Dossier de conception

Banque en ligne

# BENJILANY Boubeker

# HE Junyang

# AMGHAR Zakaria

# DJAMEN Yann

# NDJAMO TCHAPTCHET Ursula

Tuteur technique : Mr. Serais

Cliente : Mme. Bouzbouz

Coach Agile : Mme.Roueche

Année 2016 -2017

**Sommaire :**

I-Présentation de projet :

1. Objectif du projet
2. Fonctionnalités principales
3. Architecture souhaitée

II-Dossier de conception préliminaire (DSP) :

1. Architecture de la base de données
2. Requêtes de la base de données

III-Plan de Validation de Logiciel (PVL) :

**I-Document de Spécification Logiciel (DSL) :**

1. **Objectif du projet :**

L’ESIGELEC a été contactée par la DSI d’une banque classique qui souhaite se moderniser en offrant des services en ligne pour :

* améliorer le taux de satisfaction des clients,
* devenir attractive auprès des jeunes générations adeptes des offres digitales.

L’objectif est donc de créer un système d’information complet permettant aux clients ainsi qu’aux salariés de l’entreprise de gérer l’ensemble des comptes, transactions et produits financiers.

1. **Fonctionnalités principales :**

Les visiteurs devront être capables, via un client léger, de :

* consulter les offres de service de la banque
* lire les actualités de la banque
* demander des informations via le formulaire de contact
* créer un compte bancaire
* créer un compte épargne (livret A)
* créer un compte titre
* consulter les cours de la bourse

Les clients devront en plus être capables, via un client léger, de :

* visualiser et modifier leurs informations personnelles
* consulter le solde et l’historique des transactions de leurs comptes
* faire une extraction des relevés de compte(s) au format .csv
* effectuer un virement (interne ou externe)
* alimenter un compte
* acheter et vendre des actions du CAC40

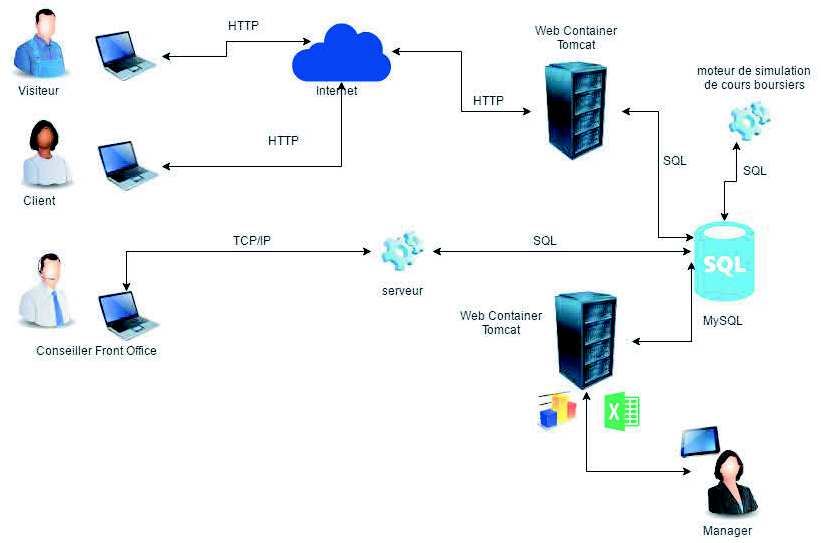
Les conseillers front office devront être capables, via un client lourd, de :

* voir la liste des clients
* consulter les informations personnelles des clients
* consulter le solde et l’historique des transactions de tous les comptes des clients

Les managers devront être capables, via un client léger, de :

