



Projet Java

Gestion de comptes bancaires

Réalisé par : *Abdelmounajja ZAKARIA & Driss ESSIKA*

Encadrés par *Monsieur Michel BUFFA*

Table des matières

1. Introduction :	3
2. Sujet du projet :	3
2.1. Configuration :	3
2.2. Fonctionnalités de l'application :	4
2.2.1. Configuration du SMTP.....	4
2.2.2. Ajouter un nouveau client.....	4
2.2.3. Liste des clients :.....	7
2.2.4. Effectuer un versement.....	8
2.2.5. Effectuer un retrait :.....	8
2.2.6. Informations d'un compte :.....	9
2.2.7. Identification des clients :	9
2.2.8. Page du client :	11
2.2.9. Mes comptes :	12
2.2.10. Mes historiques :	12
2.2.11. Les informations :	13
2.2.12. Etat de mon compte :	13
2.2.13. JMS :	14
3. Les technologies utilisées :	14
3.1. EJB :.....	14
3.2. JMS :	14
3.3. Dojo :	14
3.4. Ajax :	14
3.5. Les JSP :.....	15
3.6. Les Servlets :	15
3.7. Les taglibs :	15
3.8. Javascript :	15
4. Conclusion :	16

1. Introduction :

Ce rapport présente le travail effectué par deux étudiants en Master 2 Méthodes Informatiques Appliquées à la Gestion des Entreprises spécialité Nouvelles Technologies et Direction de Projets dans le cadre d'un projet d'application web.

Le but de ce projet consiste en l'élaboration d'une application web permettant la gestion de comptes bancaires d'une banque.

En effet, il a fallu concilier tout au long de ce projet nos compétences de techniciens ainsi que celles de gestionnaires de projet.

L'application a été modélisée tel que nous l'avons comprise et compte tenu des moyens matériels et temporels dont nous disposons.

Mots clé : EJB, Héritage, Hibernate, JSP, Servlet, Ajax, Dojo et Java.

2. Sujet du projet :

Créer un portail de gestion de comptes bancaires (ajout de nouveaux clients, leurs attribuer des comptes bancaires, effectuer des opérations telles que les virements et le retraits) et permettre aux clients d'accéder à leurs comptes via une interface web. L'identification des clients se fera à l'aide d'un login et mot de passe qui leurs seront envoyés par courrier électronique. Le but principal de ce projet est de réaliser l'ensemble de cette application avec les technologies EJB, JMS, Héritage, Dojo sous Java.

2.1. Configuration :

Afin de faire fonctionner notre application sur votre machine, vous devez :

- Créer une base de données nommée **Banque_EMZ** avec
 - un login : **test**
 - un mot de passe : **test**.
- Toutes les tables seront créées automatiquement.
- Il faudra ajouter des clients et des comptes bancaires.

2.2.Fonctionnalités de l'application :

2.2.1. Configuration du SMTP

La création d'un nouveau client implique l'envoi d'un courrier électronique contenant son login et mot de passe. Ces derniers vont lui permettre d'accéder à son portail personnel, qui lui permettra de visualiser l'ensemble des informations relatives à ses comptes bancaires. Pour cela, vous devez aller dans la rubrique Javamail, et saisir le smtp relatif à votre Fournisseur d'Accès Internet.

Gestion Comptes bancaires

Accueil Comptes Clients Opérations **Javamail**

☰ Gestion de javamail

Configuration du serveur mail

Afin d'assurer le bon fonctionnement de Java Mail, vous mettre le smtp de votre fournisseur d'accès Internet

SMTP Actuel :

Options

- Comptes
- Comptes
- Comptes

2.2.2. Ajouter un nouveau client

Pour ajouter un nouveau client, vous devez aller dans la rubrique Clients, et cliquer sur Nouveau clients dans le menu à droite. Vous aurez une page qui ressemble à

Gestion Comptes bancaires

Accueil Comptes **Clients** Opérations Javamail

☰ Gestion des clients

Nom	<input type="text" value="BUFFA"/>
Prenom	<input type="text" value="Michel"/>
adresse	<input type="text" value="Avenue Valrose"/>
Date de naissance	<input type="text" value="11/06/1965"/>
ville	<input type="text" value="BIOT"/>
code postale	<input type="text" value="06200"/>
mail	<input type="text" value="azakaria06@gmail.com"/>

Options

- Nouveau Client
- Liste des clients
- Portail des clients

Vous saisissez les informations relatives à votre client (exemple Michel BUFFA), nous avons mis en place une fonction javascript qui permet de vérifier si tous les champs ont été bien renseignés.

