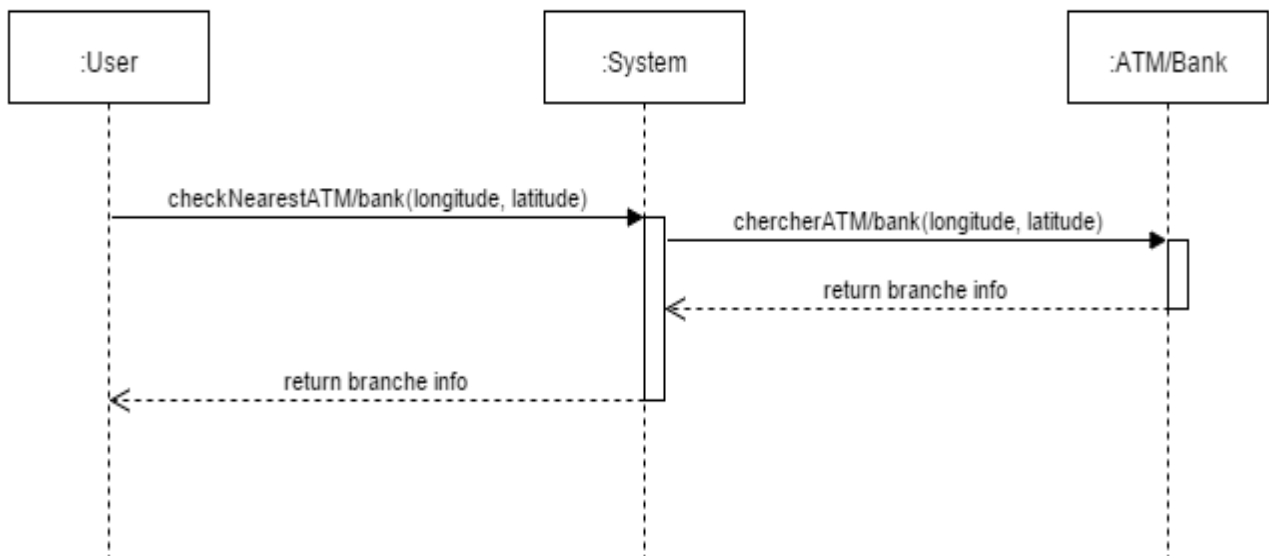
### Scénario nominal :

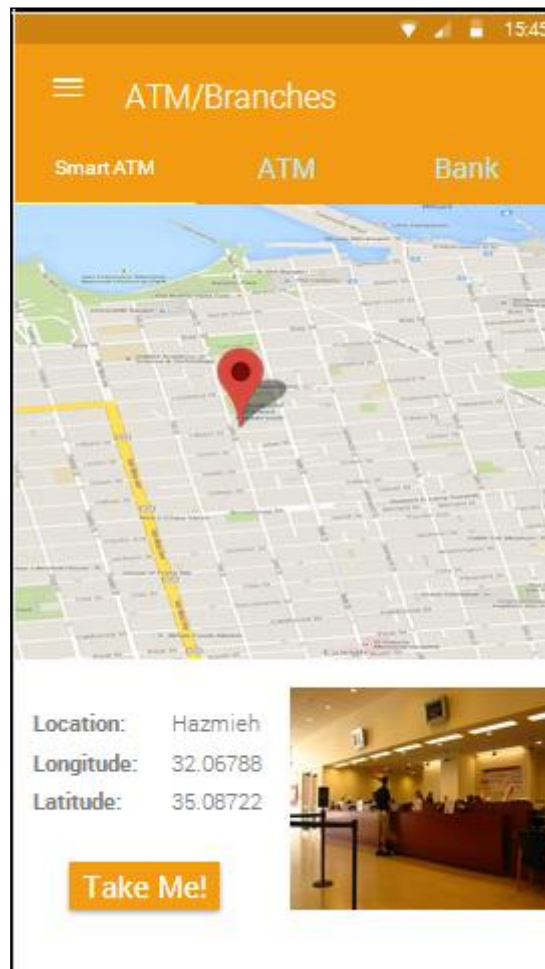
- 1- L'utilisateur (Client ou pas) appui sur « ATM/Branches »
- 2- Le système le renvoie à la page « ATM/Branches » et met par défaut la carte sur la branche de banque la plus proche de lui avec une photo et ces caractéristiques.
- 3- L'utilisateur choisit le tab qu'il veut : soit voir les branches de banque les plus proches, soit les ATM les plus proches de sa location.
- 4- Le client appuie sur 'Take me'.
- 5- Le système lui ouvre l'application Google maps (ou Here Maps) pour avoir les directions.

