

Projet Java

Rapport Final

L'application de gestion de l'argent

Classe : 18VP

Groupe 6:

Đoàn Nam Thuận – 18126034

Lê Thị Anh Thi – 18126033

Phạm Nguyễn Nhật Khan – 18126018

Tableau de contenu

L'histoire de versions	3
Introduction	4
Programmation orienté objet.....	5
Java IO/ Base de données	18
Java GUI	18
Démonstration	36
Résultat	60
Références.....	Error! Bookmark not defined.

L'histoire de versions

Date	Version	Description	Auteur
06/11/2020	V0.1	Interface de conception et interface de classe	Thuan
03/04/2021	V0.2	JavaIO/Base de données	Thuan
02/05/2021	V0.3	Interface d'application	Thuan, Thi, Khan
16/05/2021	V0.4	Final	Thuan, Thi, Khan

Introduction

Vous trouverez ci-dessous le rapport final du projet de cours avec les principaux éléments suivants:

- 1. Comment organiser la structure de la POO dans le projet sujet.*
- 2. Décrivez la structure des tables dans le modèle relationnel.*
- 3. Description graphique du projet final.*

Programmation orienté objet

I. Introduction

- *Le contexte du projet.*

Dans la vie de tous les jours, la gestion financière personnelle est très importante non seulement pour le présent, mais aussi pour nos projets d'avenir. Si on ne sait pas comment gérer ce qu'on a bien, nos vies n'apporteront évidemment aucune valeur ajoutée.

La planification et la gestion financières personnelles sont extrêmement importantes, mais ce n'est pas facile à faire. Il ne s'agit pas seulement de payer les factures à temps, mais aussi de déterminer combien dépenser et quoi dépenser.

Alors, comment gérer et planifier efficacement les finances personnelles?

Pour rendre ce travail plus efficace et inspirant, il est essentiel de créer une application d'appui qui nous aide à planifier et à allouer des revenus pour des dépenses raisonnables. C'est à la fois pratique et beaucoup plus efficace que la prise de notes traditionnelle.

- *Les raisons pour développer ce projet.*

✓ Ce sujet répond aux besoins pratiques de nombreuses personnes. Cela apporte de nombreux avantages dans la vie.

✓ Ce sujet répond aux exigences du projet :

1. L'application doit avoir un interface graphique en utilisant Swing
2. L'application doit charger et sauvegarder les données en utilisant Java IO (XML, JSON, etc.) ou un système de gestion de base de données.
3. L'application doit être désignée et implémentée en utilisant OOP.

- *Les applications/ressources existants et les fonctionnalités fournies par ces systèmes.*

Actuellement, il existe un certain nombre d'applications pour aider les utilisateurs dans la gestion financière personnelle. Mais l'utilisation de ces applications est assez difficile en raison de l'interface, de son fonctionnement, ...

- ***Expliquer les différences prévues entre le notre et l'application existante.***

Différente de ces applications, notre application aide les utilisateurs à gérer leurs finances personnelles en utilisant une méthode de gestion simple mais très efficace - la méthode des 6 pots - une des règles de la gestion des finances personnelles choisie et appliquée par de nombreuses personnes qui réussissent, y compris des millionnaires et des milliardaires mondiaux. Selon le principe des 6 pots, les revenus de l'utilisateur sont répartis dans 6 pots en fonction des 6 besoins humains nécessaires, y compris: l'essentiel, l'épargne, l'investissement, l'éducation, la jouissance et la compassion dans un pourcentage de 55-10-10-10-10-5 respectivement.

Avec cette méthode, l'application attribuera automatiquement les dépenses des utilisateurs en conséquence. Il est à la fois simple, facile à utiliser et extrêmement efficace dans la gestion des finances personnelles des utilisateurs.

Les principales fonctionnalités de notre application:

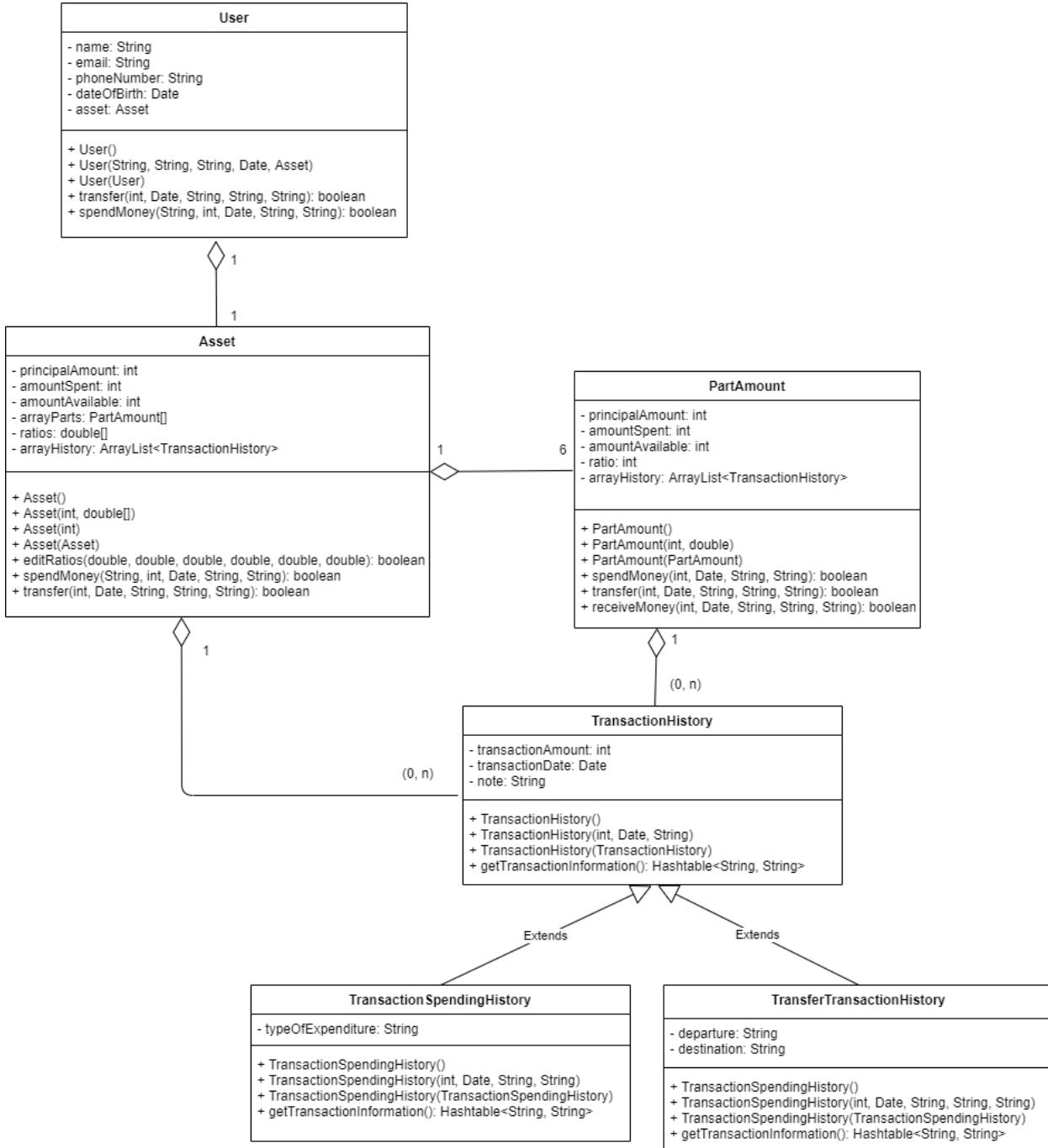
- ✓ Allouez automatiquement les revenus à chaque pot en fonction du taux de pré-commande.
- ✓ Prendre note - Modifier - Supprimer la transaction.
- ✓ Voir le solde actuel de chaque pot.
- ✓ Afficher l'historique des dépenses.
- ✓ Afficher les notes de frais avec des graphiques de visualisation.

- ***Présenter le résultat prévu de ce projet.***

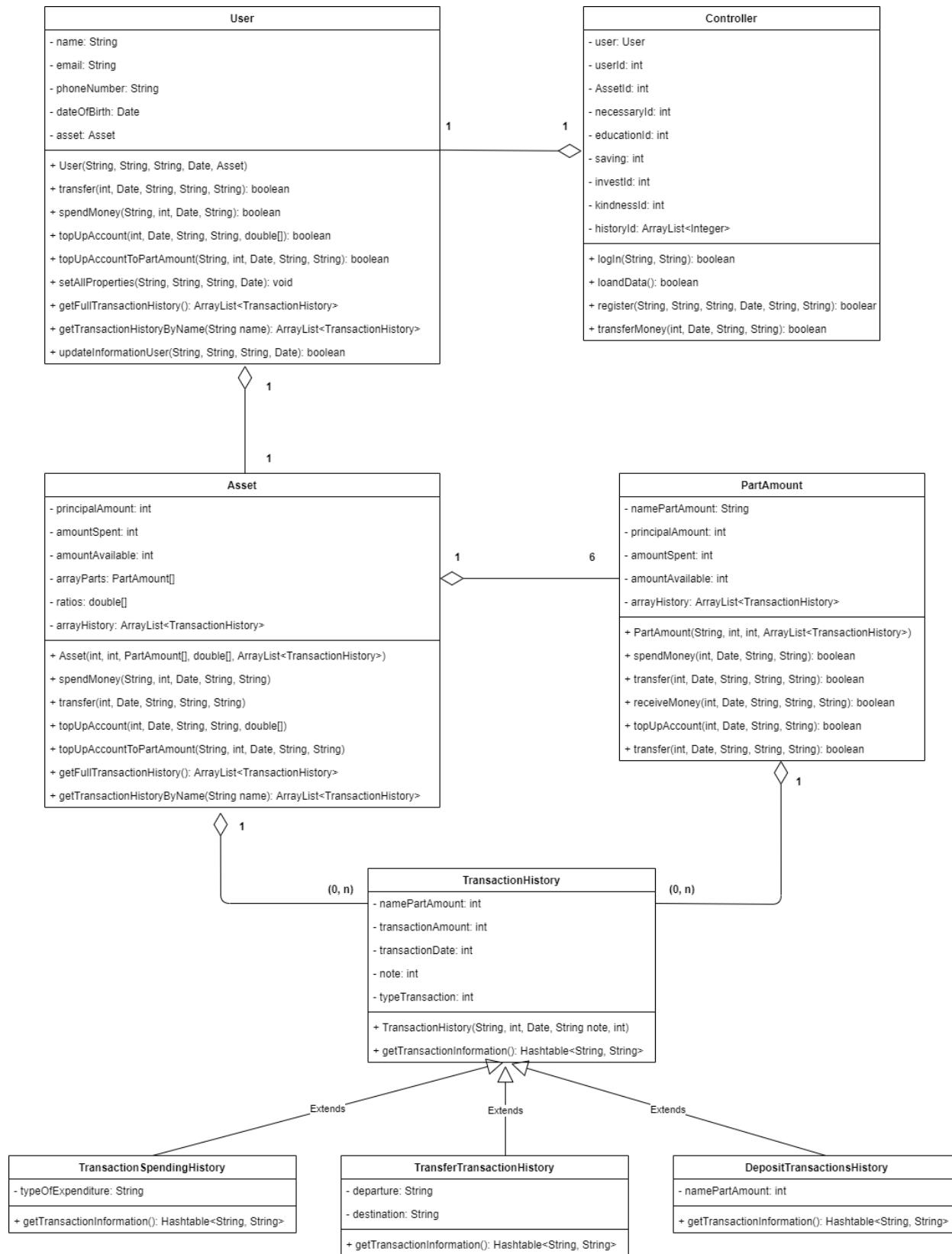
- Répondre à toutes les exigences du projet.
- Remplir toutes les fonctions et non-fonctions de l'application.
- Il a une large applicabilité dans la pratique.

II. Analyse et conception

Le diagramme de classes pour montrer l'organisation de notre code à implémenter en rapport 1



Le diagramme de classes pour montrer l'organisation de notre code à implémenter en rapport 2



Pour représenter l'historique des dépenses d'un utilisateur, nous avons 3 classes :

- *TransactionHistory*
- *TransactionSpendingHistory*
- *TransferTransactionHistory*
- *DepositeTransactionHistory*

Dans ce cas, les 2 classes suivantes(*TransactionSpendingHistory* et *TransferTransactionHistory*) héritent de la classe parente(*TransactionHistory*). Ils représentent deux types différents de transactions utilisateur comme transactions avec les consommateurs et transfert.

