

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
Description des classes	3
Package « modele »	3
Package « ui »	4
Package « utils ».....	5
Copie d'écran des sorties de l'application	6
Fenêtre de connexion	6
Fenêtre d'inscription.....	7
Fenêtre menu principal.....	9
Fenêtre d'ajout de transactions.....	10
Fenêtre liste complète de transactions	12
Fenêtre d'ajout et liste de factures.....	14
Fenêtre – Qui doit comment à qui ?	17
Conclusion.....	18
Annexe I – Diagramme de classe de JoJo Bank.....	19
Tests - Informations de connexion	21

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.

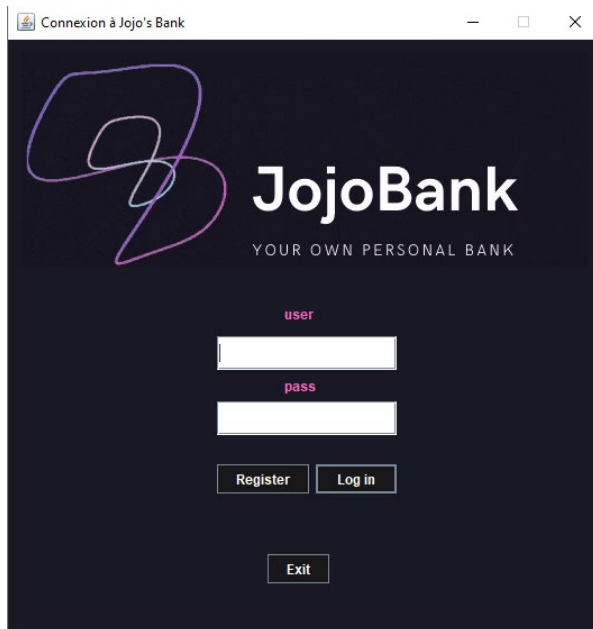
Classe : ClientExistantException

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

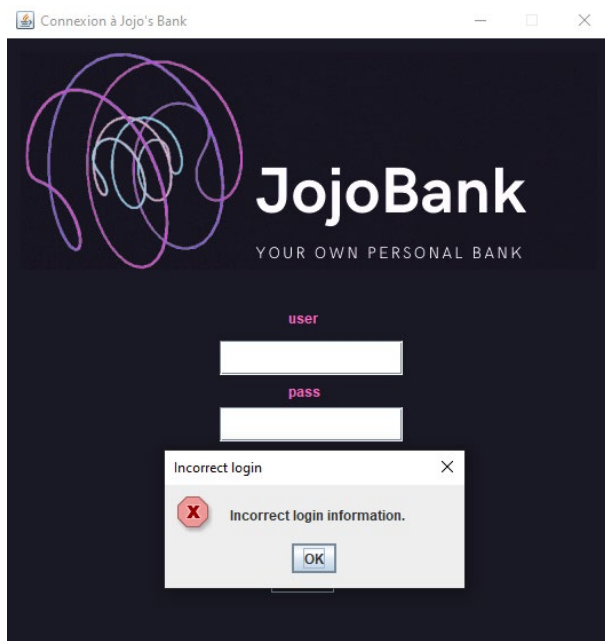
Copie d'écran des sorties de l'application

Fenêtre de connexion

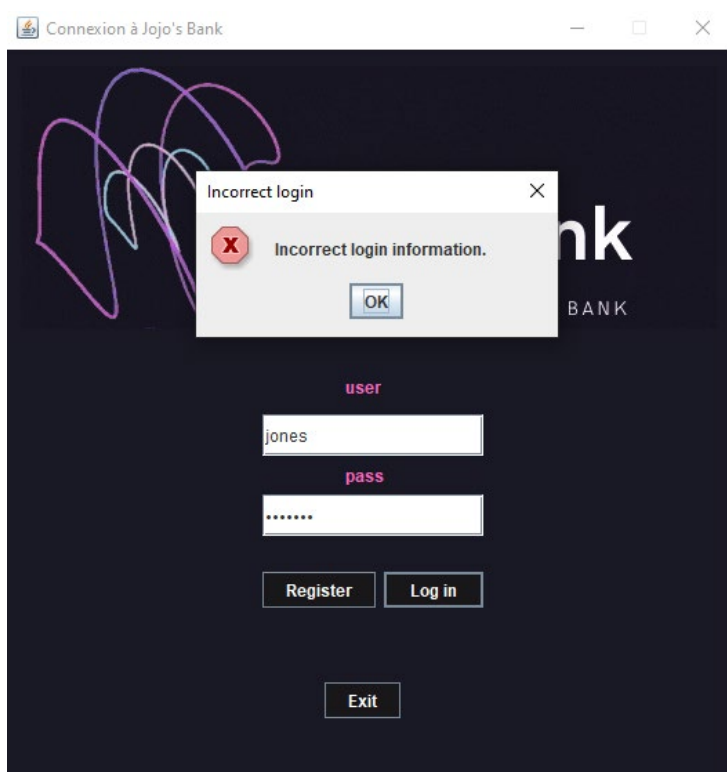
Démarrage de la fenêtre



Champs de saisie vides



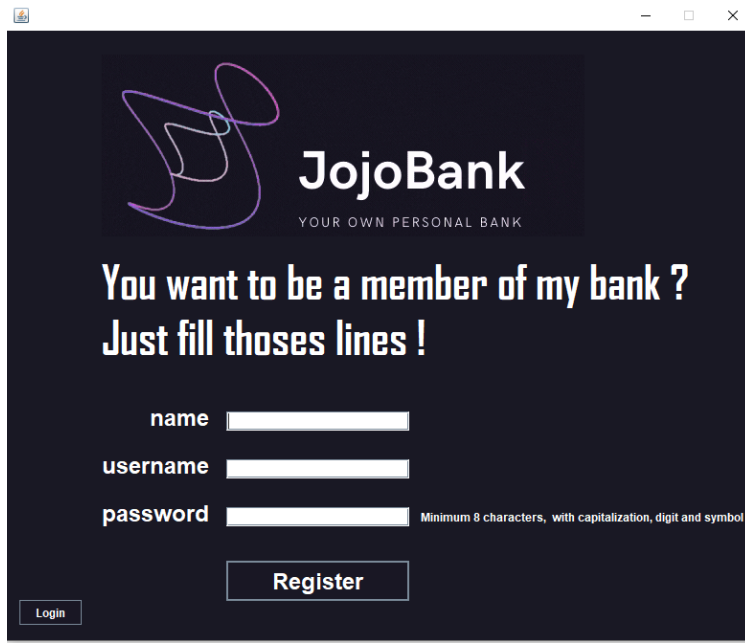
Mauvais identifiants



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

Fenêtre d'inscription

Démarrage



JojoBank
YOUR OWN PERSONAL BANK

You want to be a member of my bank ?
Just fill thoses lines !