### Scénario d'exception :

2-a- Si l'utilisateur n'a pas la caractéristique de GPS allumée, la carte indiquera toutes les branches disponibles (soit ATM ou banque).

### Diagramme de séquence :





## Products :

**Objectif :** Le client ou le public va pouvoir consulter les différents produits que la banque a à offrir.

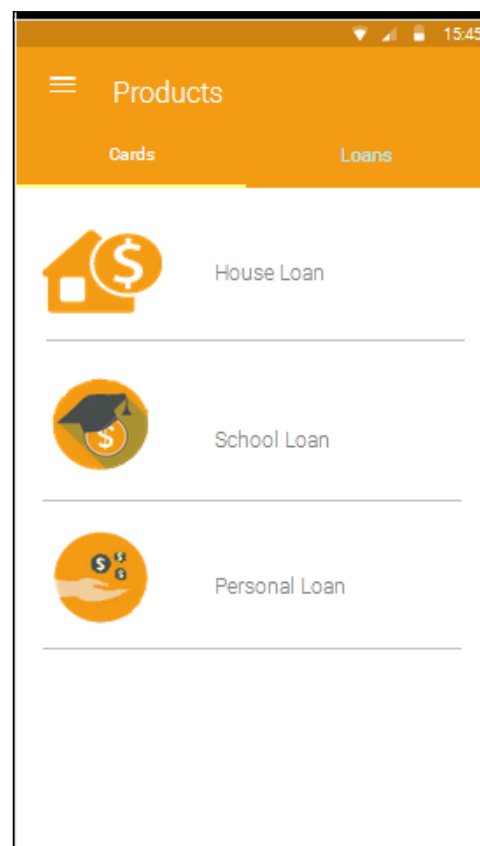
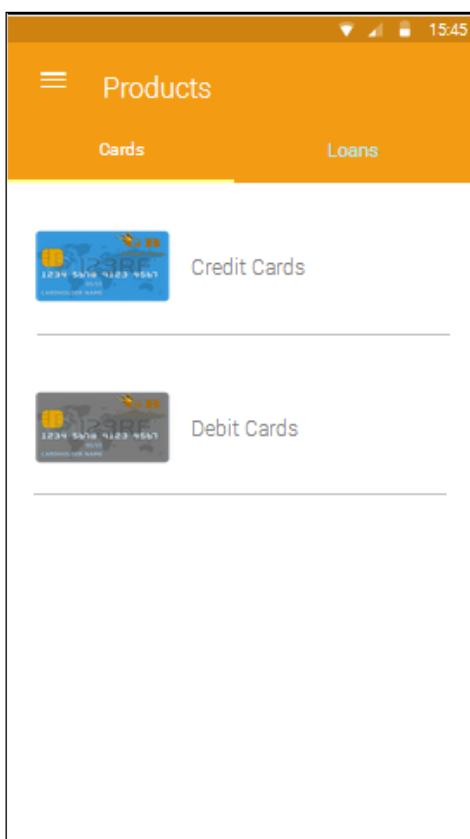
**Acteurs :** Client, public, système.