En plus, en rapport 2, nous avons ajouté un autre type de transaction. C'est l'opération consistant à ajouter de l'argent à un composant particulier d'un actif. La class **DepositeTransactionHistory** est ajoutée pour représenter ce type de transaction. Ce class hérite de la classe *TransactionHistory*.

La classe **PartAmount** représente chaque type de dépenses des utilisateurs. Dans ce cas particulier, nous avons 6 catégories utilisées dans ce projet, y compris l'essentiel, l'éducation, l'épargne, la jouissance, l'investissement, la gentillesse. Tous ces objets sont généralisés par la classe **PartAmount**.

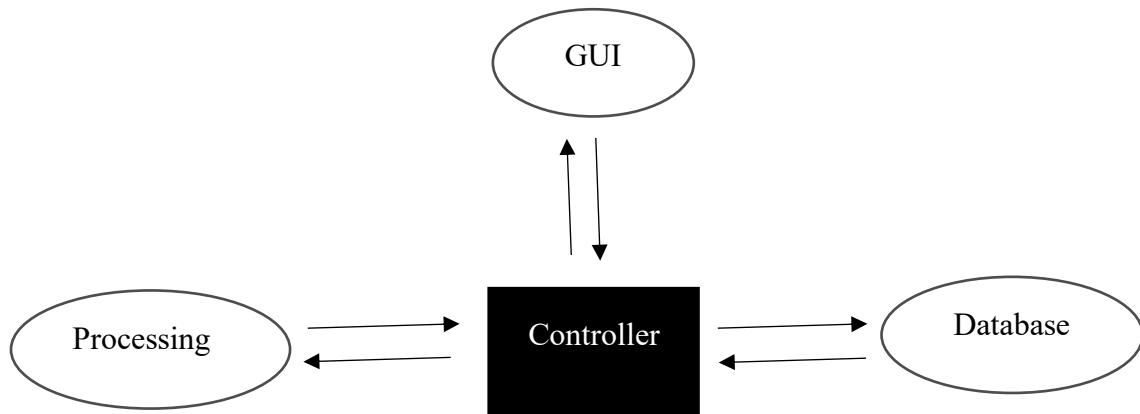
La classe **Asset** représente le composant et l'état de propriété de l'utilisateur existant, y compris les informations sur l'historique des transactions, l'état du solde et les méthodes de transaction du compte.

La classe **User** est une représentation utilisateur.

***** *Modification :***

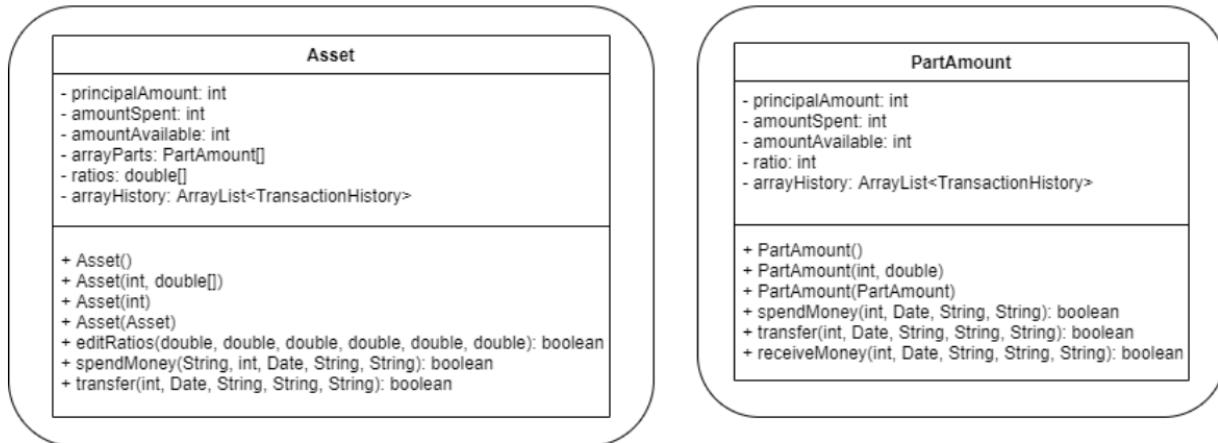
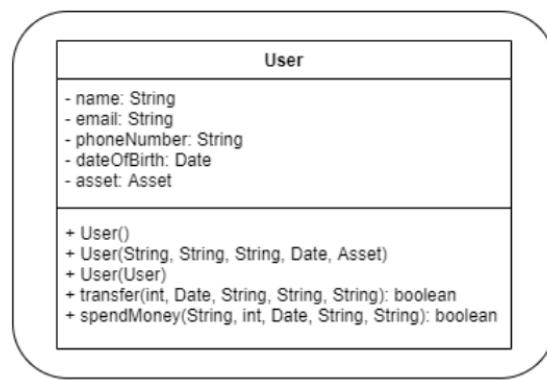
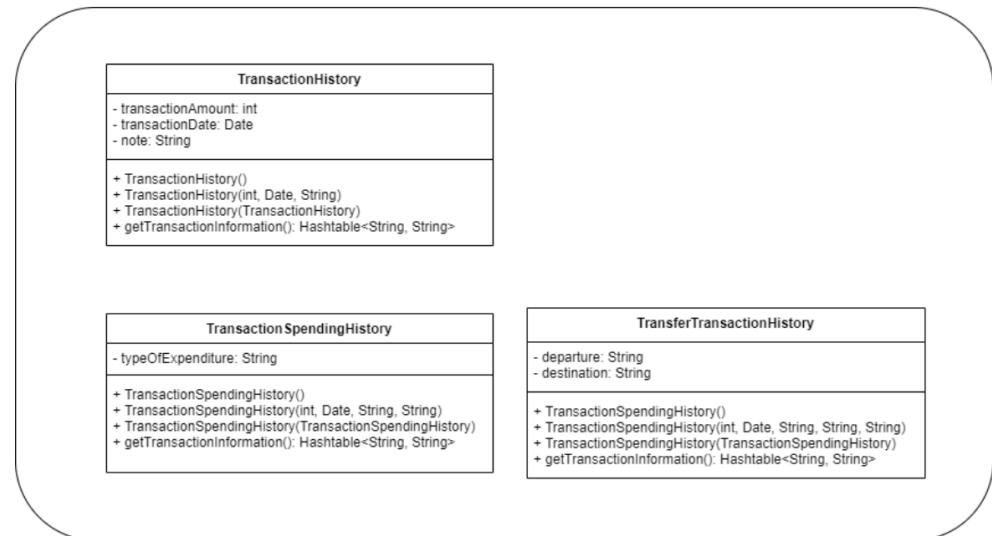
Autre part, nous utilisons la classe **Controller** comme intermédiaire pour se connecter à la base de données et interagir avec l'interface utilisateur. Cette classe recevra des événements du côté de GUI pour appeler les fonctions de gestionnaire correspondantes.

Nous avons le diagramme suivant:

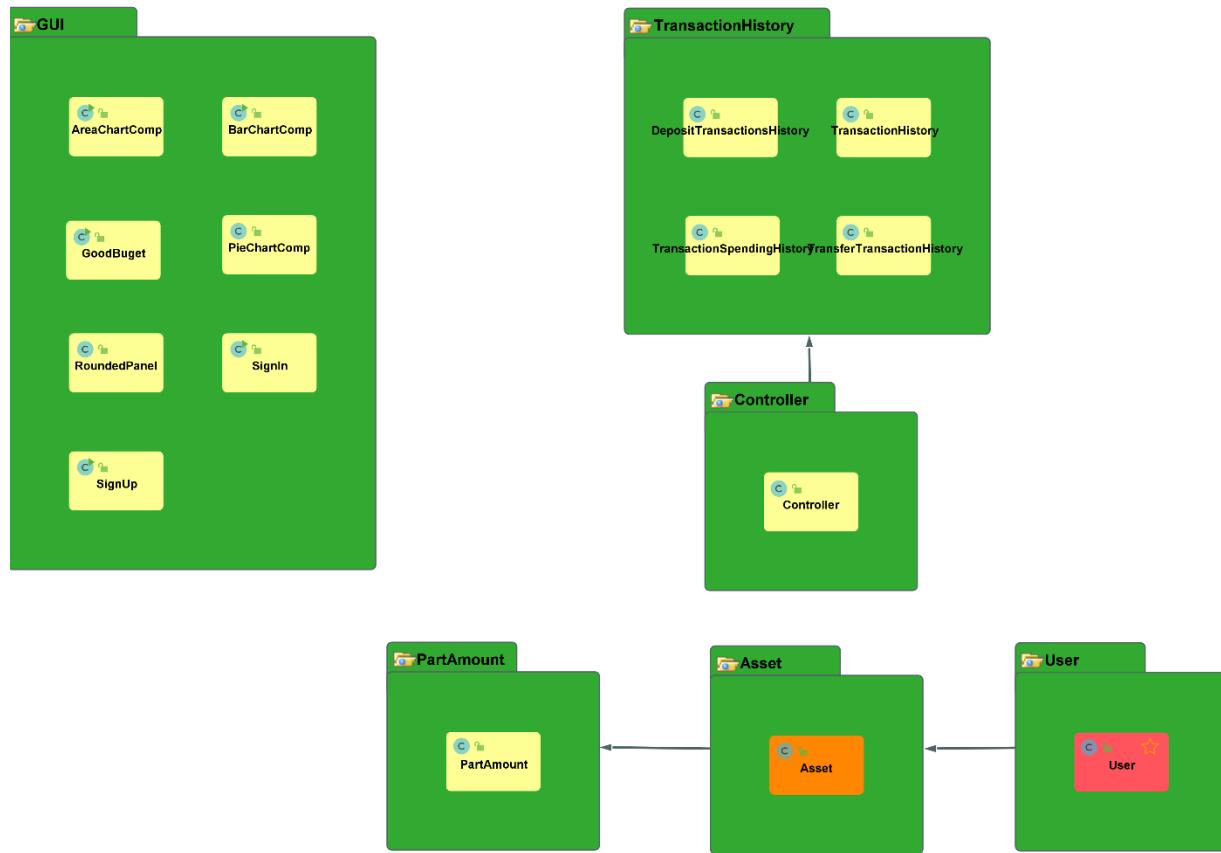


Le diagramme de paquetages pour montrer l'organisation de notre classes.

*** **L'ancien**



*** *Le nouveau*



III. Implémentation

Ci-dessous le prototype de la classe installée dans le projet (réduit) :

```
public class TransactionHistory {  
    private int transactionAmount;  
    private Date transactionDate;  
    private String note;  
  
    TransactionHistory(){...}  
  
    TransactionHistory(int transactionAmount, Date transactionDate, String note){...}  
  
    TransactionHistory(TransactionHistory item){...}  
  
    public Hashtable<String, String> getTransactionInformation() {...}  
}  
  
public class TransactionSpendingHistory extends TransactionHistory{  
    private String typeOfExpenditure;  
  
    TransactionSpendingHistory(){...}  
  
    TransactionSpendingHistory(int transactionAmount, Date transactionDate, String note, String typeOfExpenditure){...}  
  
    TransactionSpendingHistory(TransactionSpendingHistory item){...}  
  
    public Hashtable<String, String> getTransactionInformation() {...}  
}  
  
public class TransferTransactionHistory extends TransactionHistory{  
    String departure;  
    String destination;  
  
    TransferTransactionHistory(){...}  
  
    TransferTransactionHistory(int transactionAmount, Date transactionDate, String note, String departure, String destination)  
    TransferTransactionHistory(TransferTransactionHistory item){...}  
  
    public Hashtable<String, String> getTransactionInformation() {...}  
}
```

```

public class DepositTransactionsHistory extends TransactionHistory{
    String title;

    //constructor
    public DepositTransactionsHistory(String namePartAmount, int transactionAmount, Date transactionDate,
    public DepositTransactionsHistory(DepositTransactionsHistory item){...}

    //luu thong tin lich su
    public Hashtable<String, String> getTransactionInformation() {...}

    public void display(){...}
}

```

```

public class PartAmount {
    private int principalAmount;
    private int amountSpent;
    private int amountAvailable;
    private double ratio;

    ArrayList<TransactionHistory> arrayHistory;

    PartAmount(){}
    PartAmount(int principalAmount, double ratio){}
    PartAmount(PartAmount item){...}

    public boolean spendMoney(int transactionAmount, Date date, String note, String typeOfExpenditure){}
    public boolean transfer(int transactionAmount, Date date, String note, String departure, String destination){}
    public boolean receiveMoney(int transactionAmount, Date date, String note, String departure, String destination){}
}

