Jonathan Tremblay (*2095255)*

**Mon projet de fin de session : JoJo Bank**

Travail présenté à

Mr. Hafed Benteftifa

Dans le cadre du cours

420-A07-BB

Programmation par composants

Groupe 01201

Collège Bois-de-Boulogne

Le 7 février 2021

Table des matières

[Introduction 3](#_Toc63626126)

[Description des classes 3](#_Toc63626127)

[Package « modele » 3](#_Toc63626128)

[Package « ui » 4](#_Toc63626129)

[Package « utils » 5](#_Toc63626130)

[Copie d’écran des sorties de l’application 6](#_Toc63626131)

[Fenêtre de connexion 6](#_Toc63626132)

[Fenêtre d’inscription 7](#_Toc63626133)

[Fenêtre menu principal 9](#_Toc63626134)

[Fenêtre d’ajout de transactions 10](#_Toc63626135)

[Fenêtre liste complète de transactions 12](#_Toc63626136)

[Fenêtre d’ajout et liste de factures 14](#_Toc63626137)

[Fenêtre – Qui doit comment à qui ? 17](#_Toc63626138)

[Conclusion 18](#_Toc63626139)

[Annexe I – Diagramme de classe de JoJo Bank 19](#_Toc63626140)

[Tests - Informations de connexion 21](#_Toc63626141)

# Introduction

Depuis que ma copine et moi habitons sous le même toit, j’ai dû apprendre à gérer le suivi des transactions communes. Au départ, ça n’allait pas très bien. C’est pourquoi j’ai décidé de réaliser une application me permettant de garder la trace des transactions faites, par personne. De cette façon, nous pouvons calculer facilement qui doit combien à qui. Évidemment, un bon fichier Excel fais le même travail… mais pourquoi pas se pratiquer à la conception de petits projets ?

# Description des classes

## Package « modele »

**Classe : Client**

Définit ce qu’est un client pour l’application. Chaque client possède un nom, un « username » et un mot de passe.

**Classe : Facture**

Définit ce qu’est une facture pour l’application. Par facture, on entend ici des paiements mensuels, ou encore une dette (facture unique). Chaque facture possède une description, le nom du créancier, le montant et la date limite (sous chaîne de caractères).

**Classe : Transaction**

Définit ce qu’est une transaction pour l’application. Par transaction, on entend ici des dépenses émises dans un commerce, par exemple. À ne pas confondre avec les factures. Chaque transaction possède une date (LocalDate), une catégorie et le nom du commerçant. Elle possède aussi des notes facultatives, le montant payé, un numéro de transaction (généré par le système automatiquement) ainsi que le Client avec lequel la transaction est partagée.

Par défaut, si on ne veut pas partager la transaction, on se la partage soi-même.

**Classe : ListeClient**

Définit ce qu’est une liste de client pour l’application. La liste est composée d’un TreeSet statique.

**Classe : ListeFacture**

Définit ce qu’est une liste de factures pour l’application. La liste est composée d’une ArrayList<Facture> statique. On y calcule aussi la somme des factures uniques, et la somme des factures mensuelle.

**Classe : ListeTransaction**

Définit ce qu’est une liste de transactions pour l’application. La liste est composée d’une ArrayList<Facture> statique.

**Classe : FactureTableModel**

Définit le modèle d’une JTable de Facture. Chaque rangée est donc une représentation d’un objet Facture.

**Classe : ExpenseTableModel**

Définit le modèle d’une JTable de Transaction. Chaque rangée est donc une représentation d’un objet Transaction.

## Package « ui »

**Classe : FenConnection**

C’est la classe qui définit le JFrame de la fenêtre de connexion à l’application. Accessible au démarrage de l’application, via la fenêtre d’inscription, et lorsque on se déconnecte depuis le menu principal.

**Classe : FenRegister**

C’est la classe qui définit le JFrame de la fenêtre d’inscription à l’application. Accessible via la fenêtre de connexion.

**Classe : FenAppUI**

C’est la classe qui définit le JFrame de la fenêtre offrant le menu principal, une fois connecté.

**Classe : FenAddExpense**

C’est la classe qui définit le JFrame de la fenêtre permettant d’ajouter des transactions à son compte. Accessible via le menu principal, et la fenêtre « FenExpenses ».

**Classe : FenExpenses**

C’est la classe qui définit le JFrame de la fenêtre permettant de voir la liste des transactions à son compte. Accessible via le menu principal, et la fenêtre « FenAddExpense ».

**Classe : FenBills**

C’est la classe qui définit le JFrame de la fenêtre permettant d’ajouter et voir les factures à son compte. Accessible via le menu principal seulement.

**Classe : FenOwing**

C’est la classe qui définit le JFrame de la fenêtre permettant de voir le sommaire des transactions partagées entre clients. Accessible via le menu principal seulement.

## Package « utils »

**Classe : FileManip**

Cette classe, composée entièrement de méthodes statiques, me permet de manipuler les fichiers de données de l’application.

