

Génie Logiciel

Team : Joëlle.T, Herlyne.M, Marc.E, Armel.M, Daël.M

19 June 2020



Contents

1	Introduction	2
2	Présentation de l'équipe	2
3	Répartition des tâches	2
4	Présentation du projet	2
4.1	Diagramme de cas d'utilisations	3
4.2	Diagramme de classe	3
4.3	Diagramme séquentiel	4
5	Problèmes rencontrés	4
6	Réalisations	5
6.1	Réalisations front-end	5
6.2	Réalisations back-end	8
7	Github	8
8	Conclusion	9

1 Introduction

Dans le cadre du projet INF3133 - Génie Logiciel, il nous a été demandé par notre professeur M.Swaenepoel, de réaliser un logiciel avec interface utilisateur et base de données. Ce projet doit être réalisé sous python et nous faire développer des compétences en développement orienté objet python et son interface Tkinter, découvrir les fonctionnalités de l'API et nous permettre de manager une équipe grâce à Github.

2 Présentation de l'équipe

Nous sommes une équipe d'étudiants composée de 5 personnes. Nous avons décidé de collaborer ensemble car nous étions motivés à travailler ensemble et progresser dans le domaine du développement logiciel.

3 Répartition des tâches

Joëlle : Chef de projet - Responsable Git

Herlyne : Développement Python - Responsable Full Stack

Armel : Interface Tkinter / Designer UI - Responsable Front-end

Daël : Base de données PHP - Responsable Back-End

Marc : Responsable logiciel / UML, diagrammes

4 Présentation du projet

L'objectif de notre projet est de réaliser la gestion d'une banque. Cette banque est nommée EsieaBanking. Nous avons plusieurs interfaces utilisateurs afin qu'un client puisse créer et se connecter à son compte. L'utilisateur peut également ajouter ou enlever de l'argent son compte. Plusieurs clients sont gérés au sein de la banque grâce à une base de données.

4.1 Diagramme de cas d'utilisations

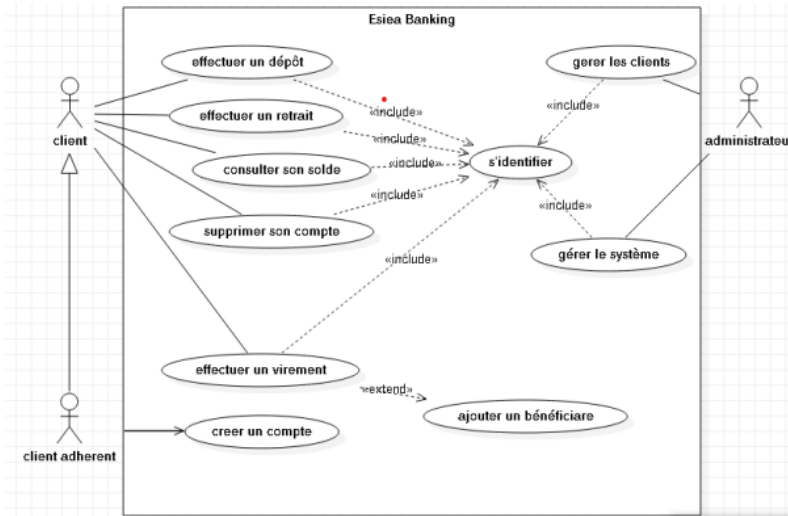


Diagramme de cas d'utilisations.

4.2 Diagramme de classe

Voici le diagramme de classe représentant les différentes étapes de création du projet.