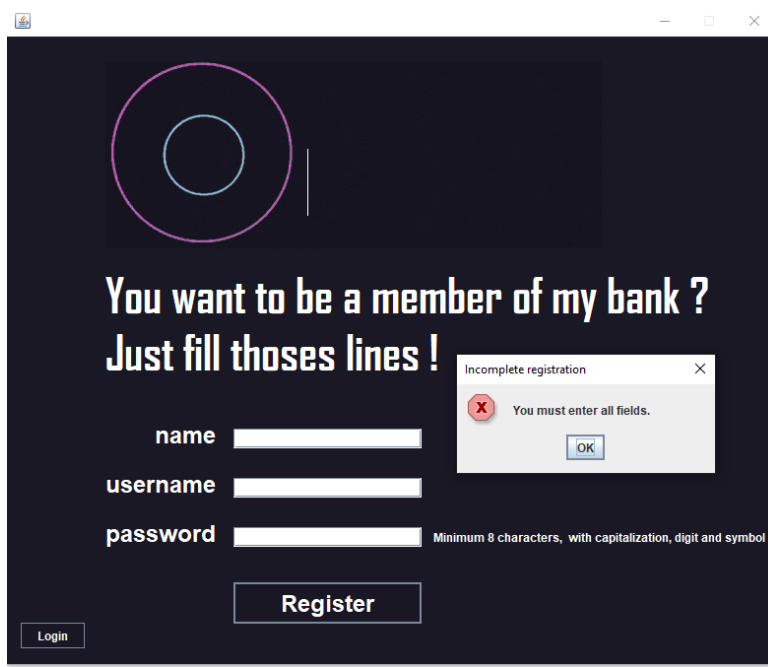
name

username

password Minimum 8 characters, with capitalization, digit and symbol

Login Register

Champs de saisies vides



JojoBank
YOUR OWN PERSONAL BANK

You want to be a member of my bank ?
Just fill thoses lines !