```

```

public class Asset {
    private int principalAmount;
    private int amountSpent;
    private int amountAvailable;

    PartAmount[] arrayParts;
    double[] ratios;
    ArrayList<TransactionHistory> arrayHistory;

    Asset(){}
    Asset(int principalAmount, double[] ratios){}
    Asset(int principalAmount) {...}
    Asset(Asset item){...}

    public boolean editRatios(double necessary, double education, double saving, double enjoy, double invest, double kindness)
    public boolean spendMoney(String partName, int transactionAmount, Date date, String note, String typeOfExpenditure){}
    public boolean transfer(int transactionAmount, Date date, String note, String departure, String destination){}
}

```

```
public class User {  
    private String name;  
    private String phoneNumber;  
    private String email;  
    private Date dateOfBirth;  
  
    public Asset asset;  
  
    public User(){...}  
  
    public User(String name, String phoneNumber, String email, Date dateOfBirth, Asset asset){...}  
  
    public User(User item){...}  
  
    public boolean transfer(int transactionAmount, Date date, String note, String departure, String destination){...}  
  
    public boolean spendMoney(String partName, int transactionAmount, Date date, String note, String typeOfExpenditure){...}  
}
```

IV. Résultat

Tâche	Avantage	Désavantage	Solution
Décrire le logiciel	Décrivez toutes les fonctionnalités		
Design diagramme de classes	Aucune expérience antérieure		
Base de données de conception		Je n'ai pas appris JDBC	Apprenez dans le futur
Document de plan	Avait de l'expérience en introduction à l'ingénierie logicielle		
Attribution de tâche	Avait de l'expérience en introduction à l'ingénierie logicielle		
Implémenter l'interface de classe	Avait de l'expérience en POO		

V. Plan

Le plan de notre projet (en détaille) jusqu'à la fin du projet

Semaine	Description de la tâche	Durée
4	Diagramme de classe de conception Base de données de conception Décomposition des tâches Créer une interface écrire du rapport 1	1 semaine
5	Construire une base de données	1 semaine
6	Interface graphique de conception pour l'application	1 semaine
7	Créer une interface graphique pour l'application	1 semaine
8	Implémenter pour l'application	2 semaines
9	Vérification et libération des bogues	1 semaine

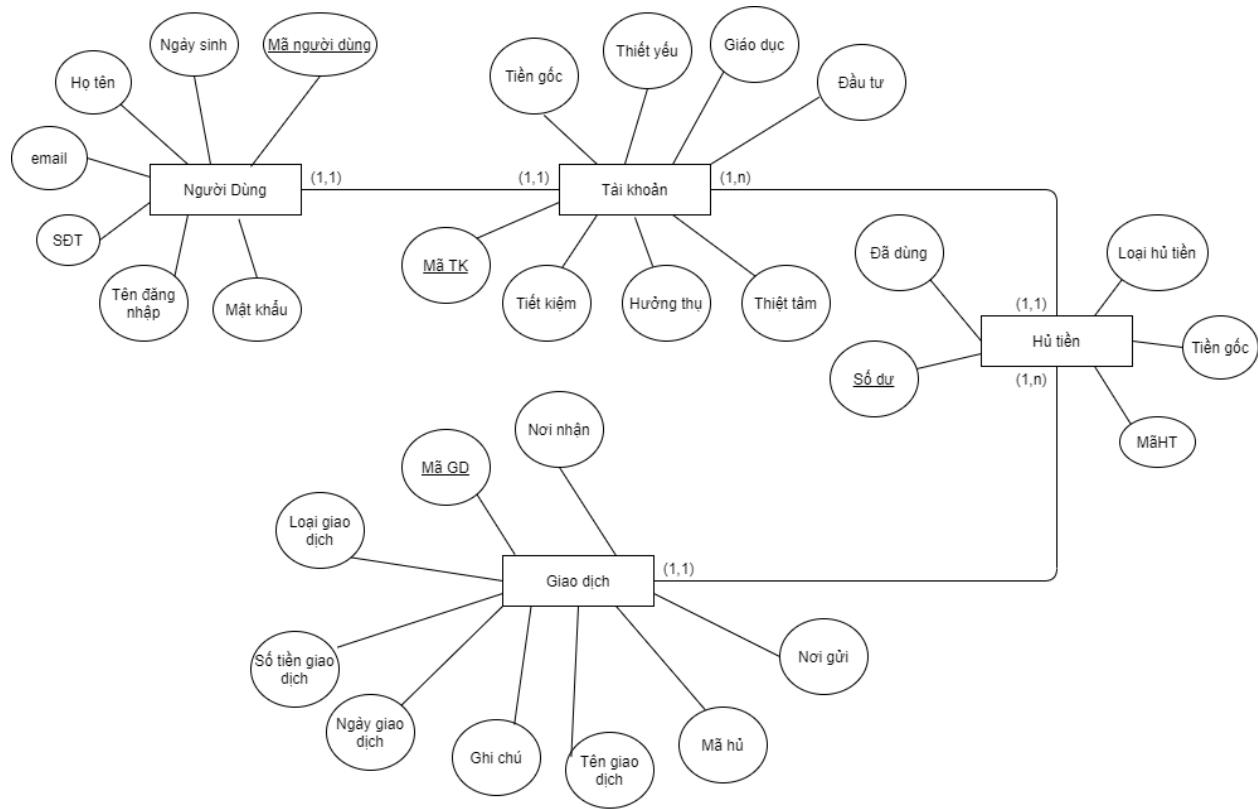
Java IO/ Base de données

I. Introduction

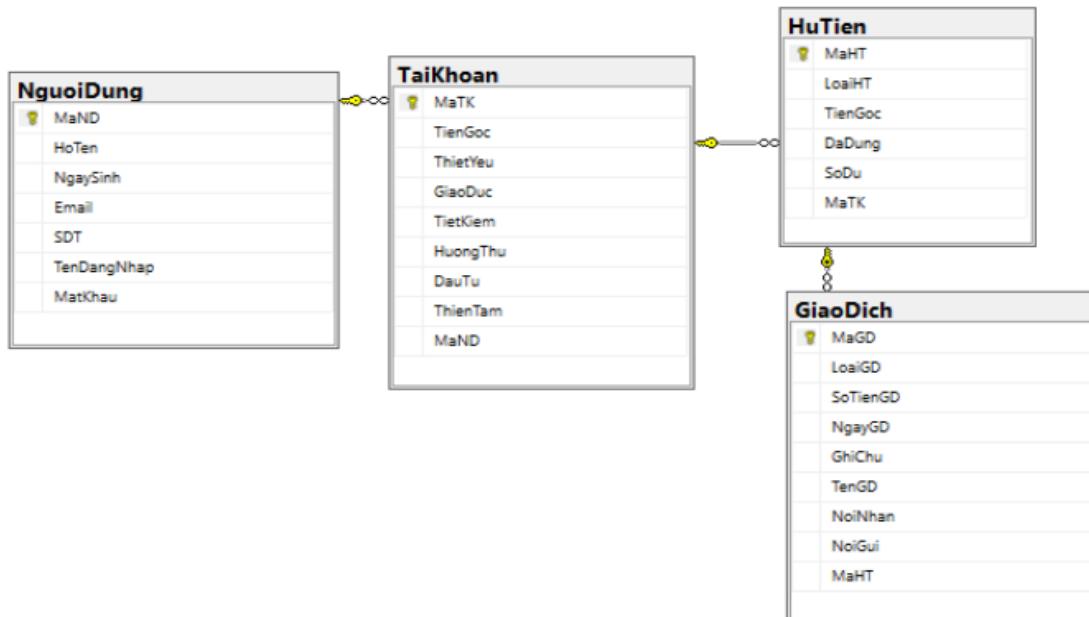
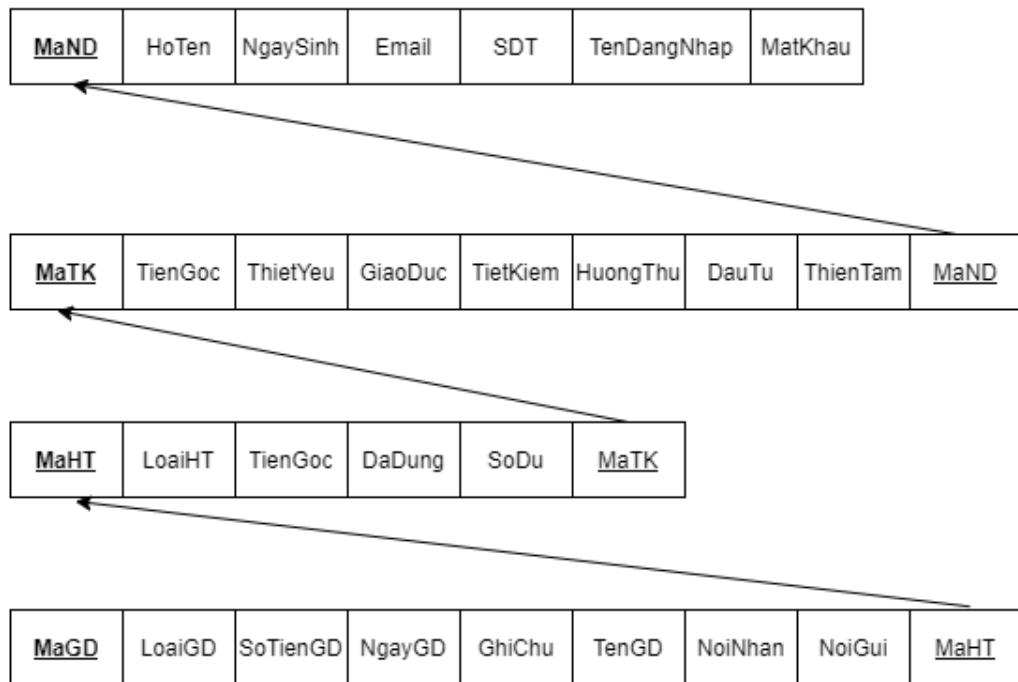
- Le but du rapport est d'expliquer comment les données d'application peuvent être stockées. Le rapport sera utilisé sur le processus de développement de l'application pour faire référence à l'architecture de la base de données et la baser pour implémenter les fonctionnalités de l'application.
- Toutes les données d'application seront stockées dans le système de gestion de base de données MySQL. Parce que l'avantage de la base de données est plus qu'un fichier tel que:
 - + Amélioration de la cohérence des fichiers.
 - + Sécurité renforcée (restriction des accès non autorisés).
 - + Partage de données dans plusieurs applications utilisateur.
 - + Application des contraintes d'intégrité.

II. Analyse et conception

1. Modèle ER



2. Modèle relationnel



- **Table utilisateur:** utiliser pour enregistrer les informations utilisateur, y compris le code utilisateur, le prénom, la date de naissance, l'e-mail, le numéro de téléphone et les informations de connexion telles que le nom d'utilisateur et le mot de passe.
- **Table de compte:** utiliser pour enregistrer les informations du compte, y compris le code de compte, le montant initial du compte et le pourcentage des dépenses (y compris les dépenses essentielles, les études, les économies périodiques, profiter, l'investissement et la charité), code d'utilisateur.
- **Table de HU TIEN:** enregistre des informations sur chaque montant dont dispose l'utilisateur, y compris le code, le type, le montant initial, le montant utilisé, le solde de ce montant et le code de compte.
- **Table de transaction :** enregistre les informations de toutes les transactions de l'utilisateur, y compris: le code de transaction, le type de transaction, le montant de la transaction, la date de la transaction, les notes, le nom de la transaction, la destination, l'expéditeur et le code de HU TIEN.

III. Implémentation

Exemples d'extraits de code qui se connectent à la base de données pour télécharger des informations dans la structure POO dans le projet.