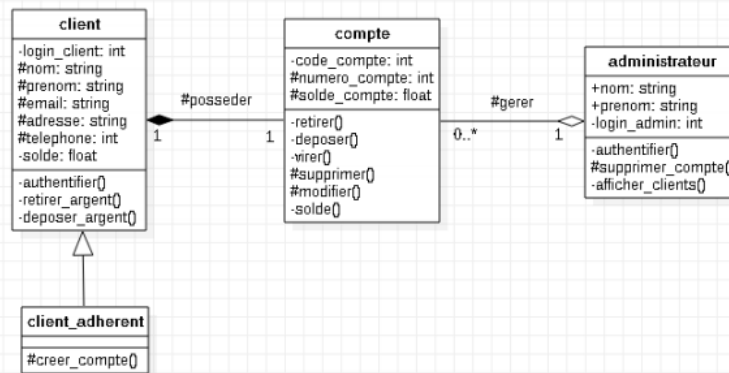
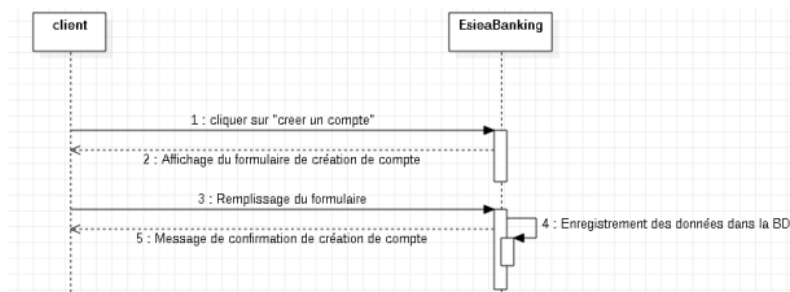
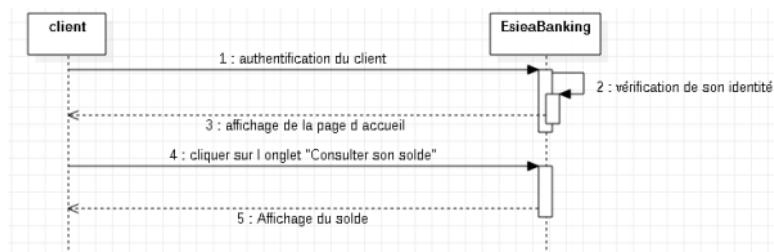


Diagramme de classe.

4.3 Diagramme séquentiel



Cette partie permet de créer un compte utilisateur.



Cette partie sequentielle permet d'afficher son solde bancaire.

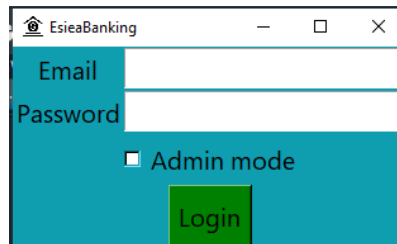
5 Problèmes rencontrés

Les problèmes importants que nous avons rencontrés sont :

1. La contrainte de temps car le projet a commencé le 29/05/20 et se termine le 19/06/20.
2. L'organisation des tâches car au vu du nombre important d'étudiants dans le groupe, nous n'avons pas pu gérer convenablement les horaires de disponibilités. Notre groupe s'est cependant réalisé en fonction des différentes affinités des membres.
3. Les connaissances en programmation orientée en Python.
4. Nous n'avons pas pu correctement exploiter notre base de données puisque nous avons utilisé PHPMyAdmin mais que nous n'avons pas réussi à insérer les données. Nous avons donc décidé de nous rabattre sur une BDD sql lite comme vu en cours. Néanmoins, l'insertion d'un nouveau client n'est pas encore fonctionnel. Pour remédier à cela, nous avons créé des utilisateurs de base qui peuvent être également utilisés en tant qu'administrateurs.

6 Réalisations

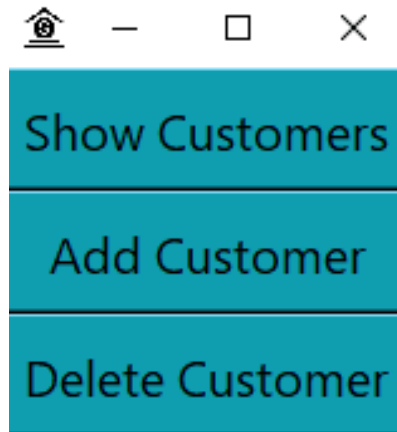
6.1 Réalisations front-end



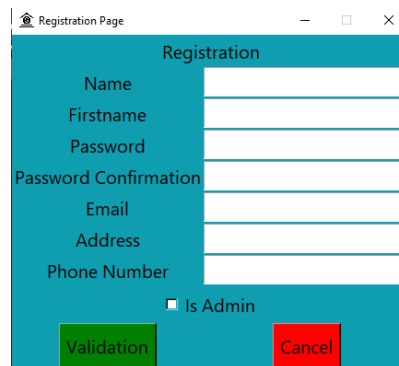
Nous avons réalisé une interface d'accueil permettant à l'utilisateur de se connecter avec son adresse mail et son mot de passe. Une fois connecté, l'utilisateur peut déposer ou retirer de l'argent de son compte.