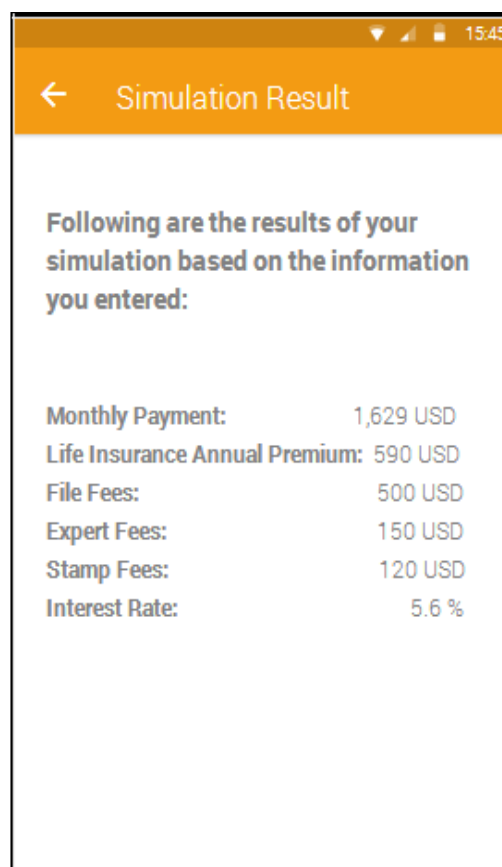
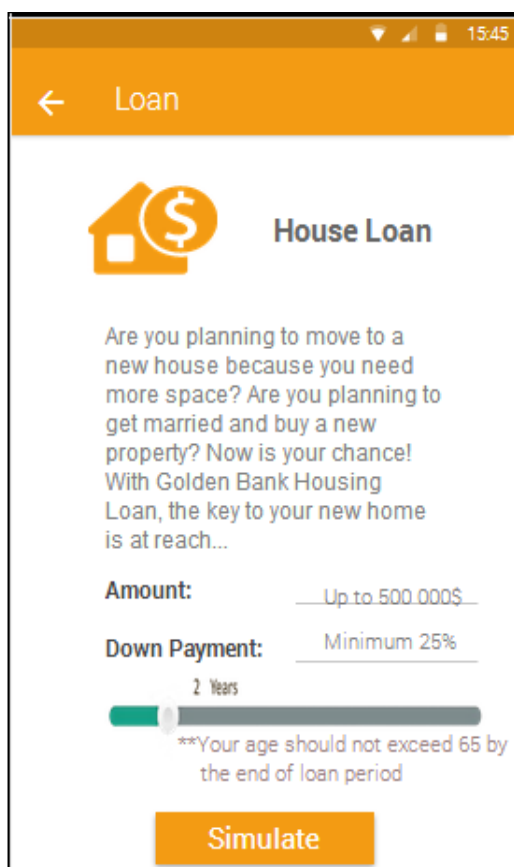
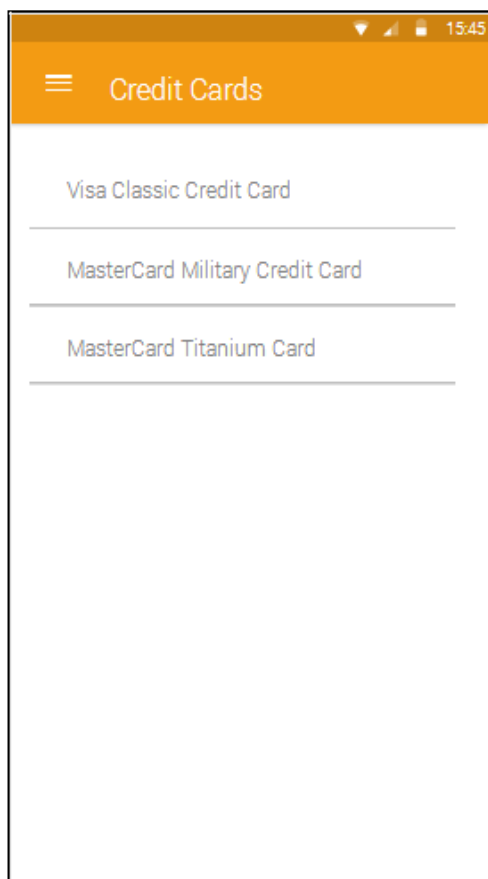
### Scénario nominal :