Voici le code qui se connecte à la base de données et exécute la requête.

```
static final String DB_URL = "jdbc:sqlserver://DESKTOP-15VLH02;databaseName=QLSV;integratedSecurity=true";

public boolean logIn(String userName, String password){
    Connection conn = null;

    try{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        conn = DriverManager.getConnection(DB_URL);
        Statement stmt = conn.createStatement();

        //truy van NGUOI DUNG
        String sql = "select MaND from NguoiDung as ND where ND.TenDangNhap = '" + userName + "' and ND.MatKhau";
        ResultSet rs = stmt.executeQuery(sql);

        while(rs.next()){
            this.userId = rs.getInt( columnLabel: "MaND");
        }
    }
}
```

```
//truy van TAI KHOAN
sql = "select MaTK from TaiKhoan as TK where TK.MaND = " + this.userId + ";";
rs = stmt.executeQuery(sql);
while(rs.next()){
    this.assetId = rs.getInt( columnLabel: "MaTK");
}
```

```
//truy van GIAODICH
sql = "select MaGD from GiaoDich as GD where GD.MaHT = " + this.necessaryId + " or GD.MaGD = " + this.educationId
rs = stmt.executeQuery(sql);
while(rs.next()){
    this.historyId.add(rs.getInt( columnLabel: "MaGD"));
}
```

Nous utilisons généralement try catch pour attraper des événements (exception)

```
conn = null;
try {
    conn = DriverManager.getConnection(DB_URL);
    Statement stmt = conn.createStatement();

    stmt.executeUpdate(sqlInsert);
    stmt.executeUpdate(sqlUpdate);

    stmt.close();
} catch (SQLException e) {
    e.printStackTrace();
}

return true;
```

IV. Exemples de données

Exemple de fiche technique dans les tableaux de conception.

Table des utilisateurs

MaND	HoTen	NgaySinh	Email	SDT	TenDangNhap	MatKhau
1	Nguyễn Thị A ...	2001-10-20	a@gmail.com ...	12345	123	123
2	Trần Thị B	2002-10-21	b@gmail.com ...	23456	234	234
3	Đoàn Thị C	2003-10-22	c@gmail.com ...	34567	345	345

Table des comptes

TienGoc	ThietYeu	GiaoDuc	TietKiem	HuongThu	DauTu	ThienTam	MaND
1000	0,55	0,1	0,1	0,1	0,1	0,05	1
5000	0,55	0,1	0,1	0,1	0,1	0,05	2
6000	0,55	0,1	0,1	0,1	0,1	0,05	3

Table des HU TIEN

MaHT	LoaiHT	TienGoc	DaDung	SoDu	MaTK
1	necessary	550	50	500	1
2	education	100	0	100	1
3	saving	100	0	100	1
4	enjoy	100	0	100	1
5	invest	100	0	100	1
6	kindness	50	0	50	1
7	necessary	2750	750	2000	2
8	education	500	0	500	2
9	saving	500	0	500	2
10	enjoy	500	0	500	2
11	invest	500	0	500	2
12	kindness	250	0	250	2
13	necessary	3300	300	3000	3
14	education	600	0	600	3
15	saving	600	0	600	3
16	enjoy	600	0	600	3
17	invest	600	0	600	3
18	kindness	300	0	300	3

Table des transactions

MaGD	LoaiGD	SoTienGD	NgayGD	GhiChu	TenGD	NoiNhan	NoiGui	MaHT
1	GDCT	50	2021-04-02	Ăn trong nhà h...	Ăn uống	NULL	NULL	1
2	GDCT	50	2021-04-02	NULL	Gửi xe	NULL	NULL	7
3	GDCH	700	2021-04-02	NULL	NULL	education	necessary	7
4	GDCH	700	2021-04-02	NULL	NULL	education	necessary	8
5	GDCT	300	2021-04-02	Vật dụng cần t...	Mua sắm	NULL	NULL	13

V. Résultat

• Compléter:

- Architecture de la base de données et instance de la base de données.
- Fonction de création de base de données, fonction de création de colonne suivie de l'architecture de base de données.
- L'application peut appeler des fonctions de base à partir du SQLServer.

• Avantages:

- A de l'expérience en base de données relationnelle, langage SQL au semestre précédent.
- A de l'expérience dans le processus de conception de bases de données relationnelles.

• Désavantages:

- Passez trop de temps à corriger les bogues de syntaxe lors de l'utilisation de SQL.

• Solutions envisagées:

- Revoir la syntaxe du langage SQL

VI. Plan

Semaine	Description de la tâche	Duration
9	Compléter les fonctions de l'application: + Fonction de connexion + Fonctions de gestion financière + Fonctions de transaction + Fonctions de notification utilisateur Compléter l'interface utilisateur	2 semaines
10	Vérification et libération des bogues	1 semaine
11		

Java GUI

I. Introduction

Ce rapport décrit l'interface d'une application bureautique: l'application de gestion financière. Nous utilisons l'outil figma pour concevoir l'interface de l'application.

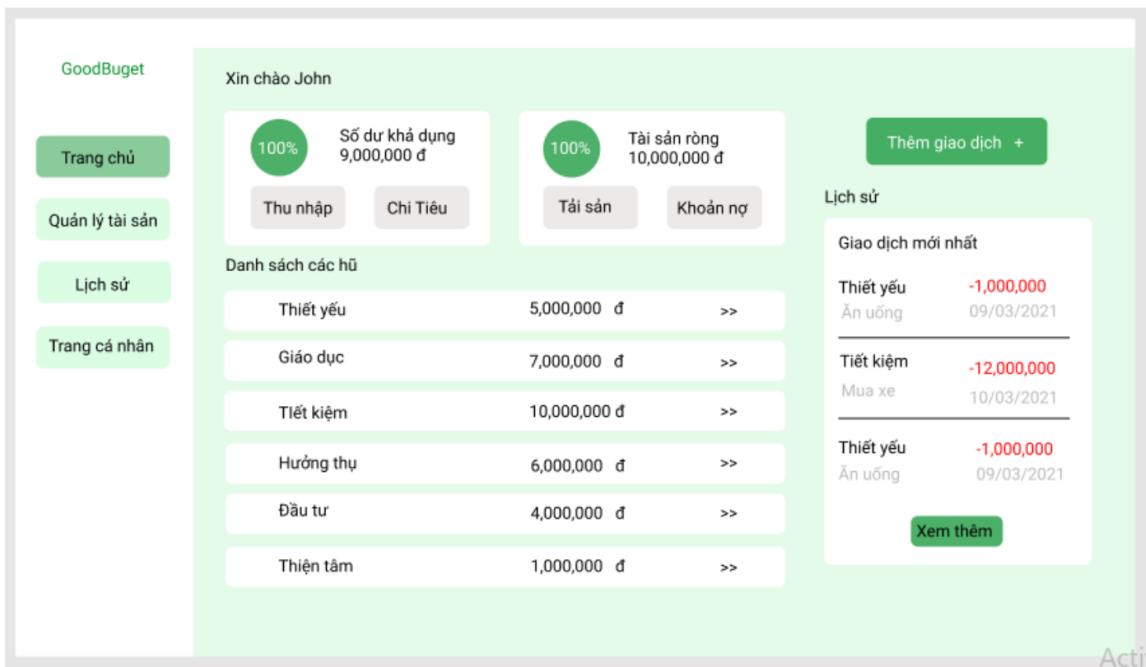
Tout d'abord, mon équipe discutera de la fonctionnalité clé de l'application et se basera sur celle-ci pour rédiger l'interface utilisateur graphique. Ensuite, nous classons les groupes fonctionnels et scriptons l'application. À partir de là, nous avons pu utiliser cette base pour concevoir des interfaces d'écrans fonctionnels à l'aide de l'outil figma. Après cela, notre examen complet garantit la cohérence et ne provoque pas d'erreurs d'interface utilisateur. Quand le brouillon de l'interface graphique terminé, mon équipe utiliser NetBean IDE pour concevoir à nouveau l'interface graphique.

Notre application est une application simple. Mais il s'agit d'une application liée à la finance, les fonctionnalités suggérées dans l'application sont assez nouvelles, ce qui peut entraîner des difficultés d'utilisation chez l'utilisateur.

A partir de ces caractéristiques, nous exigeons que l'interface de toutes les fonctionnalités soit facile à comprendre, facile à utiliser et que les utilisateurs ne prennent pas trop de temps pour s'habituer à l'application.

II. Analyse et conception

- **Interface de la page d'accueil**, inclure :
 - Informations générales sur les informations de compte.
 - Informations sur les ingrédients divisés, comme essentiel, éducatif, économique, plaisir, généreux et charitable.
 - Informations sur l'historique des transactions.
 - Si vous cliquez sur le bouton *Ajouter une transaction* (Thêm giao dịch), les groupes d'écrans fonctionnels suivants seront menés.



- Si vous cliquez sur le bouton *Ajouter une transaction* (Thêm giao dịch), les groupes d'écrans fonctionnels suivants seront menés.

- Interface pour ajouter de l'argent à votre compte

L'une des premières options est une interface pour ajouter de l'argent à votre compte

Thêm giao dịch

THU NHẬP CHI TIÊU CHUYỂN TIỀN

0 đ

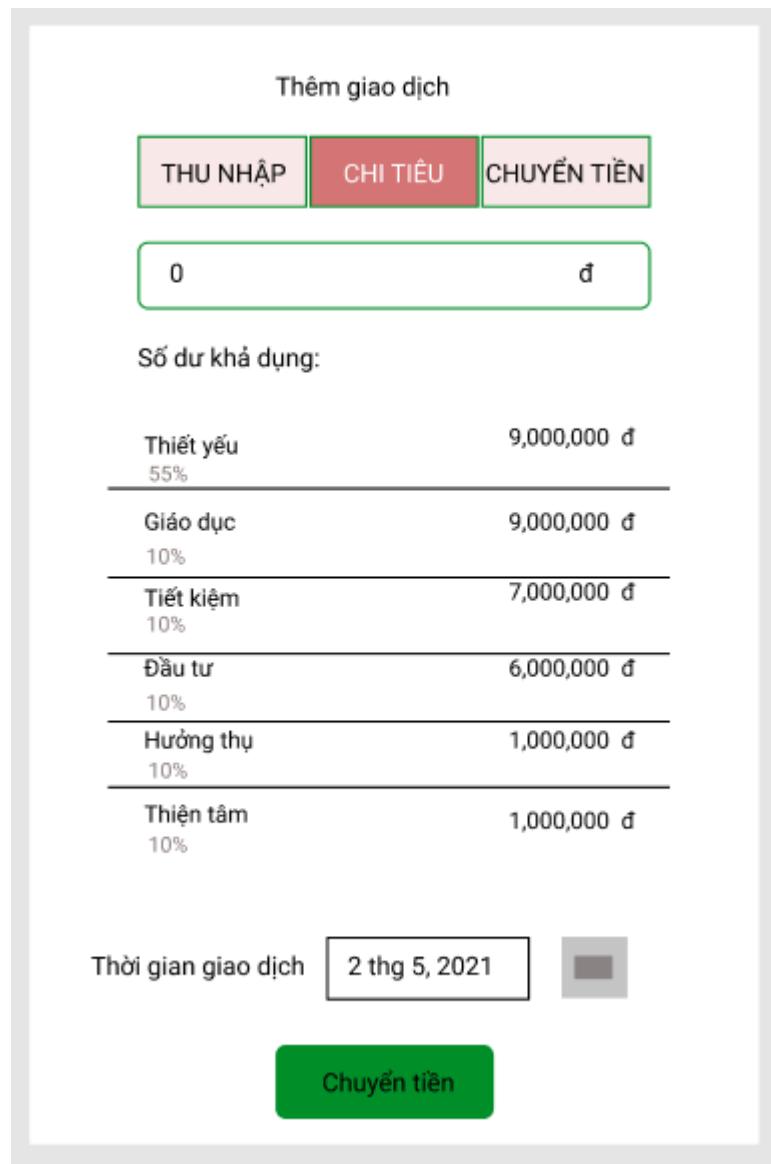
Tỷ lệ phân bổ thu nhập: đ %

Thiết yếu 55%	0 đ
Giáo dục 10%	0 đ
Tiết kiệm 10%	0 đ
Đầu tư 10%	0 đ
Hưởng thụ 10%	0 đ
Thiện tâm 10%	0 đ

Thời gian giao dịch 2 thg 5, 2021

Chuyển tiền

- Le suivant est la fonction de dépense



- L'interface décrit la **fondction de transfert** pas dans l'application



- **Interface de l'écran de l'historique des transactions**

L'interface de la fonction aide les utilisateurs à afficher l'historique détaillé des transactions.