name

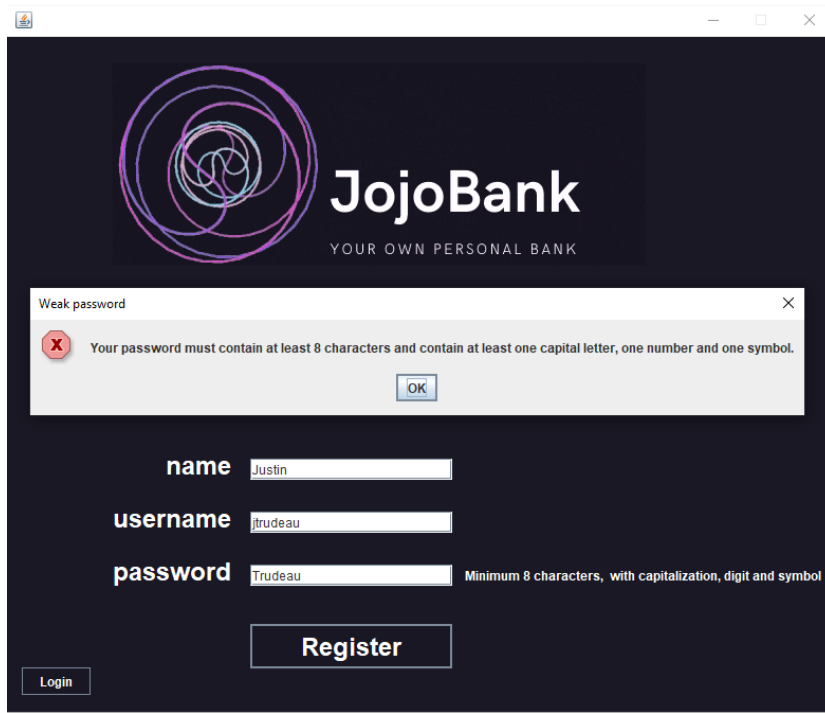
username

password Minimum 8 characters, with capitalization, digit and symbol

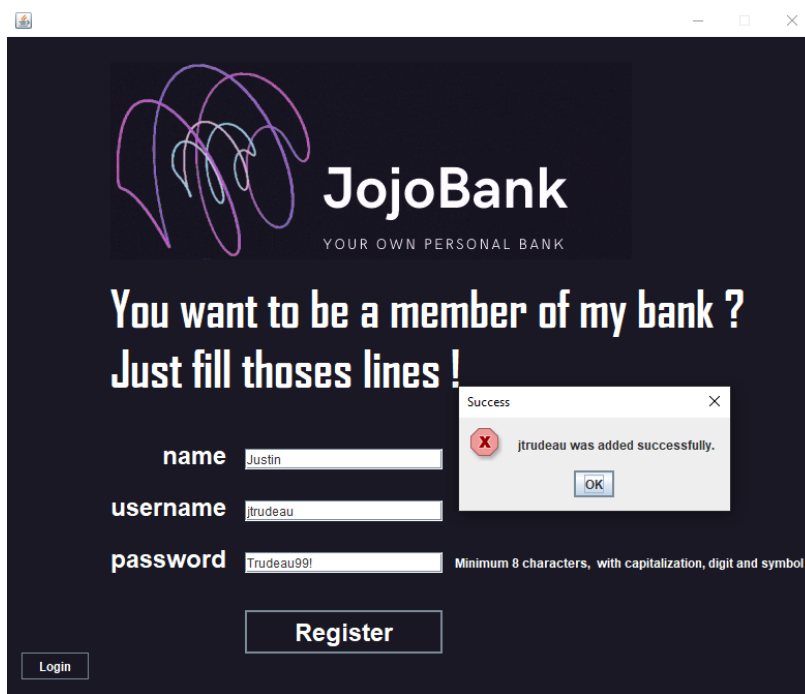
Login Register

Incomplete registration
You must enter all fields.
OK

Mot de passe faible



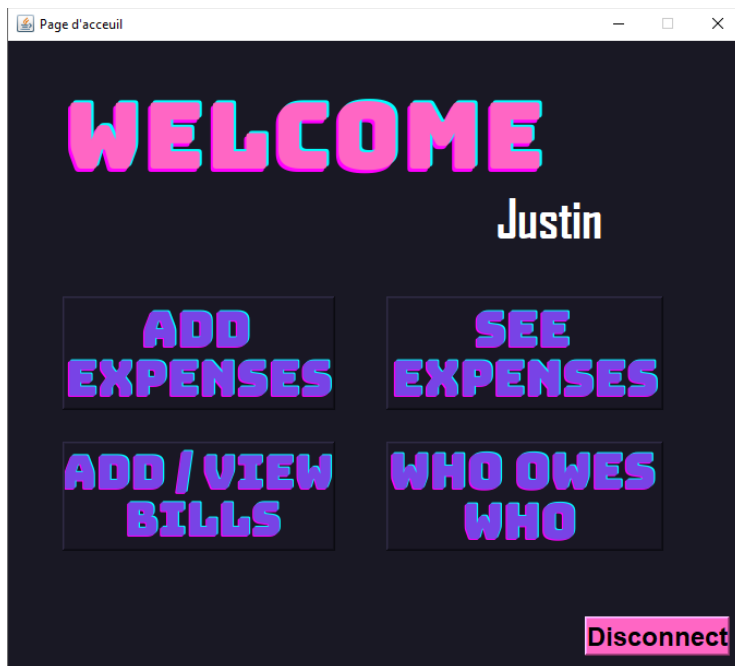
Inscription complétée



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

Fenêtre menu principal

Démarrage

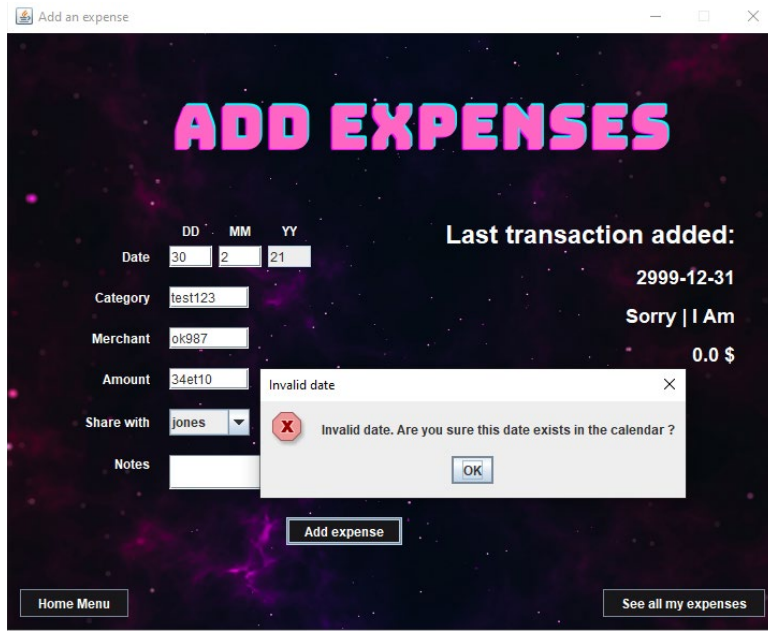


Fenêtre d'ajout de transactions

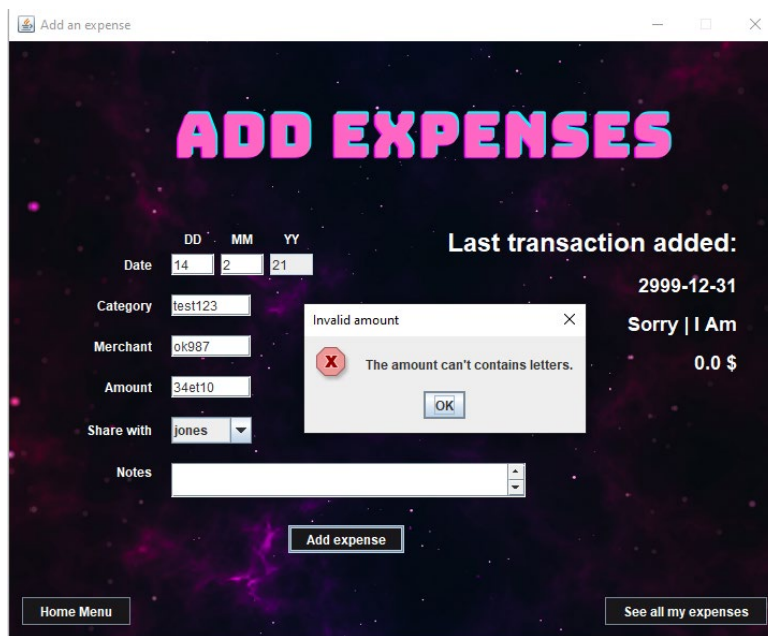
Démarrage

A screenshot of a web application window titled "Add an expense". The background is a dark, starry space theme. The main heading "ADD EXPENSES" is in large, bold, pink letters. On the left, there is a form with several input fields: "Date" (with sub-fields for DD, MM, and YY, where YY is set to 21), "Category", "Merchant", "Amount", "Share with" (a dropdown menu showing "jones"), and "Notes" (a text area). Below the form is a button labeled "Add expense". On the right side, there is a section titled "Last transaction added:" which displays the date "2999-12-31", the text "Sorry | I Am", and the amount "0.0 \$". At the bottom left, there is a button labeled "Home Menu", and at the bottom right, there is a button labeled "See all my expenses".