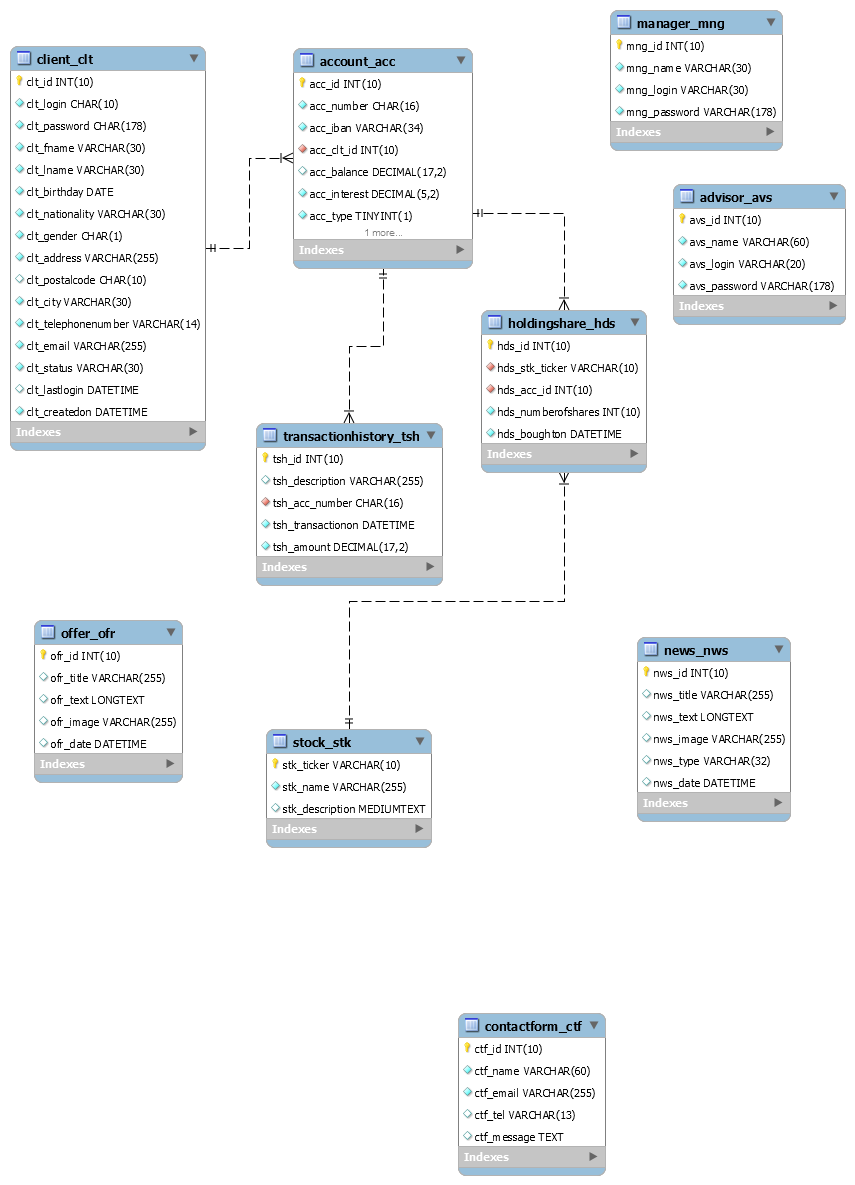
* Consulter la somme des dépôts des clients
* Générer des reportings sur les activités des clients

1. **Architecture Souhaitée :**



**II- Dossier de conception préliminaire :**

**1-Architecture de la base de données :**



**2-Requête de la base de données :**

* Pour un compte :

Trouver un compte avec son id :

"SELECT \* FROM Account\_acc WHERE acc\_id = ?";

Afficher la liste des comptes :

"SELECT \* FROM Account\_acc";

Trouver un compte avec l’id du client :

"SELECT \* FROM Account\_acc WHERE acc\_clt\_id = " + String.*valueOf*(id);

Trouver un compte avec son IBAN :

"SELECT \* FROM Account\_acc WHERE acc\_iban = ?";

Attribuer un nouveau numéro de compte utilisable :

"SELECT acc\_number FROM account\_acc WHERE acc\_id = (( "

+ "SELECT AUTO\_INCREMENT "

+ "FROM information\_schema.tables "

+ "WHERE table\_name = 'account\_acc' "

+ "AND table\_schema = DATABASE( )) - 1)";

* Pour un conseiller :

Trouver un conseiller avec son id :

"SELECT \* FROM advisor\_avs WHERE avs\_id = ?";

Afficher la liste des conseillers :

"SELECT \* FROM advisor\_avs";

Trouver un conseiller avec son login

"SELECT \* FROM advisor\_avs WHERE avs\_login=?";

* Pour un client :

Trouver un client avec son id :

"SELECT \* FROM client\_clt WHERE clt\_id = ?";

Afficher la liste des clients :

"SELECT \* FROM client\_clt";

