

Documentation utilisateur V0

Sommaire

1. Présentation de l'application

1.1. Use Case global de l'application

1.2. Diagramme de classe de la base de données

2. Architecture

2.1. Architecture générale (poste client, serveurs, ...) et rôle de chaque élément

2.2. Ressources externes (.jar, ...) utilisées et rôles

2.3. Structuration en packages de l'application documentée. Principes retenus pour cette structuration

3. Fonctionnalités

3.1. Modifier les informations d'un client

3.2. Modifier les informations d'un client

3.3. Créer un nouveau client

3.4. Consulter un compte

3.5. Débitier un compte

3.6. Rendre inactif un client

1. Présentation de l'application

1.1. Use Case global de l'application



Dans le diagramme des Use Case ci-dessus, nous pouvons observer qu'il y a deux types d'utilisateurs présents dans l'application, le guichetier et le chef d'agence.

Le guichetier est l'employé "de base" du système. Il peut réaliser certaines opérations:

- Modifier informations client (adresse, téléphone, ...)
- Créer un nouveau client
- Consulter un compte
- Débitier un compte (BD)

Le chef d'agence est un type d'utilisateur héritant de guichetier, il peut effectuer toutes les opérations de celui-ci en plus de pouvoir rendre inactif un client.

1.2. Diagramme de classe de la base de données



- Une agence bancaire est composée de plusieurs employés.
- Un employé peut être chef d'agence de une ou plusieurs agences.
- Un employé peut être un chef d'agence ou un guichetier selon les droits d'accès qu'il possède
- Une agence bancaire est composée de plusieurs clients.
- Un client possède des informations qu'il est possible de modifier.
- Un client peut devenir inactif.

- Un client peut effectuer des opérations sur un compte par l'intermédiaire des employés (débit, crédit, emprunt).
- Un compte contient des informations auxquelles il est possible d'accéder.
- Un client peut avoir plusieurs comptes.
- Un emprunt peut être assuré.

Pour l'instant l'application ne possède que certaines fonctionnalités.

2. Architecture

2.1. Architecture générale (poste client, serveurs, ...) et rôle de chaque élément



Les postes client exécutent l'application faite en javaFX et se connectent à la base de données oracle, permettant de manipuler les données des clients et des employés.

2.2. Ressources externes (.jar, ...) utilisées et rôles

L'application utilise la librairie JavaFX pour permettre de disposer d'une interface graphique.

Elle utilise également la librairie externe ojdbc6.jar pour interagir avec la base de données du système des agences bancaires et exécuter des instructions SQL avec Java.

2.3. Structuration en packages de l'application documentée. Principes retenus pour cette structuration

L'application est structurée en plusieurs packages. Le modèle retenu est le modèle MVC (modèle, view, controller). De plus on regroupe les packages en deux catégories:

Application :

- Package tools : Contient contenant les outils de l'application.
- Package view : Contient les classes de l'interface graphique
- Package control : Contient les fonctionnalités de l'application

Cette catégorie contient les données des différentes classes pour pouvoir utiliser les différentes fonctionnalités de l'application.

Model :

- Package data : Contient les classes de données.
- Package orm : Contient les classes de gestion des données.
- Package orm.exception : Contient les classes d'exceptions.

Il contient des classes de base de données correspondant à certaines tables de la base de données. Il permet également de manipuler les données de la base de données (classe orm).

3. Fonctionnalités

3.1. Modifier les informations d'un client

Partie "modifier infos client" du diagramme des use case

Partie du diagramme de classes données nécessaires : en lecture, en mise à jour

Copies écrans principales de la fonctionnalité (ou renvoi vers doc utilisateur) + maquettes états imprimés (si concerné)

image::images/dsmodifclient.png

3.2. Modifier les informations d'un client

3.3. Créer un nouveau client

Partie "créer un nouveau client" du diagramme des use case

Partie du diagramme de classes données nécessaires :

- en lecture :
- en mise à jour : Client

Copies écrans principales de la fonctionnalité (ou renvoi vers doc utilisateur) + maquettes états imprimés (si concerné)

image::images/dsmodifclient.png

3.4. Consulter un compte

3.5. Débitier un compte

3.6. Rendre inactif un client

Last updated 2022-05-22 13:11:54 +0200