The screenshot shows the GoodBuget application interface. On the left, there is a sidebar with the following menu items:

- Trang chủ
- Quản lý tài sản
- Lịch sử** (highlighted in green)
- Trang cá nhân

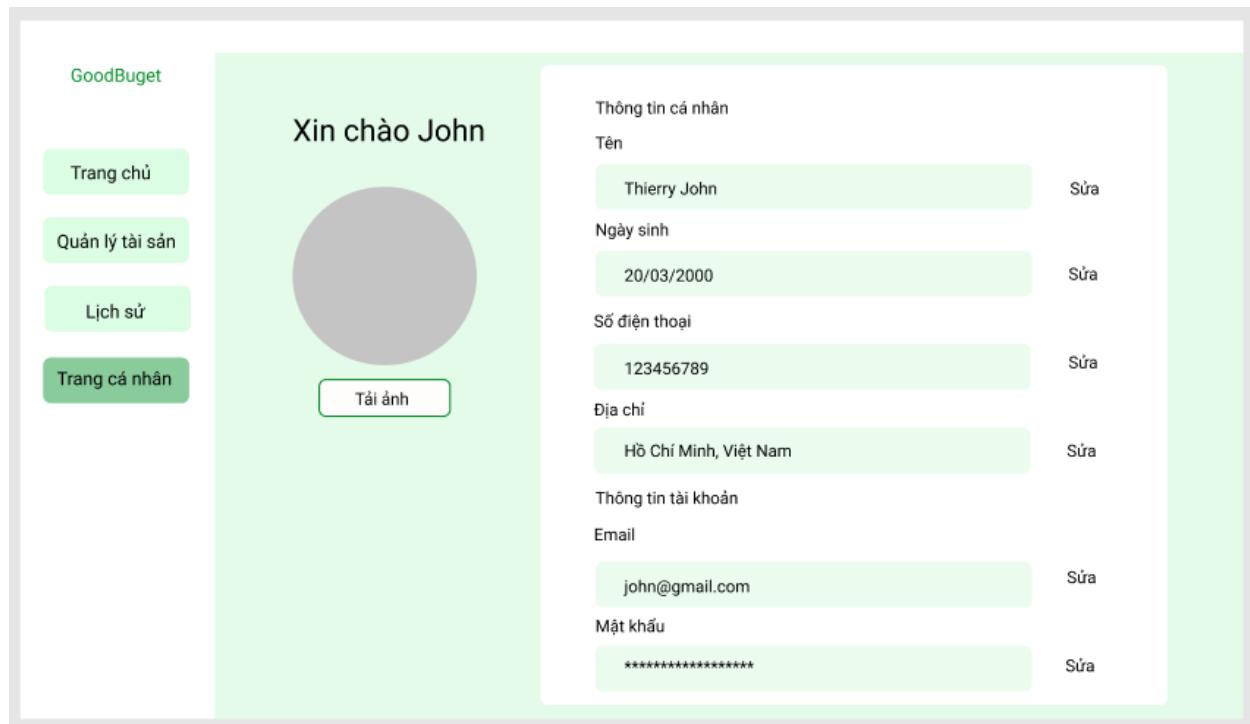
The main content area is titled "Lịch sử" (History) and contains a table of recent transactions. The table has columns for category, description, and amount. A "Chọn hú" (Select category) button is located above the table.

		Giao dịch mới nhất
Thiết yếu	-1,000,000	Ăn uống 09/03/2021
Tiết kiệm	-12,000,000	Mua xe 10/03/2021
Thiết yếu	-1,000,000	Ăn uống 09/03/2021

Below the table, there is a "Chọn ngày:" (Select date:) input field set to "2 tháng 5, 2021" and a "Xem thêm" (View more) button.

- **Interface d'écran de profil utilisateur.**

Fonction pour afficher les informations de l'utilisateur, à travers laquelle l'utilisateur peut modifier les champs d'informations.



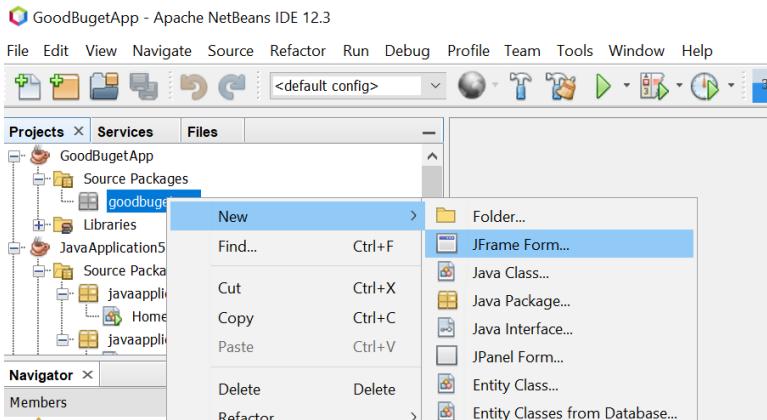
III. Implémentation

- Lorsque la conception de l'interface graphique (GUI) est terminée, notre équipe utilise l'EDI NetBeans pour créer la GUI pour les programmes Java.

Avec l'EDI NetBeans, nous créons facilement la GUI, ajoutons des contrôles par glisser-déposer, ajoutons des gestionnaires d'événements.

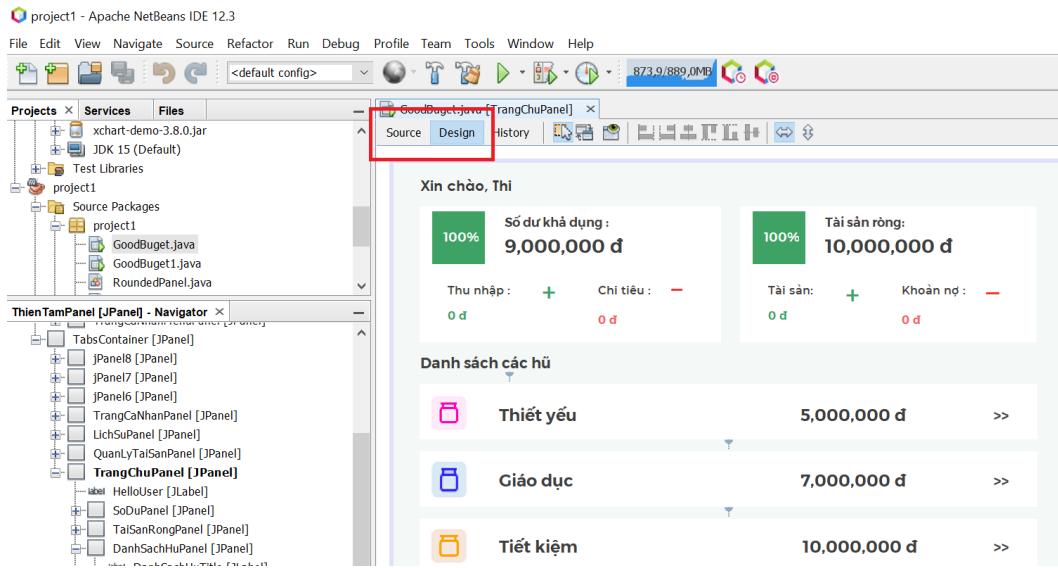
- Après avoir créé le projet d'application, nous devons créer un conteneur Java dans lequel nous placerons les autres composants GUI requis. Nous créons un conteneur en utilisant le composant JFrame.

Pour ajouter un conteneur JFrame:



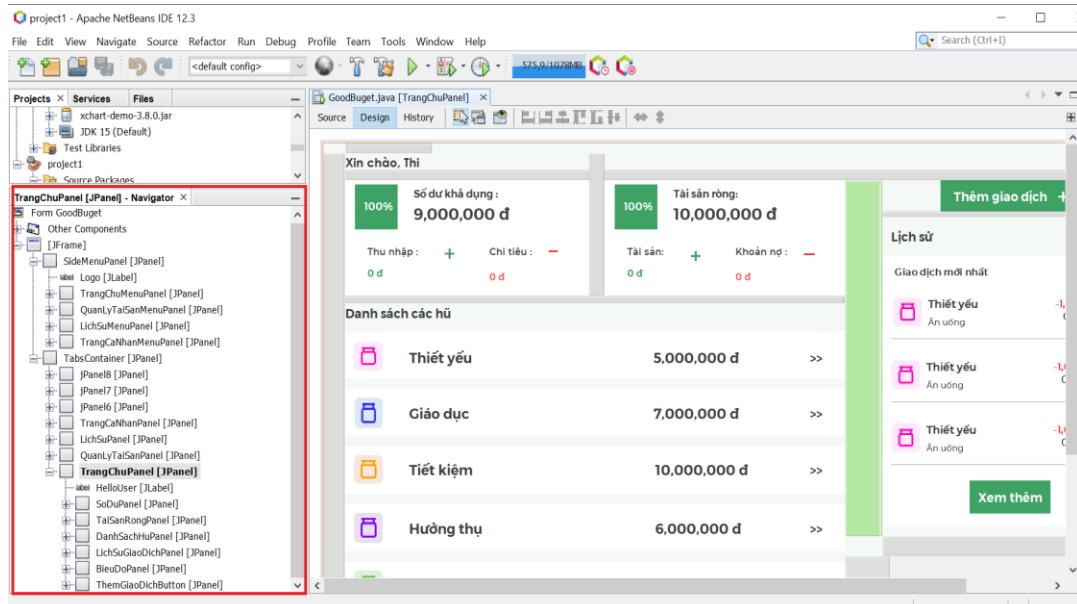
- Une fois le processus de création JFrame terminé, vous pouvez voir deux onglets utilisés pour concevoir la GUI et le code source.

Si vous cliquez sur le bouton Source, l'EDI affiche le code source Java de l'application qui est automatiquement généré par GUI Builder.



Avec l'EDI, vous pouvez également personnaliser et ajouter des événements dans le code source.

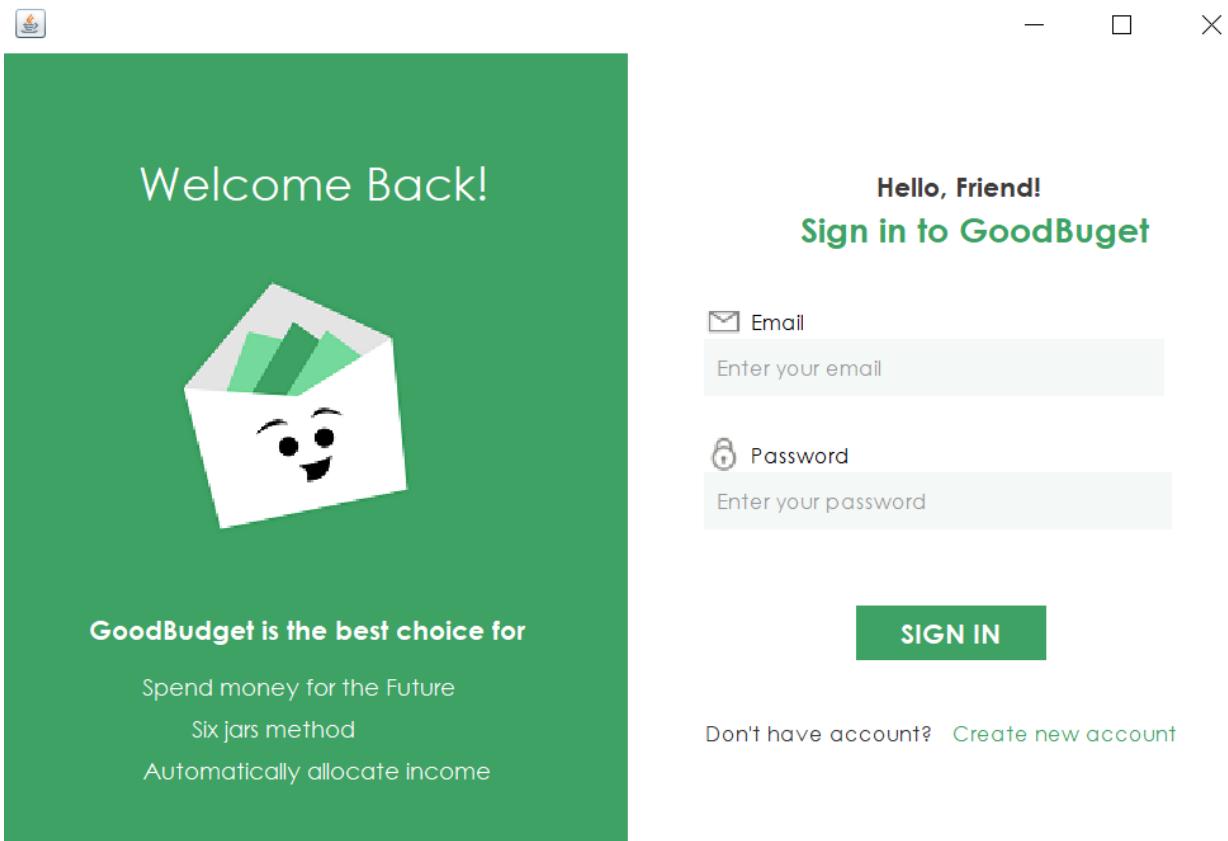
- De plus, les composants de la GUI sont organisés sous forme d'arbres. Cela facilite la manipulation des composants.