Trouver un client avec son login :

"SELECT \* FROM client\_clt WHERE clt\_login=?";

Trouver un client avec son nom :

"SELECT \* FROM client\_clt WHERE clt\_lname=?";

Attribuer un nouveau login utilisable pour un client :

"SELECT clt\_login FROM client\_clt WHERE clt\_id = (( "

+ "SELECT AUTO\_INCREMENT "

+ "FROM information\_schema.tables "

+ "WHERE table\_name = 'client\_clt' "

+ "AND table\_schema = DATABASE( )) - 1)";

* Pour un formulaire de contact :

Trouver un formulaire de contact avec son id :

"SELECT \* FROM ContactForm\_ctf WHERE ctf\_id = ?";

Afficher la liste des formulaires de contact :

"SELECT \* FROM ContactForm\_ctf";

* Pour les achats :

Trouver un achat avec son id :

"SELECT \* FROM HoldingShare\_hds WHERE hds\_id = ?";

Afficher la liste des achats :

"SELECT \* FROM HoldingShare\_hds";

Trouver un achat avec l’id du client:

"SELECT \* FROM holdingshare\_hds "

+ "INNER JOIN account\_acc ON holdingshare\_hds.hds\_acc\_id = account\_acc.acc\_id "

+ "INNER JOIN client\_clt ON account\_acc.acc\_clt\_id = client\_clt.clt\_id "

+ "WHERE client\_clt.clt\_id = " + String.*valueOf*(cltId);

Trouver un achat avec l’id du compte :

"SELECT \* FROM holdingshare\_hds "

+ "INNER JOINT account\_acc ON hds\_acc\_id = acc\_id "

+ "WHERE acc\_id = " + String.*valueOf*(accId);

* Pour un manager :

Trouver un manager avec son id :

"SELECT \* FROM manager\_mng WHERE mng\_id = ?";

Trouver un manager avec son login :

"SELECT \* FROM manager\_mng WHERE mng\_login = ?";

Afficher la liste des managers :

"SELECT \* FROM manager\_mng";

* Pour une actualité :

Trouver 3 actualités de la banque :

"SELECT \* FROM News\_nws WHERE nws\_type = 'BankRading' ORDER BY nws\_date DESC LIMIT 3";

Trouver 3 autres actualités générales :

"SELECT \* FROM News\_nws WHERE nws\_type != 'BankRading' ORDER BY nws\_date DESC LIMIT 3";

Trouver toutes les actualités concernant la banque :

"SELECT \* FROM News\_nws WHERE nws\_type = 'BankRading' ORDER BY nws\_date DESC";

Trouver toutes les autres actualités générales :

"SELECT \* FROM News\_nws WHERE nws\_type != 'BankRading' ORDER BY nws\_date DESC";

Trouver une actualité avec son id :

"SELECT \* FROM news\_nws WHERE nws\_id = ?";

Afficher la liste des actualités :

"SELECT \* FROM News\_nws";

* Pour une offre :

Trouver une offre avec son id :

"SELECT \* FROM offer\_ofr WHERE ofr\_id = ?";

Afficher la liste des offres :

"SELECT \* FROM offer\_ofr";

* Pour une action :

Trouver une action avec son id :

"SELECT \* FROM Stock\_stk WHERE stk\_ticker = ?";

Afficher la liste des actions :

"SELECT \* FROM Stock\_stk";

* Pour l’historique des prix des actions :

Trouver l’historique des prix avec son id :

"SELECT \* FROM StockHistoricalPrice\_shp WHERE shp\_id = ?";

Afficher la liste des historiques des prix :

"SELECT \* FROM StockHistoricalPrice\_shp";

Trouver l’historique des prix d’une action avec son id :

"SELECT \* FROM stockhistoricalprice\_shp “+ "WHERE shp\_stk\_id = " + String.*valueOf*(stkId);

* Pour l’historique des transactions :

Trouver l’historique des transactions avec son id :

"SELECT \* FROM TransactionHistory\_tsh WHERE tsh\_id = ?";

Afficher la liste des historiques des transactions :

"SELECT \* FROM TransactionHistory\_tsh";

Trouver l’historique des transactions par le numéro du compte :

"SELECT \* FROM transactionhistory\_tsh WHERE tsh\_acc\_number = " + number;