Date invalide



Montant invalide



Ajout de transaction complété

Add an expense

ADD EXPENSES

DD

MM

YY

Date

21

Last transaction added:

2021-02-14

ok987 | test123

34.1 \$

Category

Merchant

Amount

Share with

jones

Notes

Add expense

Home Menu

See all my expenses

Fenêtre liste complète de transactions

Démarrage

See expenses

MY EXPENSES

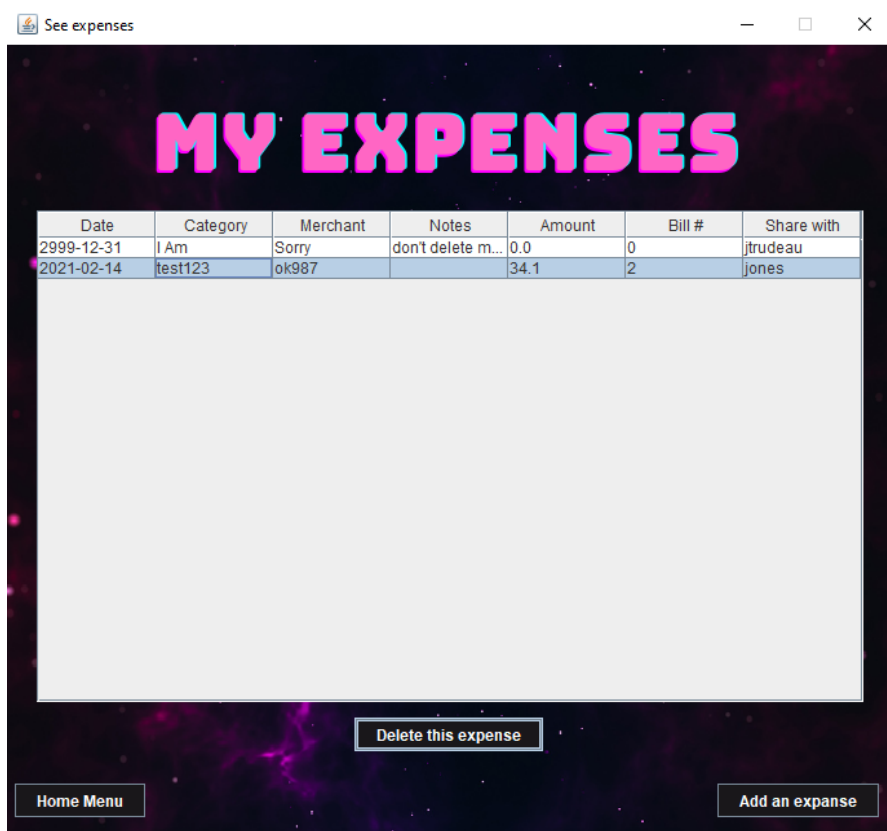
Date	Category	Merchant	Notes	Amount	Bill #	Share with
2999-12-31	I Am	Sorry	don't delete m...	0.0	0	jtrudeau
2021-02-14	test123	ok987		34.1	2	jones

Delete this expense

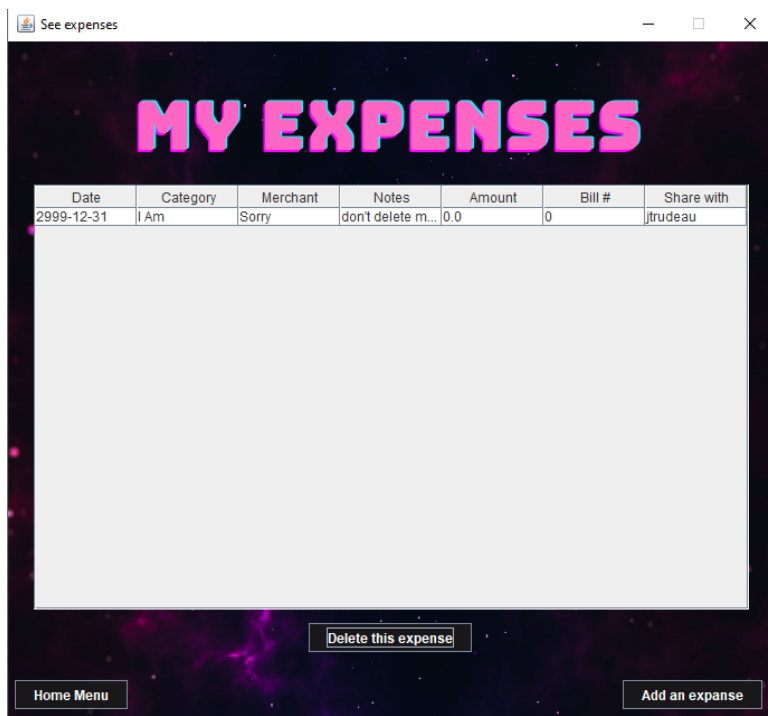
Home Menu

Add an expanse

Effacer une transaction



Lorsque on clique sur le bouton ...



Fenêtre d'ajout et liste de factures

Démarrage

The screenshot shows the 'MY BILLS' application window. At the top, there's a title bar with a small icon and the text 'Your bills'. Below the title bar, the main content area has a dark background with a purple nebula. The title 'MY BILLS' is displayed in large, pink, stylized letters. Below the title, there's a table with four columns: 'Creditor', 'Description', 'Due date', and 'Amount'. The table is currently empty. To the left of the table, there's a 'Delete row' button. To the right of the table, there's a 'Total monthly payment : 0.0 \$' label. Below the table, there's a large 'ADD BILLS' button. To the right of the 'ADD BILLS' button, there's a form with several input fields: 'Creditor', 'Description', 'Due date' (with sub-fields for DD, MM, and YY), and 'Amount'. There are also two radio buttons labeled 'Monthly bill' and 'Single invoice'. At the bottom left, there's a 'Home Menu' button, and at the bottom right, there's an 'Add bill' button.