IV. Démonstration

- Tout d'abord, les utilisateurs doivent créer un compte pour pouvoir utiliser l'application.

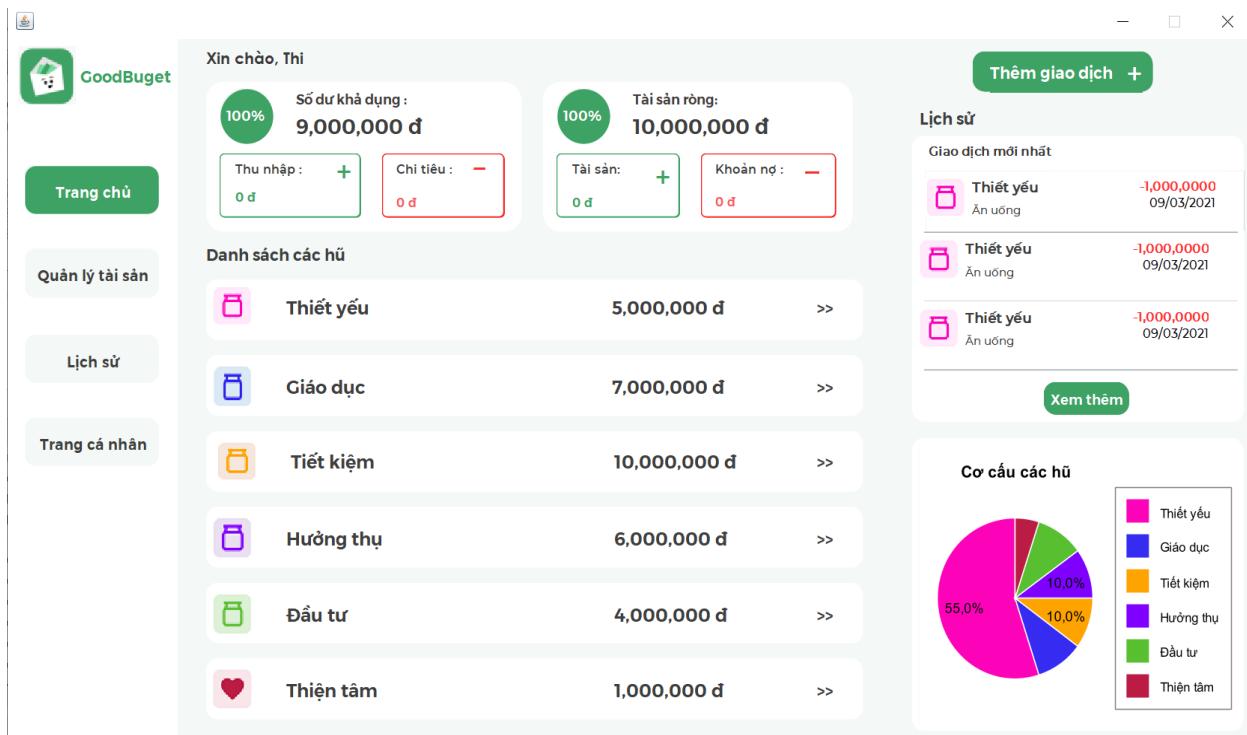
Si vous avez déjà un compte, les utilisateurs peuvent se connecter en utilisant le formulaire ci-dessous.



- Nous concevons en outre des interfaces pour afficher des informations détaillées sur la composition de la propriété divisée. Voici un exemple de l'ingrédient essentiel.

Le solde restant pour les ingrédients essentiels est de 9 millions de VND.

Ci-dessous, c'est la zone qui affiche les informations d'historique des transactions.

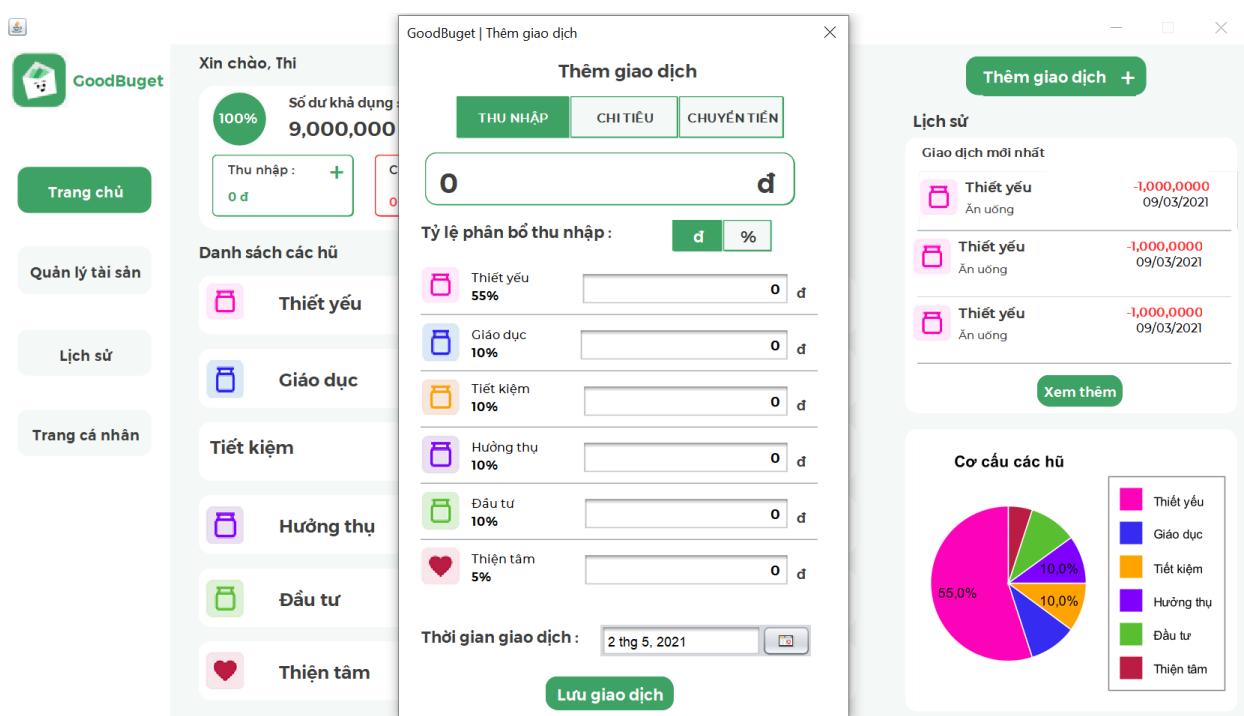


C'est l'interface pour recharger votre compte.

Puis, les utilisateurs peuvent diviser le montant proportionnellement ou à volonté (VND) en fonction du montant total ajouté.

L'utilisateur peut choisir une heure de transaction ou l'ignorer, auquel cas l'application tentera d'obtenir l'heure du système

Cliquez sur Enregistrer pour terminer la transaction



Vous trouverez ci-dessous une interface vers la fonctionnalité de transaction.

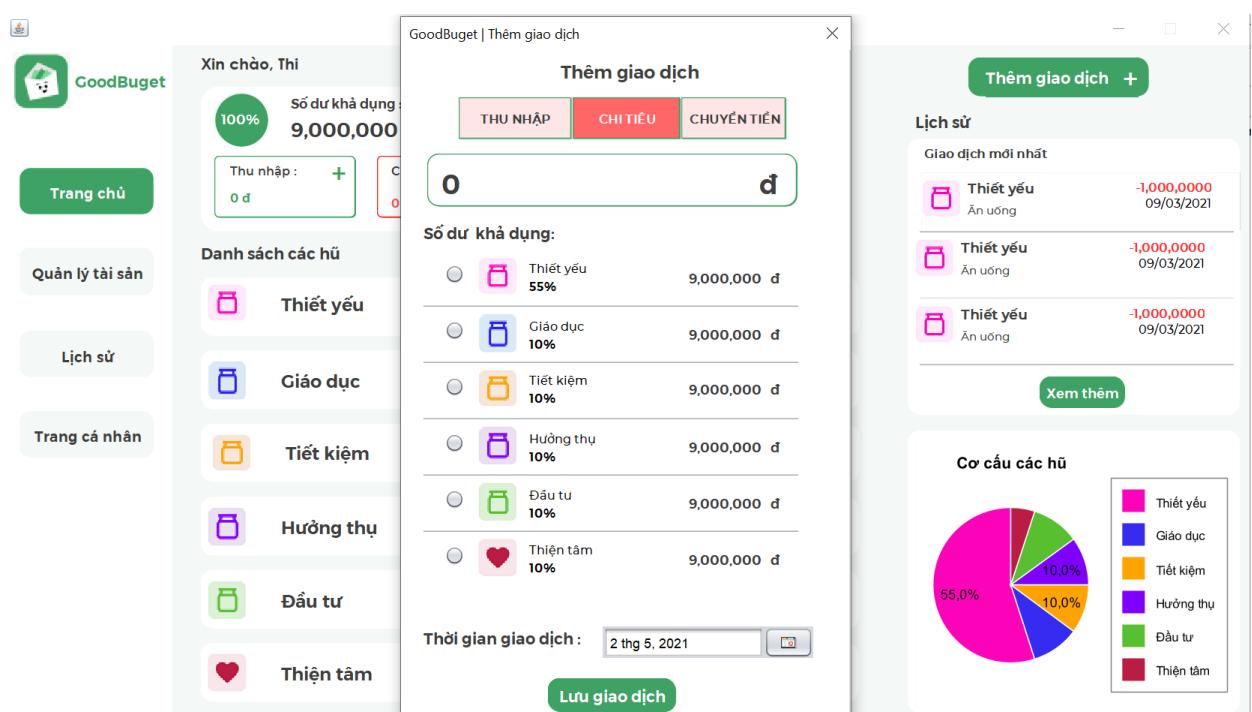
La première consiste à saisir le montant à dépenser.

En suite, les utilisateurs doivent choisir la composante de dépenses appropriée.

Si le solde correspondant est satisfait du montant de la transaction de dépenses, cette transaction est valide et vice versa.

L'utilisateur peut choisir une heure de transaction ou l'ignorer, auquel cas l'application tentera d'obtenir l'heure du système

Cliquez sur Enregistrer pour terminer la transaction.

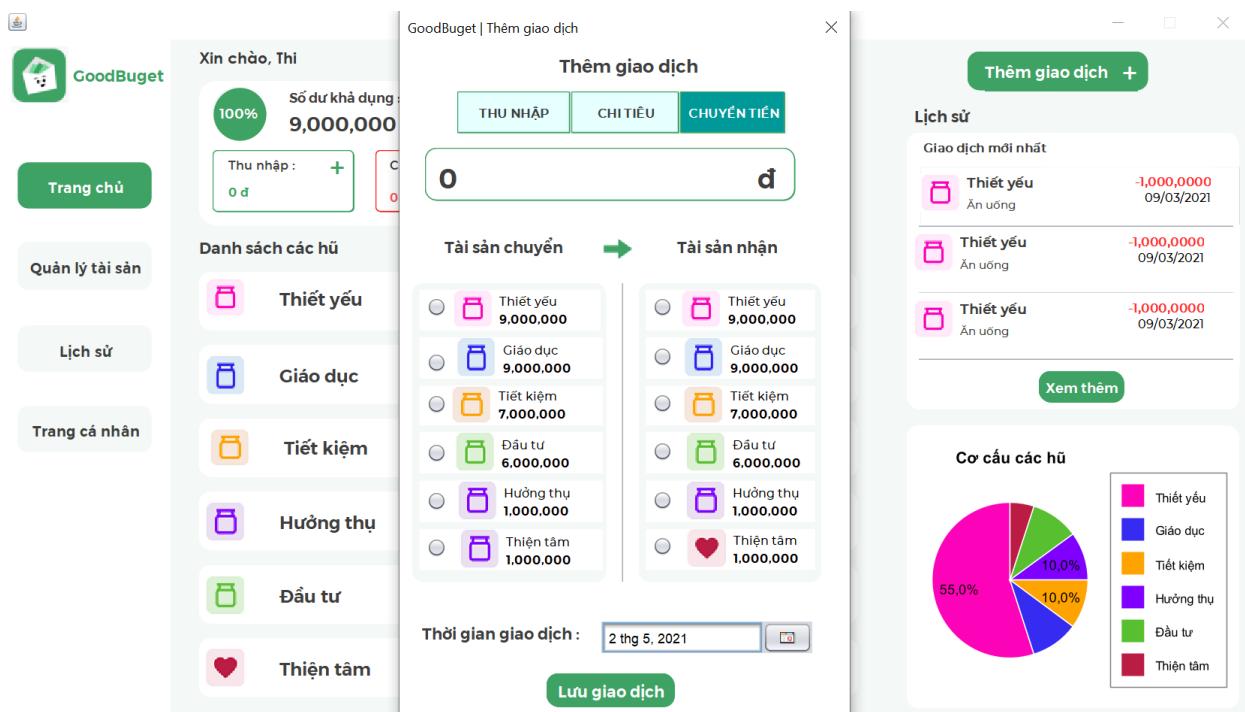


Ci-dessous la transaction de transfert.

l'utilisateur entre le montant à transférer. Après cela, Choisissez ensuite la destination d'envoi et de réception souhaitée.