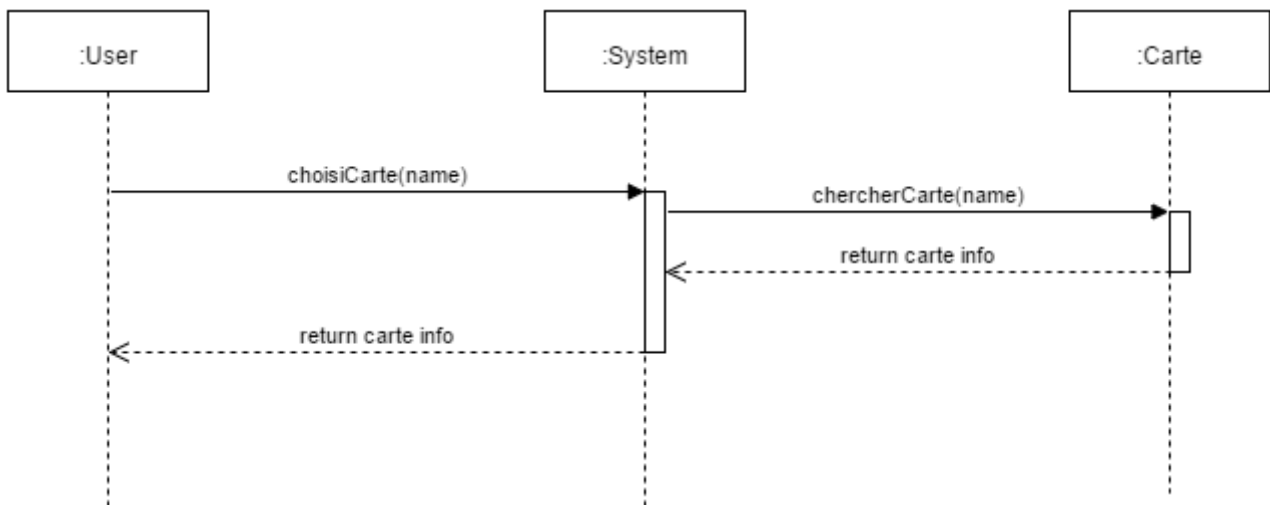
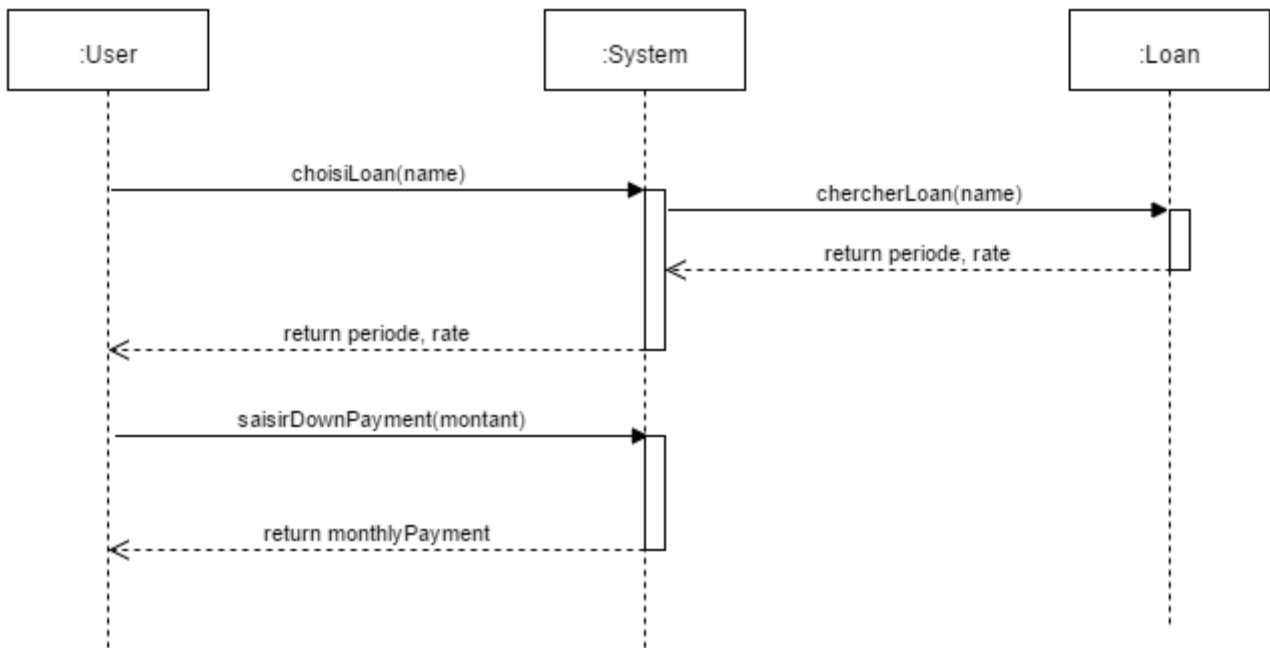
- 1- Le client appui sur « Products »
- 2- Le système le renvoie à la page « Products » qui contient 2 tabs : loans et cards (sur cards pas défaut).
- 3- Le client clique sur « Loans ».
- 4- Le système le renvoie à la page qui contient tous les loans,
- 5- Le client choisi le prêt qu'il veut, il aura : la période de remboursement, l'intérêt et le montant, puis le client choisi la somme du premier versement du prêt et puis clique sur 'Simulate'.
- 6- Le système fait le calcul et lui renvoie le payement mensuel.