Le nom le prénom, la ville et l'adresse email doivent avoir une longueur minimale de 5. Le code postale ne doit contenir que des chiffres, c'est pour cela que nous avons fais de manière à ce que toute lettre saisie soit vérifiée, s'elle s'agit d'un chiffre on l'accepte sinon on la supprime automatiquement. La vérification de la bonne syntaxe de l'adresse email est aussi vérifiée (une adresse email doit contenir : @ et un point).

Un focus sur le champ date de naissance entraine l'apparition d'un calendrier permettant à l'utilisateur de sélectionner la date de naissance du client.

Dim	Lun	Mar	Me	Jeu	Ven	Sam
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Un clic sur créer, enclenche la vérification des données saisies, s'elles sont valides, un script Ajax permettra d'envoyer les données au serveur qui s'occupera de la création de l'objet client et de l'ajout de ses données dans la base de données grâce à l'EJB ainsi qu'une notification par mail contenant un login et un mot de passe. En retour, le script ajax récupère le numéro du client que l'on vient d'ajouter qui nous sera utile pour la création du compte.

Notre banque propose trois types de comptes. Un compte courant, un livret jeune et un livret A. Vous devez choisir le type de compte que vous voulez attribuer au client que vous venez d'enregistrer.

Compte Bancaire :

- COMPTÉ COURANT
- LIVRET A
- LIVRET JEUNE

Quand vous choisissez par exemple Compte Courant, vous aurez une page :

Compte Bancaire :

COMPTE COURANT

Compte courant, utilise dans les relations commerciales et financieres. Ce compte centralise les versements et les retraits en espece, les encaissements et paiements utilisant un autre moyen de paiement que les especes : cheque, carte de paiement, virement, prelevement, et les mouvements de fonds avec les autres comptes et produits bancaires (compte d'epargne, plan d'epargne, compte de pret, compte titres...);

Montant solde initiale

Solde Initiale €

10000

CrÃ©er le compte

Et là vous devez saisir le montant du compte initial du client. 10 000€ pour l'exemple. Entre temps, votre client recevra un courrier électronique contenant son login et son mot de passe. :



2.2.3. Liste des clients :

Pour afficher la liste des clients de la banque, il suffit d'aller dans la rubrique **Clients**, et cliquer sur **Liste des clients** dans le menu à droite.

Effectuer un virement

Pour effectuer un virement d'un compte à un autre, il faut aller dans la rubrique **Opérations**, et cliquer sur **Effectuer un virement** dans le menu à droite.

Pour cet exemple, nous considérons que Zakaria versera une somme de 1000€ sur le compte de M. Michel BUFFA :

Opérations sur les comptes

Compte à débiter

2424832 - Compte Courant - 8000.0

Numero de compte

2424832

Numero Client :

2392064

Client :

ZAKARIA Abdelmounajja

Solde actuel :

8000

Euros

Compte à créditer

2555904 - Compte Courant - 10000.0

Numero de compte

2555904

Numero Client :

2523136

Client :

BUFFA Michel

Solde actuel :

10000

Euros

Montant Transaction : 1000

Effectuer le virement

Techniquement, la récupération des informations relatives aux comptes sélectionnés se fait par Ajax, qui envoie une sorte de requête à une Servlet. En effet, quand vous choisissez le compte de Zakaria en tant que compte débiteur, vous ne le voyez plus apparaître dans la liste des comptes à créditer (Logique, on ne peut pas effectuer une opération de virement sur le même compte).