Montant invalide

The screenshot shows the 'MY BILLS' application window with an error message displayed. The error message is a small dialog box with a red 'X' icon and the text 'Invalid amount' and 'The amount can't contains letters.' Below the error message, there's an 'OK' button. The background of the application window is the same as the previous screenshot, but the 'Amount' input field now contains the text '120et99'. The 'Add bill' button is still visible at the bottom right.

Ajout de facture complété

Facture mensuelle

🏠 Your bills

MY BILLS

Creditor	Description	Due date	Amount
Mastercard	Credit card	12 of the month	120.99

Delete row

Total monthly payment : **120.99**

Total debts : **0.0**

Creditor

Description

Due date
 DD MM YY

☒ Monthly bill
☐ Single invoice

Amount

Home Menu Add bill

Facture unique

🏠 Your bills

MY BILLS

Creditor	Description	Due date	Amount
Mastercard	Credit card	12 of the month	120.99
Hydro-Québec	Électricité	March 23th 2021	182.25

Delete row

Total monthly payment : **120.99**

Total debts : **182.25**

Creditor

Description

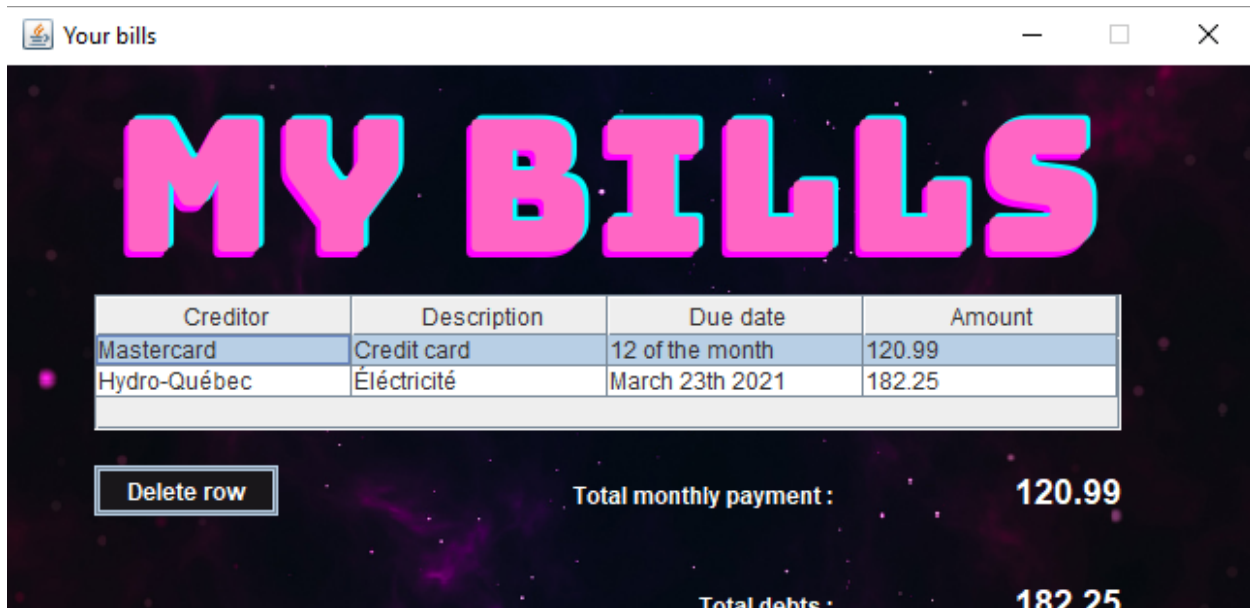
Due date
 DD MM YY

☒ Monthly bill
☐ Single invoice

Amount

Home Menu Add bill

Effacer une facture



The screenshot shows a window titled "Your bills" with a dark background and the text "MY BILLS" in large, stylized letters. Below the title is a table with four columns: Creditor, Description, Due date, and Amount. The table contains two rows: Mastercard (Credit card, 12 of the month, 120.99) and Hydro-Québec (Électricité, March 23th 2021, 182.25). A "Delete row" button is located below the table. To the right of the table, the "Total monthly payment" is displayed as 120.99, and the "Total debts" is displayed as 182.25.

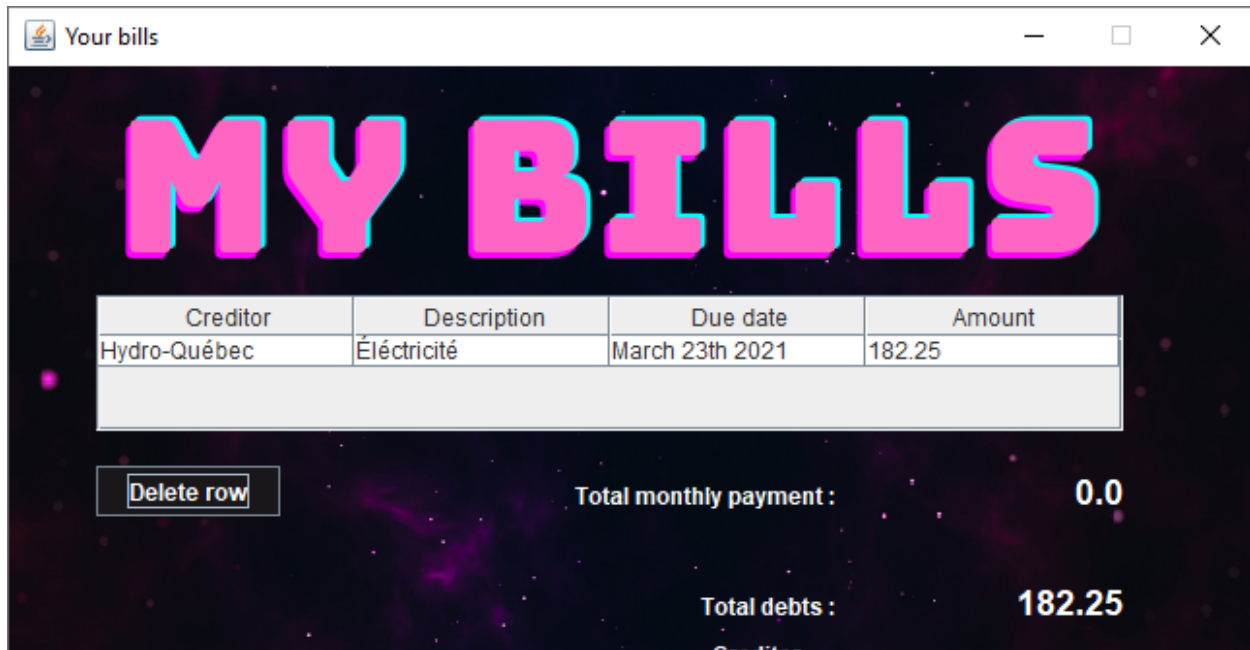
Creditor	Description	Due date	Amount
Mastercard	Credit card	12 of the month	120.99
Hydro-Québec	Électricité	March 23th 2021	182.25