**Classe : FormValidation**

Cette classe, composée entièrement de méthodes statiques, me permet d’organiser mes méthodes servant à valider la saisie de l’utilisateur dans divers espaces de l’application.

**Classe : UserSession**

Cette classe abstraite n’a qu’un seul but : sauvegarder l’objet Client du client qui se connecte à l’application. De cette façon, il m’est possible d’y avoir accès pour aller lire SA liste de factures, SA listes de transactions, etc. Je conserve aussi le nombre de transactions de ce même client, afin de générer facilement un identifiant de transaction, de manière automatique.

**Classe : ClientComparator**

Cette classe implémente la classe Comparator, afin de créer un Comparator de Client, pour éviter les doublons.

**Classe : ClientExistantException**

Cette classe étend la classe Exception, afin de créer une exception personnalisée pour des messages d’erreur de l’objet Client.

# Copie d’écran des sorties de l’application

## Fenêtre de connexion

Démarrage de la fenêtre

Graphical user interface, application

Description automatically generated

Champs de saisie vides

Graphical user interface

Description automatically generated

Mauvais identifiants

Graphical user interface, website

Description automatically generated

Lorsque la connexion est un succès, la fenêtre menu principal s’ouvre.

## Fenêtre d’inscription

Démarrage

Graphical user interface

Description automatically generated

Champs de saisies vides

Graphical user interface, website

Description automatically generated

Mot de passe faible

Graphical user interface

Description automatically generated

Inscription complétée

Graphical user interface, website

Description automatically generated

Ensuite, la fenêtre se ferme pour laisser place à la fenêtre de connexion.

## Fenêtre menu principal

Démarrage

A picture containing text, screen, display

Description automatically generated

## Fenêtre d’ajout de transactions

Démarrage

A screenshot of a video game

Description automatically generated

Date invalide

A screenshot of a video game

Description automatically generated with medium confidence

Montant invalide

A screenshot of a computer

Description automatically generated with medium confidence

Ajout de transaction complété

A screenshot of a video game

Description automatically generated

## Fenêtre liste complète de transactions

Démarrage

A screenshot of a computer

Description automatically generated

Effacer une transaction

A screenshot of a computer

Description automatically generated with medium confidence

Lorsque on clique sur le bouton …

A screenshot of a computer

Description automatically generated

## Fenêtre d’ajout et liste de factures

Démarrage

A screenshot of a computer

Description automatically generated with low confidence

Montant invalide

A screenshot of a video game

Description automatically generated with medium confidence

Ajout de facture complété

*Facture mensuelle*

A screenshot of a computer

Description automatically generated with medium confidence

*Facture unique*

*A screenshot of a computer

Description automatically generated with low confidence*

Effacer une facture

A screenshot of a video game

Description automatically generated with medium confidence

Lorsqu’on clique sur le bouton …

A screenshot of a video game

Description automatically generated with medium confidence

## Fenêtre – Qui doit comment à qui ?

Démarrage

A screenshot of a computer

Description automatically generated with medium confidence

Cas : l’utilisateur connecté doit de l’argent au « user » sélectionné

A screenshot of a video game

Description automatically generated with medium confidence

Cas : le « user » sélectionné doit de l’argent à l’utilisateur connecté

A screenshot of a video game

Description automatically generated with medium confidence

# Conclusion

Je suis plutôt fier du résultat donné de mon projet. Me considérant horrible en termes de design et art, je m’attendais à une interface plus simple. Hormis tout ça, j’aurais aimé avoir le temps d’ajouté le i18n à mon application, ou encore d’apprendre comment le faire avant de débuter le projet ! Cela aurait ajouter une belle touche finale, je trouve. Malgré tout, j’ai beaucoup apprécié mon parcours de conception de JoJoBank. Malgré avoir appris des notions essentielles au développement de tout bon logiciel, je ne les ai pas appliqués assez à mon avantage dès le début de conception. Je parle ici de diagrammes, ou encore tout simplement un plan global de comment je désire que l’application fonctionne.

En bref, je suis assez content du résultat final, qu’il ne restera pas final pour autant ! Je prévois l’améliorer, cette fois-ci en ajoutant le i18n, un tri des transactions, et une section contenant les statistiques de dépenses d’un client ! Je tiens à dire que la version actuelle encrypte vos données de connexion.

# Annexe I – Diagramme de classe de JoJo Bank

**Diagramme de classe du package « modele »**

*Aussi disponible dans le dossier « rapport » du projet JojoBank*

**Diagram

Description automatically generated**

**Diagramme de classe du package « ui »**

*Aussi disponible dans le dossier « rapport » du projet JojoBank*

**Polygon

Description automatically generated with medium confidence**

**Diagramme de classe du package « utils »**

*Aussi disponible dans le dossier « rapport » du projet JojoBank*

**Diagram

Description automatically generated**

# Tests - Informations de connexion

Utilisateur déjà présent : Jonathan et Zakiyyah

User : jones User : zaza

Mdp : Test123! Mdp : Test123!