Suite au clic que le bouton **Effectuer le virement**, une fonction ajax envoie toutes ces données à une Servlet qui se chargera de mettre à jour la base de données.

2.2.4. Effectuer un versement

Nous allons maintenant verser une somme de 1000€ sur le compte de M. BUFFA, pour cela nous allons sur la rubrique **Opérations** puis cliquer sur **Effectuer un versement** dans le menu à droite. Il suffira de sélectionner le compte, des informations relatives au compte seront affichées, ceci se fait grâce à Ajax. Il ne reste qu'à saisir le montant du versement.

Opérations sur les comptes

Options

- Effectuer un virement
- Effectuer un versement
- Effectuer un retrait
- Portail des clients

Veillez choisir le compte

2555904 - Compte Courant - 11000.0

Numero de compte

2555904

Numero Client :

2523136

Client :

BUFFA Michel

Solde actuel :

11000

Euros

Montant versement

Montant versement €

1000

Valider

2.2.5. Effectuer un retrait :

L'opération du retrait similaire à opération de versement, sauf qu'on débite le compte.

2.2.6. Informations d'un compte :

Pour visualiser les informations relatives à un compte, il faut aller dans la rubrique **Comptes** puis sélectionner un compte. A la sélection, une fonction ajax s'active pour récupérer toutes les données du compte. Dans l'exemple, on peut voir le nom et prénom du client, le solde actuel, et un historique de l'ensemble des opérations qui ont eu lieu sur le compte. (Virements, Retraits et Versements). Quand il s'agit d'un virement, nous pouvons visualiser le nom du client qui nous a donné ou à qui nous avons donné de l'argent.

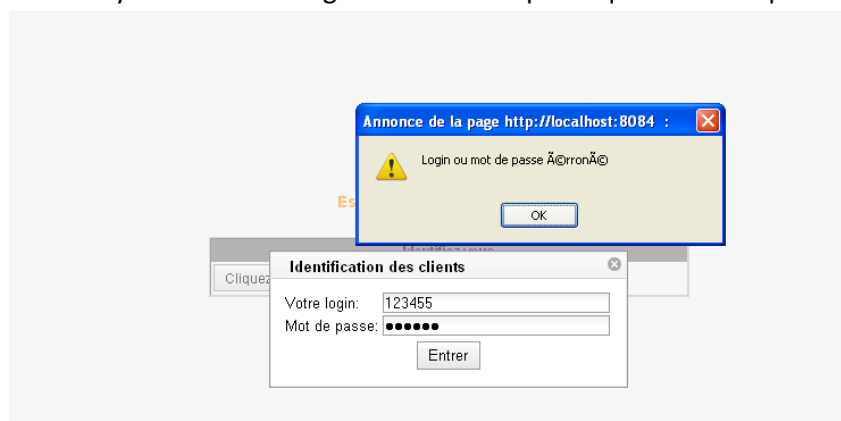
The screenshot displays a web interface titled "Informations sur les comptes bancaires". It features a dropdown menu for "Liste des comptes bancaires" with the following options: "2555904 - Compte Courant - 12000.0", "2424832 - Compte Courant - 7000.0", "2490368 - Compte Courant - 3000.0", and "2555904 - Compte Courant - 12000.0". The selected account number is "2555904". Below this, the client's name is "Client : BUFFA Michel", the current balance is "Solde actuel : 12000 Euros", and the transaction history is titled "Historique des opérations :". The history includes: "11/12/2009 Virement de 1000€ reçu de la part de ZAKARIA Abdelmounajja", "11/12/2009 Versement de 1000 €", "11/12/2009 Versement de 1000 €", and "11/12/2009 Retrait de 1000 €". On the right side, there is an "Options" menu with three items: "Comptes", "Comptes", and "Portail des clients".

2.2.7. Identification des clients :

Pour accéder au portail des clients, il faut cliquer sur **Portail des clients** ce liens existe dans toutes les rubriques, sinon vous pouvez y accéder en ajouter **/identification.jsp** dans la barre d'adresse de votre navigateur.