### Scénario d'exception :

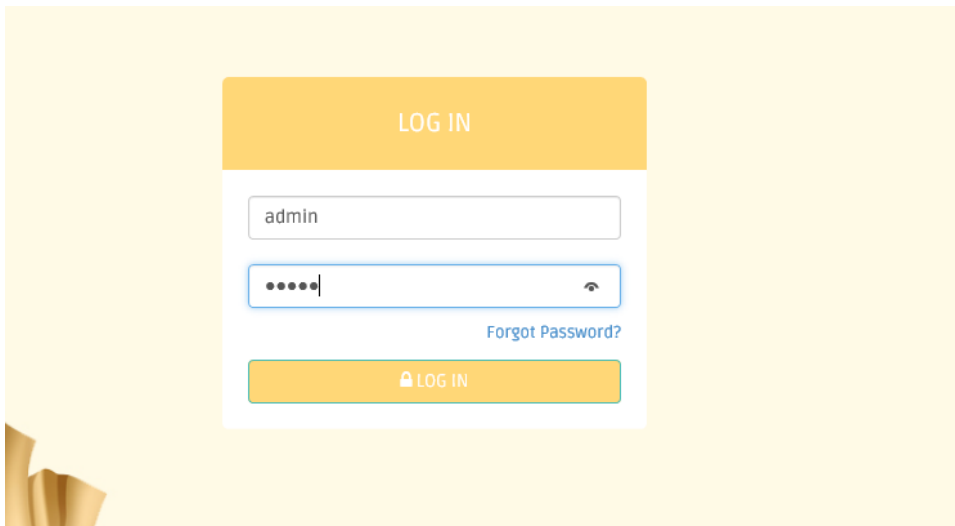
- 5-a- Le montant saisi est invalide, affiche un message d'erreur.







## Login :



A login form titled "LOG IN" is centered on a light yellow background. The form is white with a yellow header bar containing the text "LOG IN". It features a username input field with the text "admin", a password input field with masked characters "•••••" and a toggle icon, a "Forgot Password?" link, and a yellow "LOG IN" button with a lock icon.

LOG IN

admin

•••••

[Forgot Password?](#)

LOG IN

## Ajouter Client :



The "Add New Client" form is part of the Golden Bank dashboard. It has a yellow header bar with the Golden Bank logo and a "Logout" button. A dark blue sidebar on the left contains navigation links: "Clients", "Bank/ATM Branches", and "Loans / Cards". The main content area is titled "> New Client" and contains a form with fields for First Name, Last Name, Username, Password, Date Of Birth, Phone Number, and Email. An "Add" button is at the bottom left of the form.

GB GOLDEN BANK Logout

Clients > New Client

Bank/ATM Branches

Loans / Cards

First Name

Last Name

Username

Password

Date Of Birth

Phone Number

Email

Add

## Liste des produits :

GB GOLDEN BANK

Logout

Clients

Bank/ATM Branches

Loans / Cards

+ Add New Loan

> All Loans Products

Name	Maximum Amount	Interest	Description	
Product Loan 1	40000.0	1.4 %	Test Test	Delete

+ Add New Card

> All Products Cards

Name	Type	Monthly Limit	Daily Limit	Description	
Product Card 1	Credit	10000.0	1000.0	Test Test	Delete