L'utilisateur peut choisir une heure de transaction ou l'ignorer, auquel cas l'application tentera d'obtenir l'heure du système

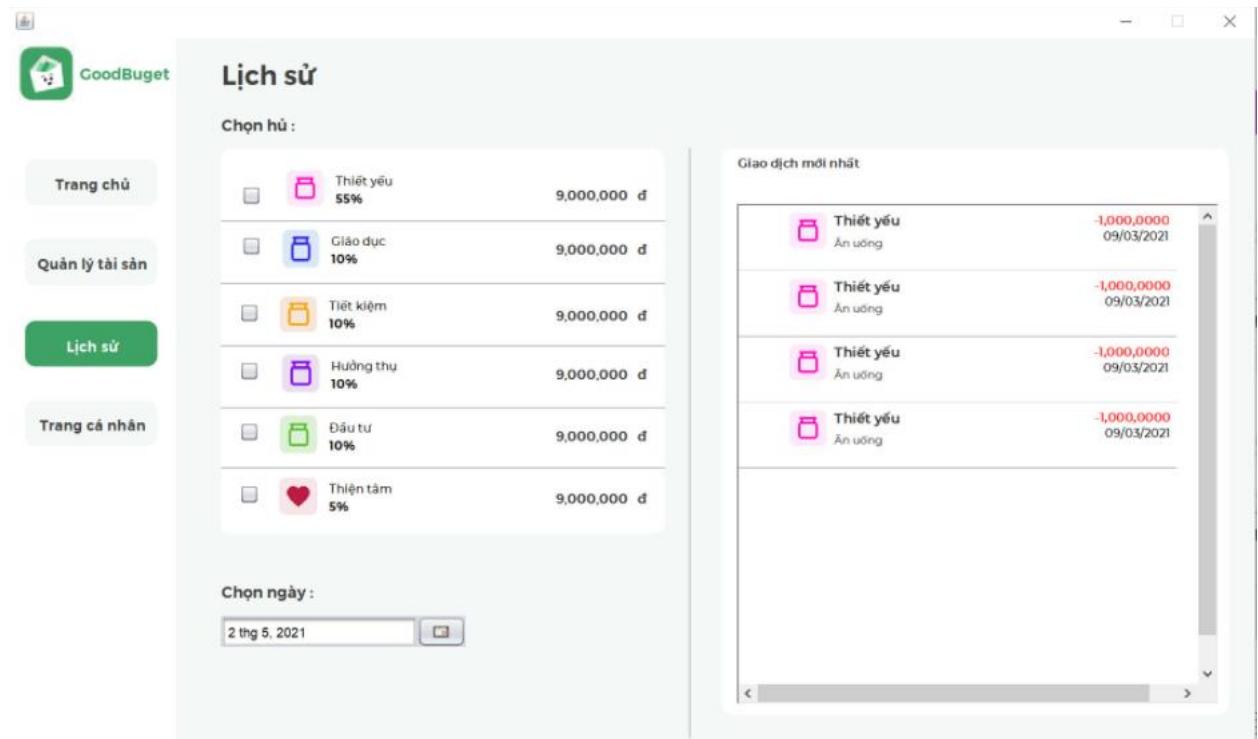
Cliquez sur Enregistrer pour terminer la transaction.



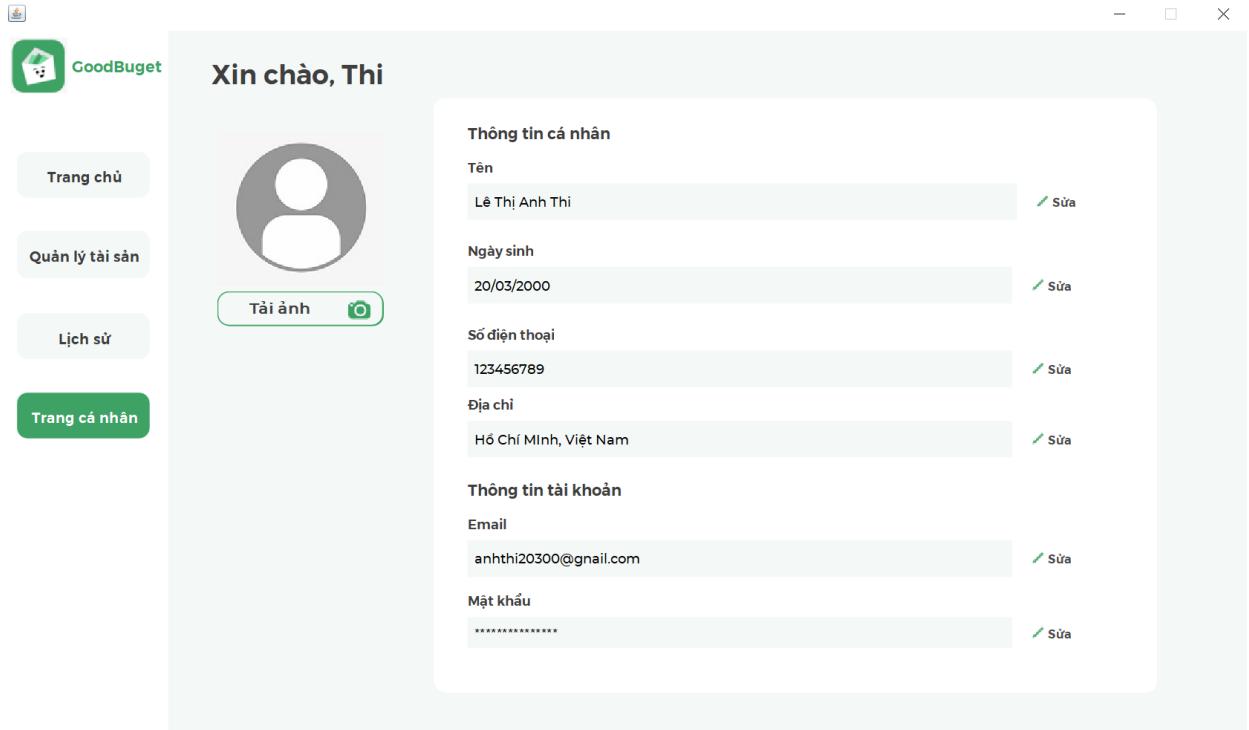
Vous trouverez ci-dessous la fonction permettant aux utilisateurs de rechercher des informations détaillées sur l'historique des transactions.

Vous pouvez visualiser sur la paroi du composant ou dans son ensemble en cliquant sur l'ingrédient respectif.

L'utilisateur peut rechercher des transactions par date.



Vous trouverez ci-dessous la fonction d'affichage des informations utilisateur et d'édition.



IV. Résultat

- Archivé:
 - + Le rapport pour l'interface utilisateur graphique.
 - + L'implémentation de l'interface graphique en Java.
 - + Le plan pour la prochaine fois pour terminer le projet.
- Avantages:
 - + NetBean IDE aide mon équipe à concevoir et à implémenter rapidement des gestionnaires d'événements.
 - + A des expériences sur l'interface graphique dans le développement d'applications Windows Form
- Inconvénient :
 - + difficulté à combiner le back-end avec l'interface
=> *solutions* : parlez à l'instructeur pour la partie pratique du sujet.

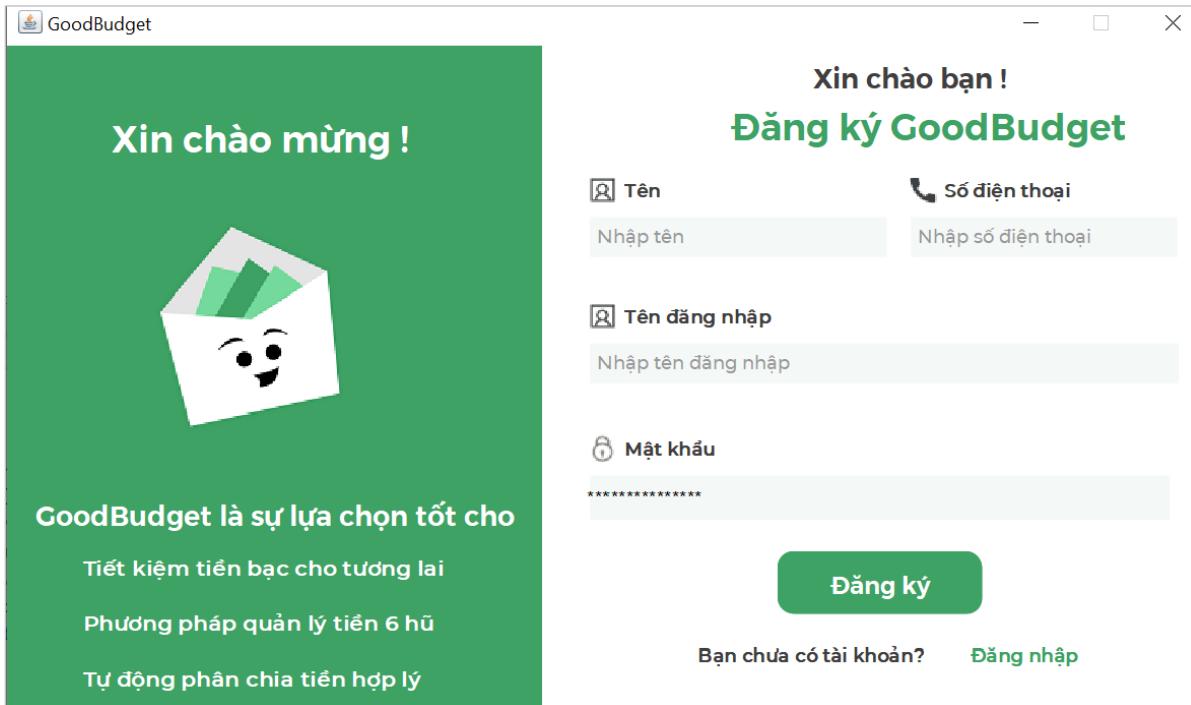
V. Plan

Durée	Tâchs
1 semaine	Fusion front-end et back-end pour obtenir le produit fini
	- Correction et gestion des exceptions - Test de fonction

Démonstration

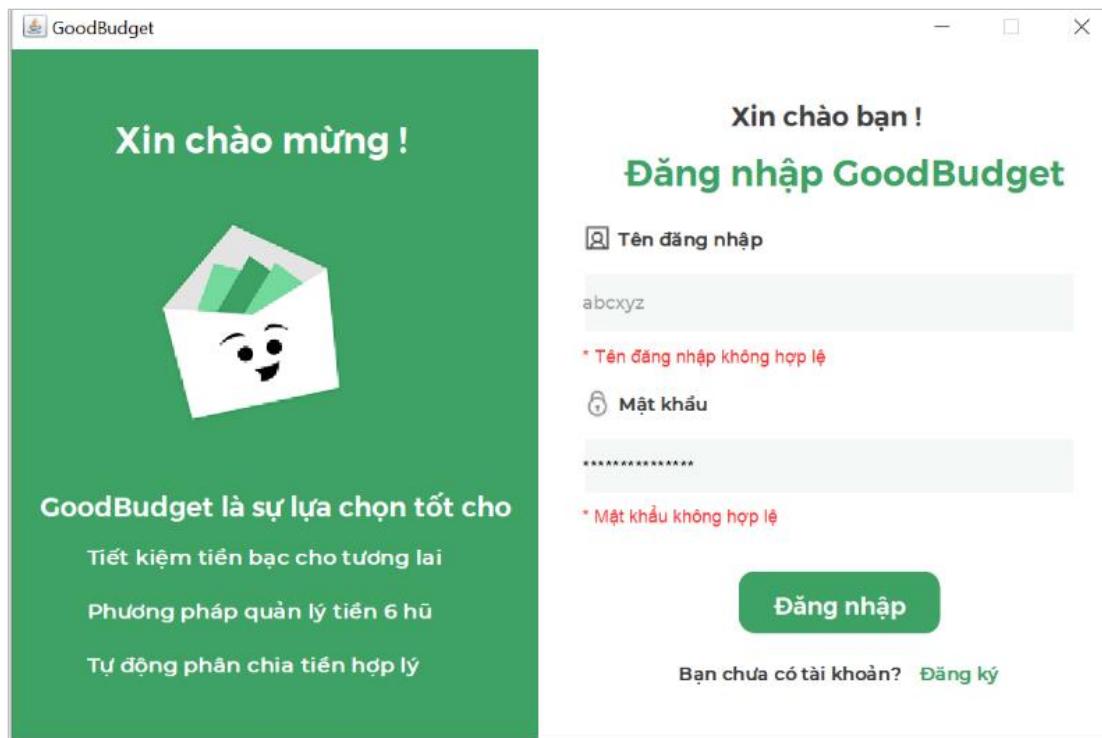
I. Interface de l'écran d'enregistrement du compte utilisateur.