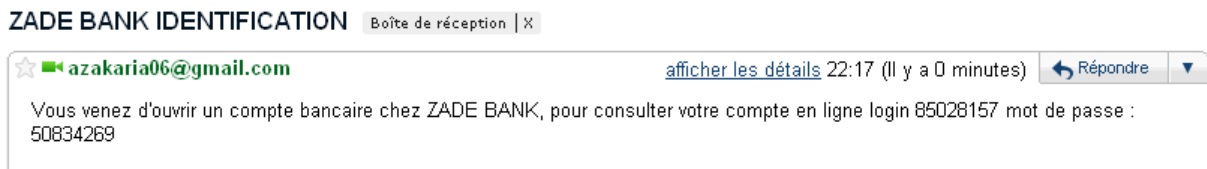
Cette page est entièrement faite par Dojo. Pour se accéder à votre compte, vous devez cliquer sur le bouton **Cliquez ici pour vous identifier**. Le clic sur ce dernier entraine l'affichage d'un formulaire entièrement fait par Dojo.

Nous allons essayer de saisir un login et un mot de passe qui n'existent pas :

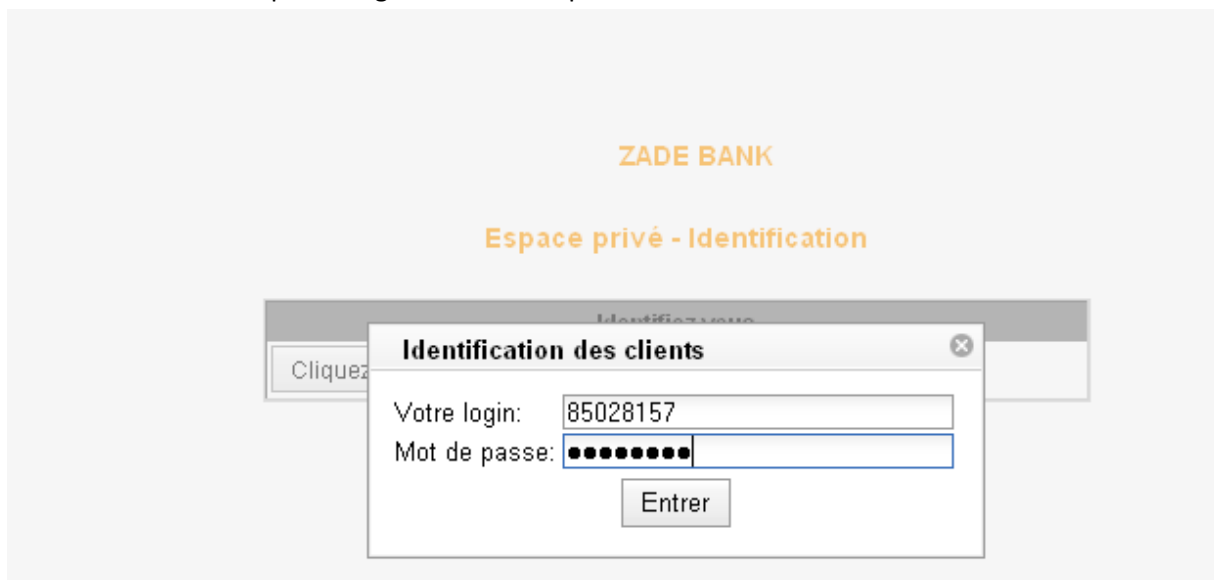


Au clic sur le bouton entrer, Ajax envoie le login et le mot de passe à une Servlet, qui elle-même va vérifier s'il existe un client possède ce login puis ce mot de passe. Dans notre cas, ce login et mot de passe ne correspond à aucun client, la servlet renvoie un message d'erreur à Ajax, ce dernier interprète le message envoyé par la Servlet et puisqu'il s'agit d'un message, on alerte le client pour l'informer que le login et mot de passe saisis sont incorrects. Cette opération ne nécessite aucun rechargement de la page grâce à Ajax.