Delete row

Total monthly payment : 120.99

Total debts : 182.25

Lorsqu'on clique sur le bouton ...



The screenshot shows the same "Your bills" window after the "Delete row" button has been clicked. The Mastercard bill has been removed from the table, leaving only the Hydro-Québec bill. The "Total monthly payment" is now 0.0, and the "Total debts" remains 182.25.

Creditor	Description	Due date	Amount
Hydro-Québec	Électricité	March 23th 2021	182.25

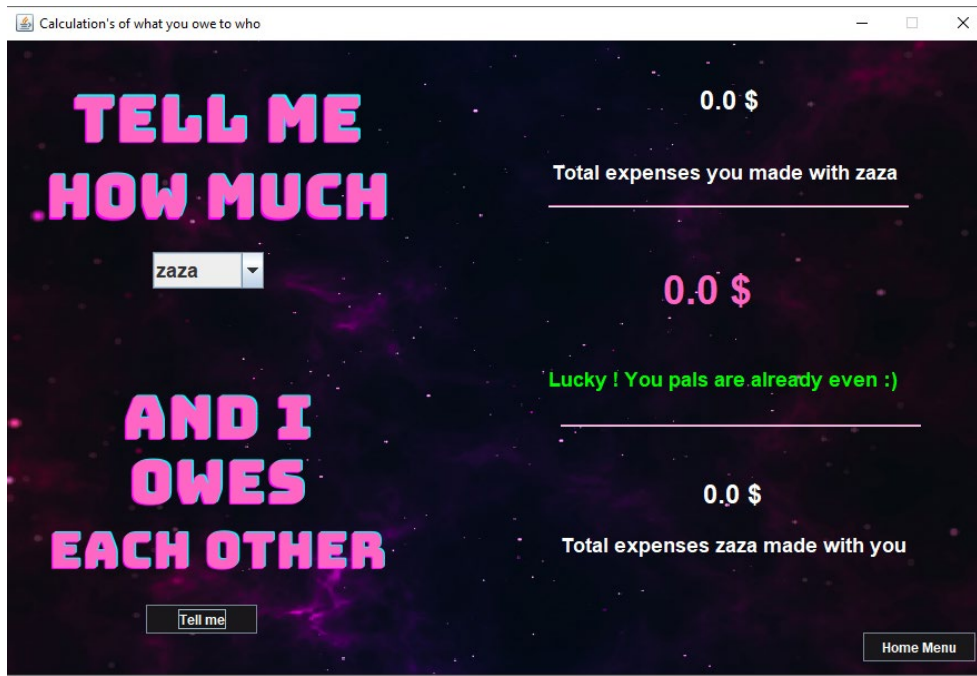
Delete row

Total monthly payment : 0.0

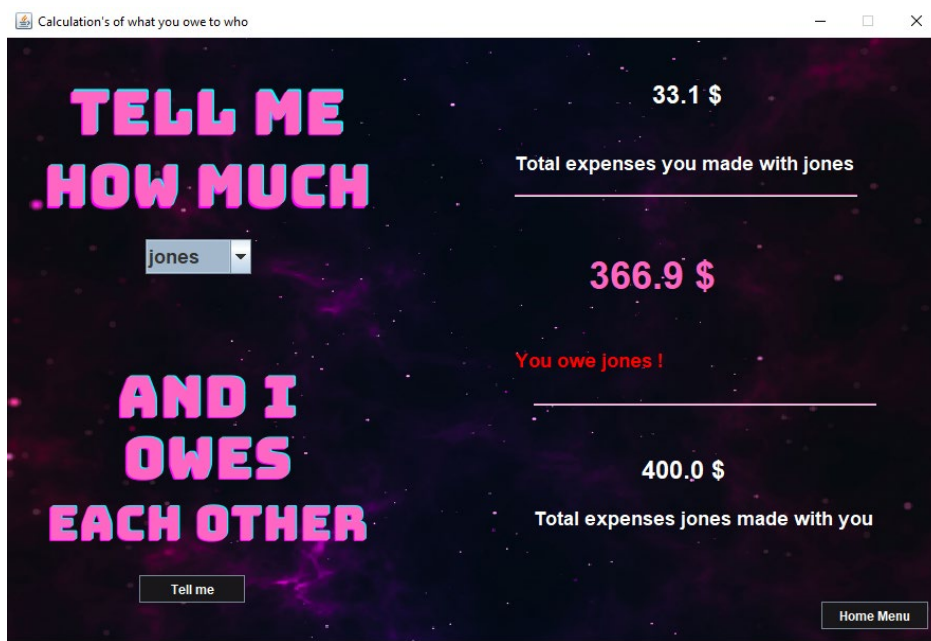
Total debts : 182.25

Fenêtre – Qui doit comment à qui ?