Si l'on est en mode administrateur, on peut visualiser les clients, créer un nouveau client ou supprimer un client.

Une fois qu'il appuie sur le bouton login, il peut avoir accès à un menu permettant de choisir ce qui intéresse l'administrateur (/! il faut être en mode administrateur pour voir la fenêtre ci-dessous).



Lorsqu'on appuie sur add customer, cela permet de créer un nouveau client. On accède ainsi à l'interface ci-dessous. On peut également décider si le client est un administrateur ou non.



A screenshot of a web application window titled "Registration Page". The window has a blue header bar with the title "Registration". Below the header, there are several input fields for user registration: "Name", "Firstname", "Password", "Password Confirmation", "Email", "Address", and "Phone Number". Each field is represented by a text label on the left and a white input box on the right. At the bottom of the form, there is a checkbox labeled "Is Admin" and two buttons: a green "Validation" button and a red "Cancel" button.

Une fois un nouveau client crée, on peut déposer, retirer, consulter et supprimer son compte.

Pour accéder à cette partie, nous avons créer des clients de base :

1) Adresse mail : Alice@gmail.com

Mot de passe : Bob

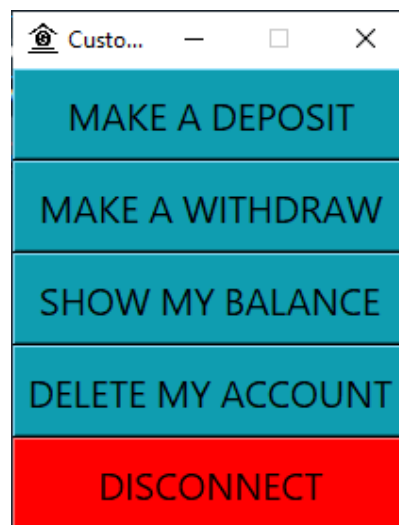
2) Adresse mail : Bob@gmail.com

Mot de passe : Alice

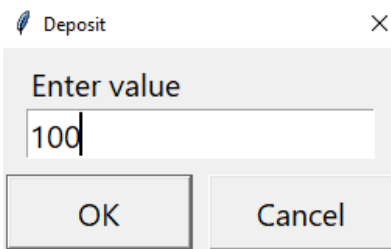
3) Bobi@gmail.com

Mot de passe : Alice

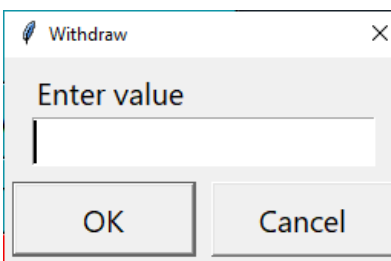
Attention : Il ne faut pas être en mode administrateur pour avoir accès à cette vue.



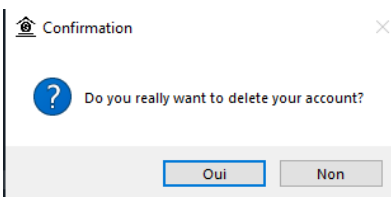
A screenshot of a web application window titled "Custo...". The window has a blue header bar with the title "Custo...". Below the header, there are five buttons stacked vertically. The first four buttons are blue with white text: "MAKE A DEPOSIT", "MAKE A WITHDRAW", "SHOW MY BALANCE", and "DELETE MY ACCOUNT". The fifth button is red with white text: "DISCONNECT".



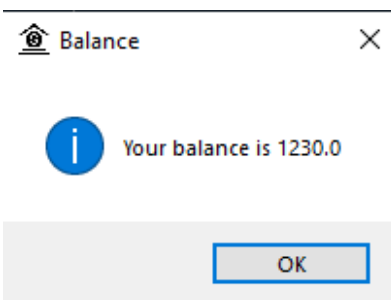
Ici par exemple, nous mettons 100 euros sur le compte de Alice.



Les interfaces suivantes permettent le retrait ou le dépôt d'une somme souhaitée par l'utilisateur. Nous n'avons pas eu le temps de modifier les couleurs, mais cela fait partie d'une amélioration possible.