Maintenant nous allons effectuer une vraie identification, référons nous au mail envoyé par l'application :



Nous allons copier le login et le mot de passe :



De point de vue technique, le script Ajax envoie le login et le mot de passe à la Servlet, cette dernière effectue un traitement et trouve que le login et le mot de passe saisis correspondent au client Michel Buffa. Donc elle va créer une nouvelle session et insérera dedans l'objet Client, et enverra une notification au JMS, pour l'informer qu'une nouvelle connexion au portail de M. BUFFA. A la réception de ce message, le JMS créera un objet connexion et l'ajoutera dans la base de données. Cette opération permettra au banquier de contrôler le nombre de connexion de ses clients, ainsi que pour des raisons de sécurité. Finalement, le client sera redirigé vers son portail.

2.2.8. Page du client :



En premier temps, l'application nous rappelle la date de notre dernière connexion au portail. Ceci est fait par mesure de sécurité. Imaginons que M. BUFFA était en vacances durant le mois d'Aout, et n'a jamais eu accès à Internet pour consulter son compte. Au début de Septembre, il consulte son compte et se rend compte qu'une connexion à sa page a été effectuée en mi Août. Cela dit que quelqu'un lui a volé ses données d'identification et doit prévenir la banque dans les plus brefs délais pour leur demander de lui changer son mot de passe.

En fait, la récupération des données relatives au client connecté se fait grâce à l'objet Client stocké dans la session.

Par mesure de sécurité, nous avons forcé l'expiration de la session au bout de quelques minutes d'inactivités.

2.2.9. Mes comptes :

Ajax envoie une demande à la servlet, qui lui renvoie en retour du texte contenant toutes les informations relatives au compte. Le javascript édite le fichier html et met à jour la balise concerné.



2.2.10. Mes historiques :

Même chose que mes comptes.



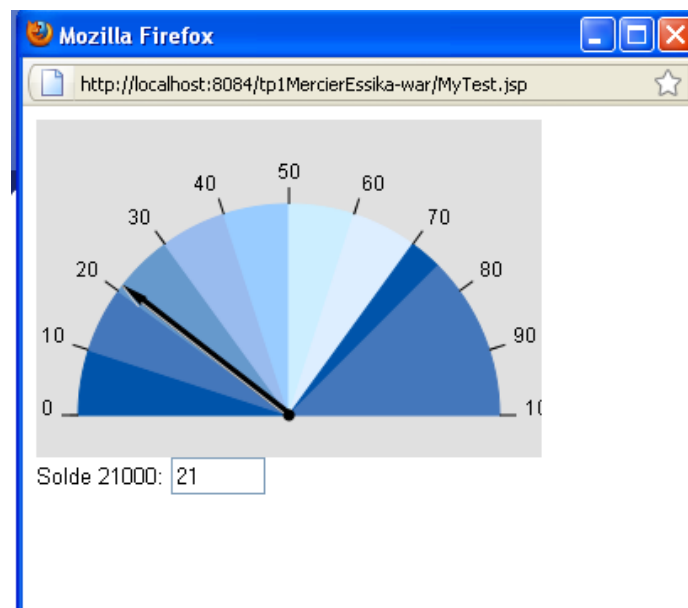
2.2.11. Les informations :

Pour afficher les informations du client, il suffit de cliquer sur Mes informations. Il nous suffit de faire des Get sur l'objet client pour récupérer son nom, adresse etc.



2.2.12. Etat de mon compte :

Le clic sur ce lien entraîne l'ouverture d'une nouvelle fenêtre entièrement faite par Dojo. En effet, Ajax récupère le solde du compte et le renseigne à Dojo qui va placer l'aiguille au bon endroit. La position de l'aiguille reflète le solde du compte.



2.2.13. JMS :

Nous avons utilisé le Java Message Service dans notre projet pour enregistrer l'ensemble de connexion des utilisateurs, dès qu'un client s'identifie et accède à son compte, nous alertons le serveur de cette nouvelle connexion, ce dernier ajoute l'ensemble des informations liées à cette identification dans la base de données.