## Ajouter Produit - Loan:

> New Product Loan

Name

Description

Amount

Interest

Add

## Ajouter Produit - Card:

> New Product Card

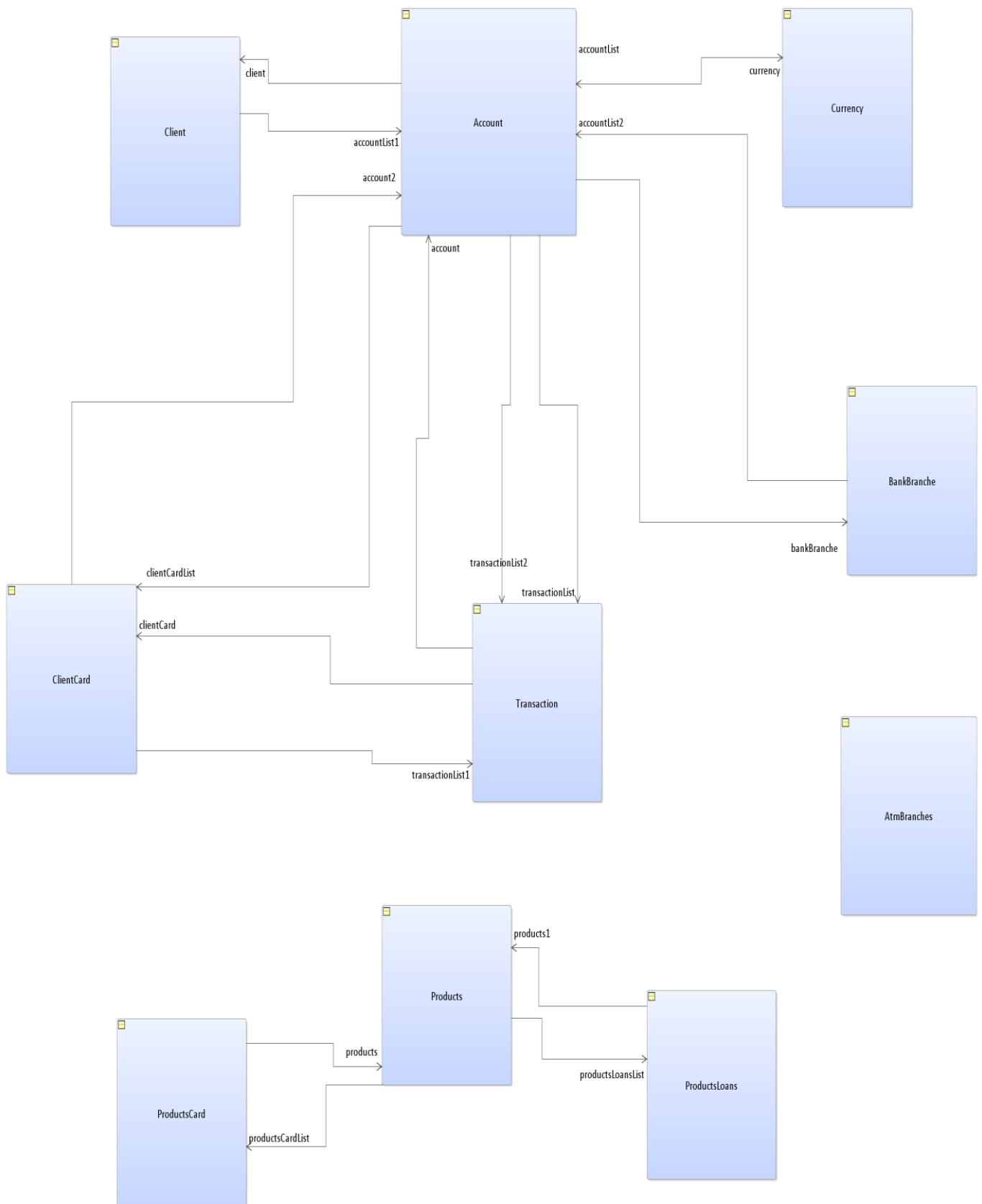
Name	<input type="text"/>
Card Type	<input type="radio"/> Credit <input type="radio"/> Debit
Monthly Limit	<input type="text"/>
Daily Limit	<input type="text"/>
Description	<input type="text"/>

Add



# G - Le modèle des données

## | Le diagramme de classes



### G.1.a - Classe Client

Cette classe servira à stocker toutes les informations des clients : Nom, prénom, username, password, etc...

Client									
Client_ID	Last_Name	First_Name	DOB	Username	Password	Phone_Number	Image	Email	Notification
4997	Saade	Rhea	24-07-95	rheasaade	Rhea4997	13234	<a href="#">Images/rhea</a>	<a href="mailto:rs@hotmail.com">rs@hotmail.com</a>	SMS

### G.1.b - Classe Account

Cette classe servira à stocker toutes les données d'un compte : Numéro, devise, montant etc... Le solde sera Credit - debit. Elle contient aussi l'id du client à qui appartient ce compte. Elle contient 2 méthodes : showBalance pour pouvoir voir le solde du compte et changeCurrency pour qu'on puisse changer la devise d'un compte.