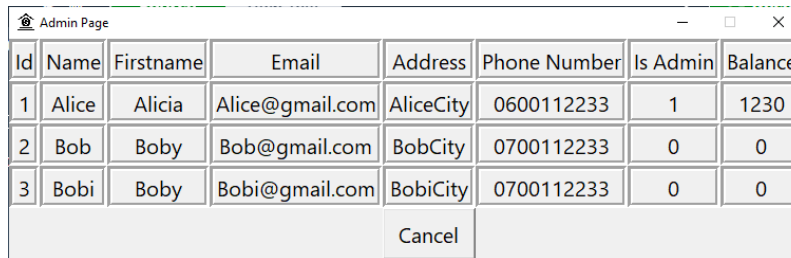


Nous pouvons supprimer également le compte d'un client.



Nous pouvons aussi afficher le montant de l'utilisateur en fonction de la somme ajoutée, modifiée ou supprimée.

6.2 Réalisations back-end



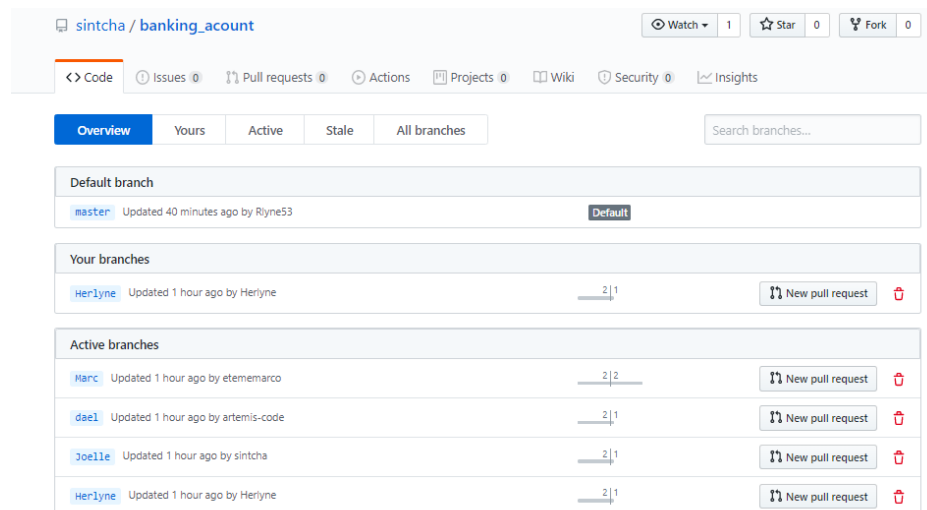
Id	Name	Firstname	Email	Address	Phone Number	Is Admin	Balance
1	Alice	Alicia	Alice@gmail.com	AliceCity	0600112233	1	1230
2	Bob	Boby	Bob@gmail.com	BobCity	0700112233	0	0
3	Bobi	Boby	Bobi@gmail.com	BobiCity	0700112233	0	0

Cancel

Voici la base de données que nous avons pu créer pour enregistrer l'ensemble des informations des clients. Nous avons opté pour ce type de base de données car nous n'avons pas pu gérer en local les données via PHPMyAdmin.

Néanmoins, nous avons travaillé sur PHPMyadmin pour créer notre base de données. Nous avons réalisé un MCD sur Jmerise afin de modéliser le fonctionnement de notre base de données en back-end.

7 Github



sintcha / banking_account

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights

Overview Yours Active Stale All branches Search branches...

Default branch

master Updated 40 minutes ago by Rlyne53 Default

Your branches

Herlyne Updated 1 hour ago by Herlyne 2/1 New pull request

Active branches

Marc Updated 1 hour ago by etememarco 2/2 New pull request

dse1 Updated 1 hour ago by artemis-code 2/1 New pull request

Joelle Updated 1 hour ago by sintcha 2/1 New pull request

Herlyne Updated 1 hour ago by Herlyne 2/1 New pull request

Voici les différentes branches que nous avons pu réaliser. Au total, nous avons créé 6 branches différentes pour pousser l'avancement de notre projet et nous avons poussé notre version finale sur la branche master.

8 Conclusion

En conclusion, on peut dire que ce projet nous a permis de découvrir davantage l'univers python et de gagner des compétences techniques dans ce domaine. Nous avons également pu ensemble aborder les notions de Github et réaliser un travail de groupe et ainsi acquérir des compétences humaines grâce à ce projet.