3. Les technologies utilisées :

L'utilisation des technologies EJB, JMS, Dojo, Ajax, Taglibs , javascript nous a été imposée pour réaliser notre application.

3.1. EJB :

Eterprise JavaBeans (EJB) est une architecture pour la mise en place des composantes d'un programme écrit sous Java, qui s'exécutent dans la partie serveur d'un réseau informatique qui utilise le modèle client/serveur. Entreprise Java Beans est construit sur la technologie JavaBeans pour la distribution des composantes du programme à des clients dans un réseau.

3.2. JMS :

Java Message Service est une interface de programmation qui permet d'envoyer et de recevoir des messages de manière asynchrone entre applications ou composants Java. JMS permet d'implémenter une architecture de type MOM.

Un client peut également recevoir des messages de façon synchrone dans le mode de communication point à point (pour cela, il doit invoquer la méthode receive() qui est bloquante).

L'API JMS permet aux applications Java de s'interfacer avec des intergiciels (middleware) à messages ou MOM (Message Oriented Middleware), les MOMs permettent des l'interaction entre composants applicatifs dans un cadre faiblement couplé, asynchrone et fiable.

3.3. Dojo :

Dojo est un framework de developement AJAX permettant une écriture simplifiée de d'interfaces interactives pour les applications web.

Plus qu'un simple set de widget, dojo permet de plus une nouvelle approche de la programmation Javascript avec de nombreuse API telles que Reflect, Math, Lang et bien d'autres.

3.4. Ajax :

Ajax est un raccourci pour Asynchronous JavaScript and XML (« XML et Javascript asynchrones ») et désignant une solution informatique libre pour le développement de pages dymaniques et d'applications Web.

À l'image de DHTML ou de LAMP, AJAX n'est pas une technologie en elle-même, mais un terme qui évoque l'utilisation conjointe d'un ensemble de technologies libres couramment utilisées sur le Web:

3.5. Les JSP :

Le JavaServer Pages ou JSP est une technologie basée sur Java qui permet aux développeurs de générer dynamiquement du code HTML, XML ou tout autre type de page web. La technologie permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique. Depuis la version 2.0 des spécifications, la syntaxe JSP est complètement XML.

3.6. Les Servlets :

Les servlets (on dit généralement une servlet) sont au serveur Web ce que les applets sont au navigateur pour le client. Les servlets sont donc des applications Java fonctionnant du côté serveur au même titre que les CGI et les langages de script côté serveur tels que ASP ou bien PHP. Les servlets permettent donc de gérer des requêtes HTTP et de fournir au client une réponse HTTP dynamique (donc de créer des pages web dynamiques).

3.7. Les taglibs :

Les JSP Tag Bibliothèques permettent la création et l'utilisation de bibliothèques de tags au sein des pages JSP. Une JSP Taglib est une collection d'actions prédéfinies destinée à être utilisée dans une page JSP sous forme de tags (balises XML). Elle se compose d'un descripteur de taglib (Tag Library Descriptor) et d'un ensemble de classes Java implémentant l'interface `JspTag`. Le descripteur de taglib (.tld) est un document XML qui décrit les associations entre les balises et la classe Java. Ces actions sont représentées dans la source JSP comme une balise XML. Lors de la compilation de la JSP, ces balises sont remplacées par des appels vers la classe Java correspondante.

3.8. Javascript :

JavaScript est un langage de programmation de scripts principalement utilisé pour les pages web interactives. C'est un langage orienté objets à prototype, c'est à dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas instanciés au sein de classes mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés. Le Javascript est une technologie persistante afin de réaliser des appels AJAX afin d'éviter des rafraichissements permanents des pages.

4. Conclusion :

Ce projet fut, pour nous, l'occasion de mettre en œuvre et d'approfondir nos connaissances en termes d'analyse et de programmation. Cela nous a permis de mettre en place une application web J2EE, et d'utiliser les technologies EJB, JMS Hibernate, Dojo... Des technologies qui semblent être porteuses sur le marché du travail à l'heure actuelle. L'application que nous avons développée fonctionne correctement et respecte les contraintes imposées.