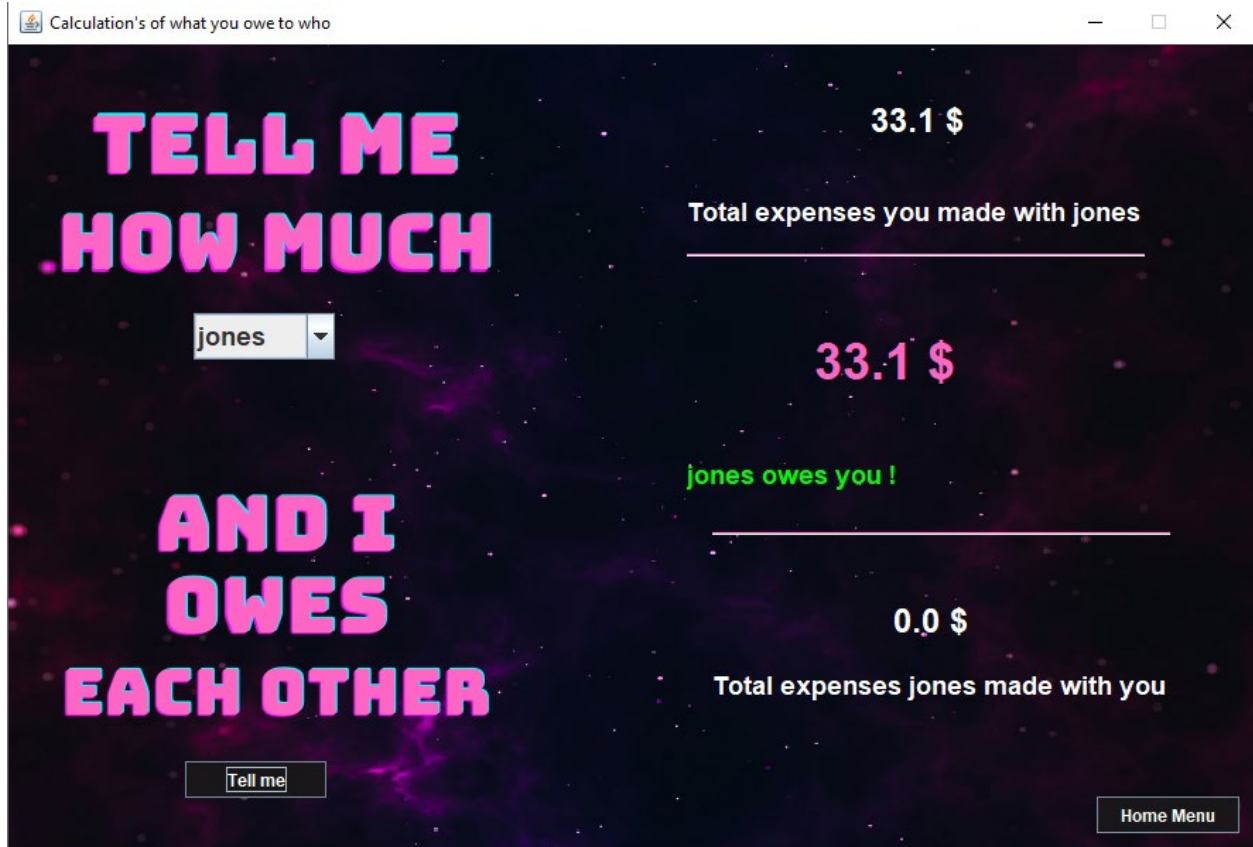
Démarrage



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



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



Conclusion

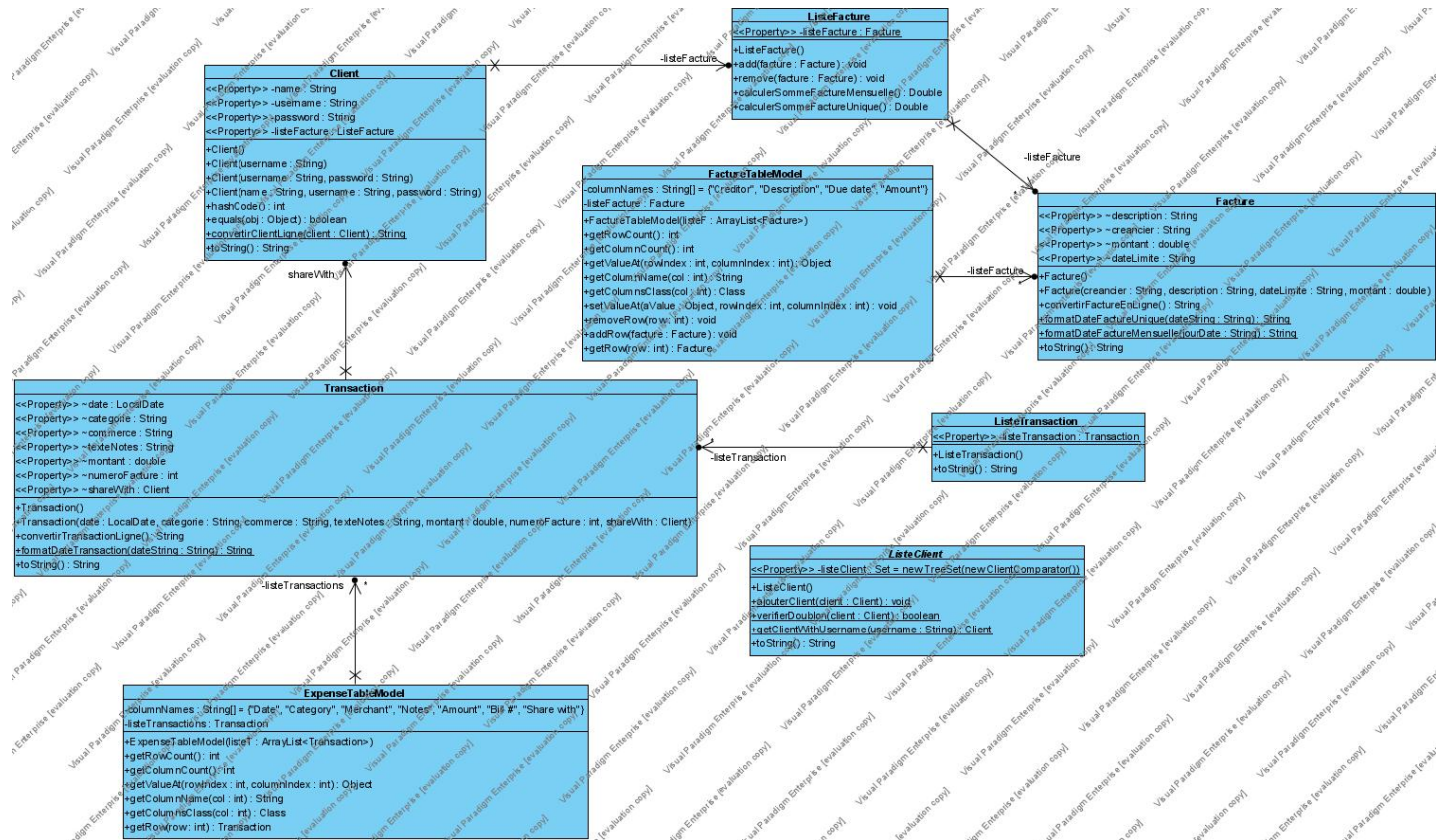
Je suis plutôt fier du résultat donné de mon projet. Me considérant horrible en termes de design et art, je m'attendais à une interface plus simple. Hormis tout ça, j'aurais aimé avoir le temps d'ajouter le i18n à mon application, ou encore d'apprendre comment le faire avant de débiter le projet ! Cela 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