Account						
Account_ID	Client_ID	Currency_Id	Branche_id	Credit	Debit	Type
10024997	4997	1	100	200	300	Savings
10124998	4998	2	101	300	200	Deposit
10214998	4998	3	102	100	100	Deposit

### G.1.c - Classe Transaction

Cette classe contiendra toutes les transactions faites sur une carte donnée. Comme elle contiendra aussi 2 attributs pour account : un pour le compte de départ et l'autre pour le compte sur lequel on fait un transfert, account2 sera nulle au cas de withdraw et deposit. Dans cette classe il a aussi 3 méthodes : Withdraw et deposit qui prennent en paramètre l'id du compte et le montant. Et la méthode Transfer qui sert à transférer d'un compte à un autre qui prend en paramètre 2 comptes et le montant.

Transaction							
Transaction_id	Date	Carte_number	Account_1	Account_2	Amount	Type	Rate
1	oct , 13 2015	249971	10024997	10124998	1000	Transfer	1507.5
2	oct , 14 2015	249972	10124998	10214998	2000	Transfer	1507.5
3	oct , 15 2016	249972	10214998	10124998	500	Transfer	1507.5
4	oct , 16 2016	249977	10024997	null	200	Withdrawal	1507.5
5	oct , 17 2017	249972	10024997	null	3650	Deposit	1507.5

### G.1.d - Classe Client Card

Chaque compte peut avoir une ou plusieurs carte(s), cette carte sera liée au compte par « compte id ». Elle a un type : credit ou debit card, et elle a aussi deux valeurs limites : transaction possible pour un jour et transaction possible pour un mois.

**Client\_Card**

Carte_ID	Account_ID	type	month_limit	Debit	Credit	day_limit
249971	10024997	Debit	5000\$	300	300	300\$
249972	10024997	Credit	5000\$	200	100	300\$
249973	10024997	Credit	5000\$	100	300	300\$

**G.1.e - Classe Currency**

Cette classe sert à stocker les devises disponibles dans la banque, chaque devise à son id. Dans notre banque il y aura 2 devises seulement : Livre Libanaise et Dollars. On a aussi besoin d'une variable pour stocker la valeur du taux d'échange.

**Currency**

Currency_ID	Name	Rate
1	LL	1507.5
2	USD	1507.5

**G.1.f - Classe BankBranche**

Cette classe sert à stocker toutes les branches de la banque avec leur location (longitude et latitude)

**Banque\_Branche**

Branche_ID	Nom	Image	Longitude	Latitude
100	Hazmieh	Images/bank/hazmieh	89.896276	21.390360
101	Baabda	Images/bank/baabda	74.044636	40.689060

**G.1.g - Classe ATMBranch**

Comme la classe BankBranche qui contient les différentes branches de la banque, cette classe contient les locations des ATM avec Nature : Smart ou Normal -- Genre : Interne ou externe.

**ATM\_Branches**

ATM_id	Longitude	Latitude	Genre	Nature
30	74.044636	21.390360	external	Smart
20	89.896276	40.689060	external	Normal
25	74.044636	21.390360	internal	Normal
40	89.896276	40.689060	external	Smart

**G.1.h - Classe Products**

C'est la classe mère des 2 classes : Products\_loans et Products\_cards

**Products**

Product_ID	Name		Description
------------	------	--	-------------

1	School loan	Images/products/1	Ce prêt vous aide à payer les frais de votre école...
2	House loan	Images/products/2	Ce prêt vous aide à payer les frais de votre maison future...
3	Platinum card	Images/products/3	Cette carte vous permet d'être en charge de votre compte...
4	Gold card	Images/products/4	Nombre illimitée d'accès à votre compte...

#### G.1.i - Classe Products\_Loans

C'est une classe fille de la classe Products, elle a comme attributs les caractéristiques qu'un prêt peut avoir : downpayment, montant, intérêt et le période. Elle a une seule méthode qui permet de calculer les paiements mensuels de l'utilisateur (simulate).

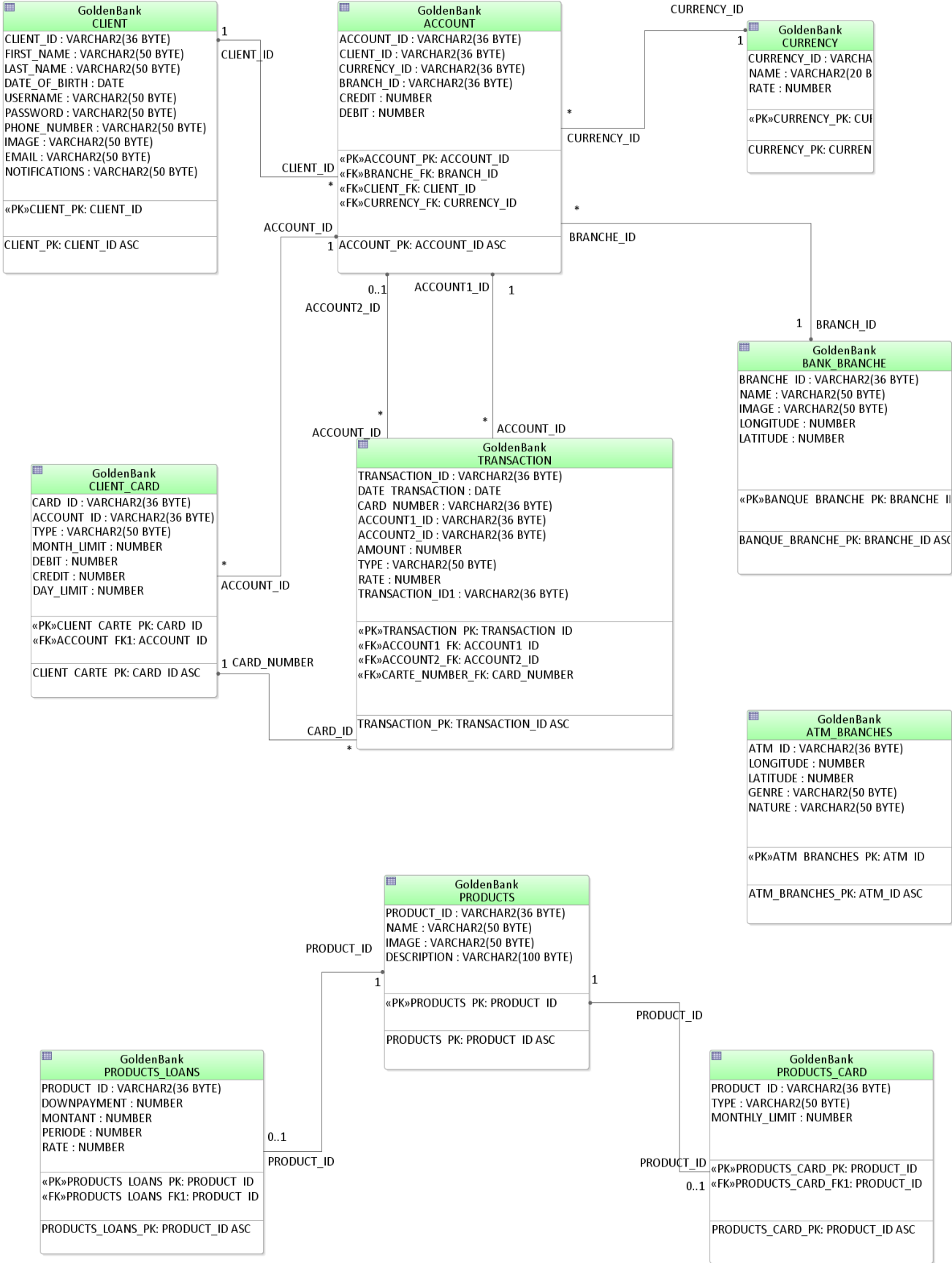
Products_Loans				
Product_ID	DownPayment	Montant	Periode	Rate
1	2000	10 000	10	3.4%
2	5000	600000	20	6%

#### G.1.j - Classe Products\_Cards

C'est la deuxième class fille de Products, ces attributs sont les caractéristiques d'une carte de banque : monthlyLimit, type (crédit ou débit).

Products_Card		
Product_ID	Type	Monthly_limit
3	Credit	20 000
4	Debit	60 000

## Le diagramme de base des données



## **H - Conclusion:**

Nous avons présenté dans ce rapport les diagrammes nécessaires pour la réalisation de notre projet :

Diagramme de classe et Diagramme de base de données.

En plus, on a fait les interfaces graphiques de toutes les pages de notre application avec leur diagramme de séquence.