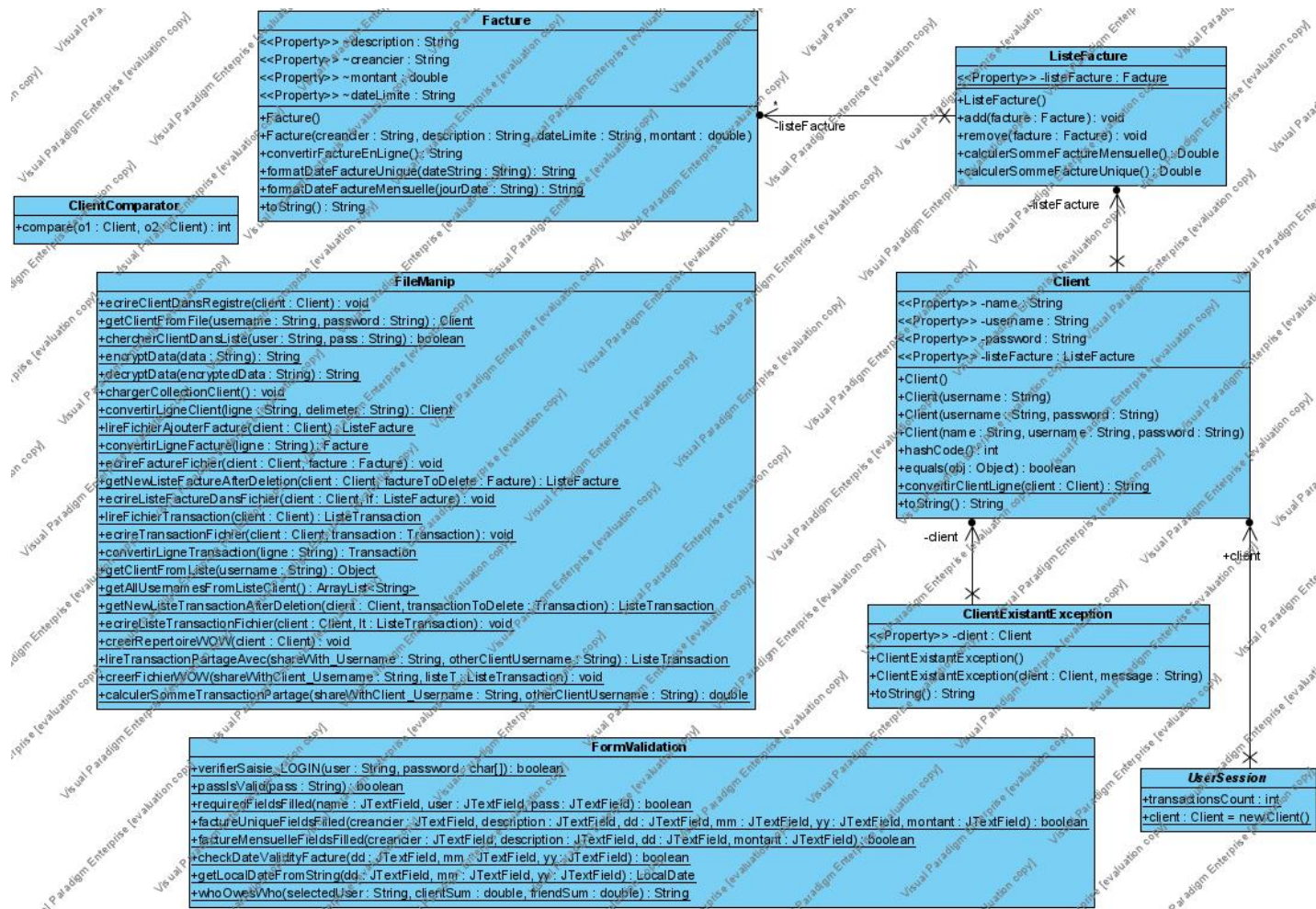


Aussi disponible dans le dossier « rapport » du projet JojoBank



Diagramme de classe du package « utils »

Aussi disponible dans le dossier « rapport » du projet JojoBank



Tests - Informations de connexion

Utilisateur déjà présent : Jonathan et Zakiyyah

User : jones

User : zaza

Mdp : Test123!

Mdp : Test123!