- Tout d'abord, les utilisateurs doivent créer un compte pour pouvoir utiliser l'application.



- Ici, l'utilisateur saisira respectivement les informations personnelles (nom d'utilisateur, numéro de téléphone) et les informations de compte utilisateur (nom d'utilisateur, mot de passe).
- Si vous avez déjà un compte, les utilisateurs peuvent se connecter en utilisant le formulaire cidessous.

II. Interface de l'écran de connexion au compte utilisateur.

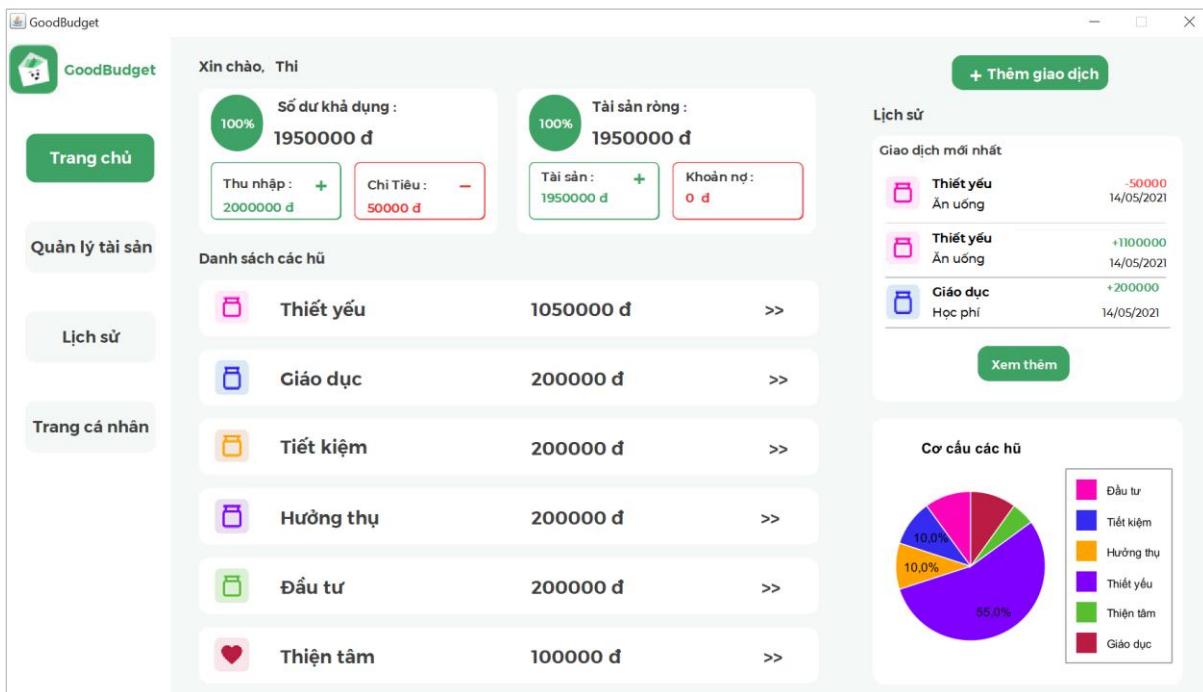


Ici, l'utilisateur entre le nom d'utilisateur et le mot de passe pour se connecter à l'application.

La validité des champs de saisie est vérifiée. Si le nom d'utilisateur ou le mot de passe est incorrect, il en informera l'utilisateur comme sur l'écran ci-dessous :

III. Interface de l'écran d'accueil

- Interface de la page d'accueil, inclure :
 - o Informations générales sur les informations de compte.
 - o Informations sur les ingrédients divisés, comme essentiel, éducatif, économie, plaisir, généreux et charitable.
 - o Informations sur l'historique des transactions.
 - o Informations sur la structure des proportions des ingrédients divisés.



1. Informations générales sur les informations de compte :



Voici les informations de compte de l'utilisateur, il existe deux types d'informations :

- * Soldes disponibles : le solde disponible de l'utilisateur est égal au revenu total moins les dépenses totales.
- * Valeur nette des actifs: la valeur nette d'un utilisateur correspond au total de l'actif moins le passif.

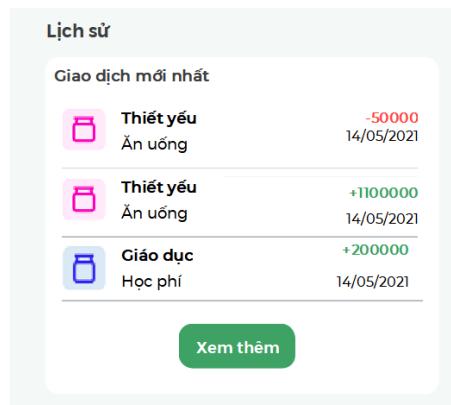
2. Informations sur les ingrédients divisés :

Danh sách các hủ			
	Thiết yếu	1050000 đ	>>
	Giáo dục	200000 đ	>>
	Tiết kiệm	200000 đ	>>
	Hưởng thụ	200000 đ	>>
	Đầu tư	200000 đ	>>
	Thiện tâm	100000 đ	>>

- Les finances de l'utilisateur sont divisées en ingrédients, comme essentiel, éducatif, économe, plaisir, généreux et charitable.
- Pour chaque type d'ingrédient, il y a un pourcentage correspondant.

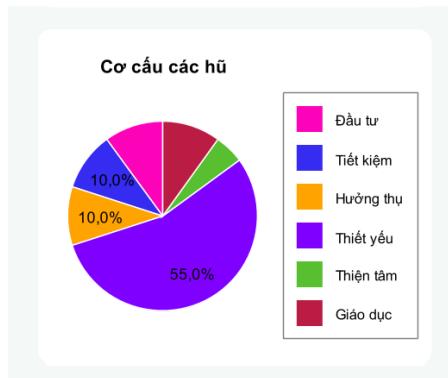
3. Informations sur l'historique des transactions

- Ici, afficher le dernier historique des transactions de l'utilisateur. Pour chaque transaction, on obtient les informations de transaction dans lesquelles « pot », le nom de la transaction spécifique, le montant de la transaction, la date de la transaction.
- Pour les voir tous, cliquez sur le bouton « Voir plus »

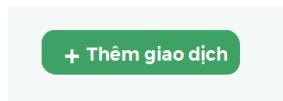


4. Informations sur la structure des proportions des ingrédients divisés.

- Fournir des informations structurelles sur les ingrédients divisés à partir des finances de l'utilisateur (pourcentage de 6 pots).
- Le pourcentage de chaque ingrédient est prédefini dans l'application selon la «méthode des 6 pots» (55,10,10,10,10,5). Cependant, l'utilisateur peut personnaliser ces pourcentages.



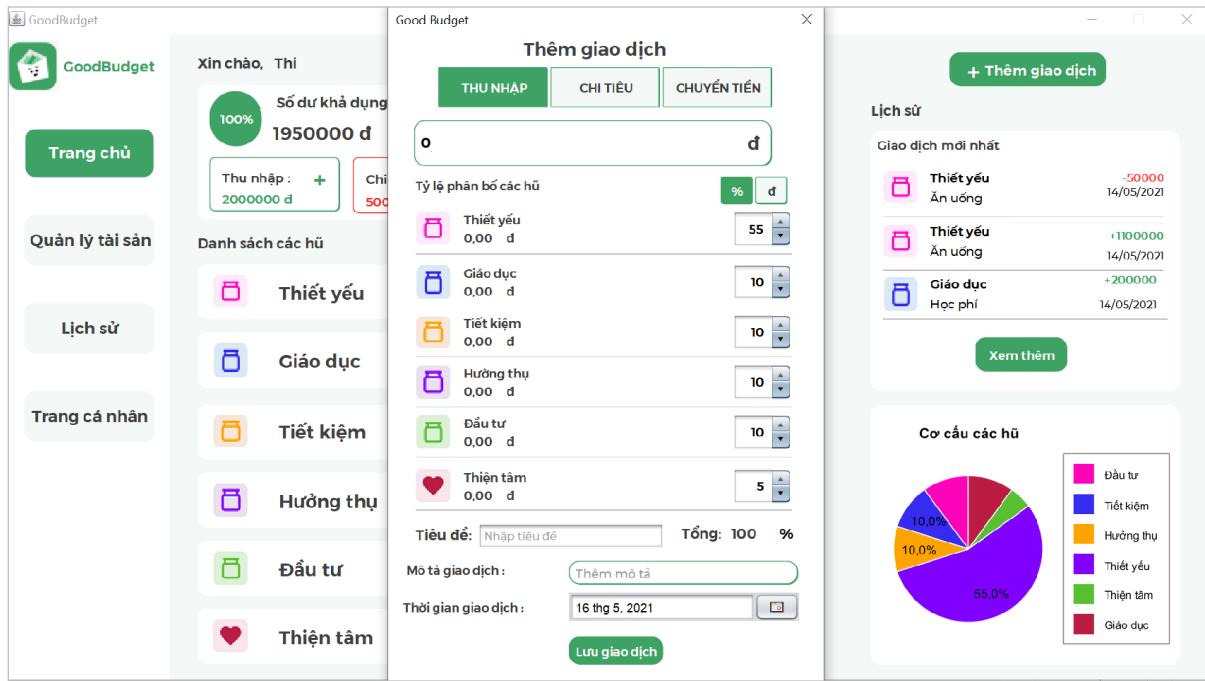
*** En particulier, en cliquant sur le bouton Ajouter une transaction, on exécute les principales fonctionalités de l'application, telles que l'ajout de transactions de revenus, l'ajout de revenus de dépenses, l'ajout de transactions de transfert d'argent.



Et on obtient les écrans d'interface ci-dessous.

II. Interface de l'écran des fonctionnalités principales

1. Interface l'écran d'exécution la fonctionnalité d'ajout de transactions de revenus.



- Tout d'abord, l'utilisateur entre le montant de la transaction dans la zone de saisie.

- L'application affichera le pourcentage par défaut des "pots". Ces pourcentages peuvent être personnalisés par l'utilisateur.



- Les utilisateurs remplissent les informations nécessaires puis cliquent sur le bouton «Enregistrer la transaction».

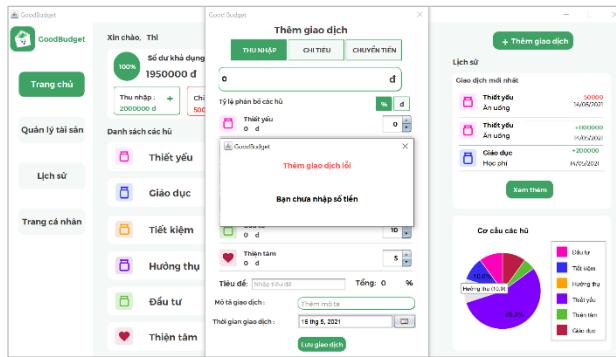
Tiêu đề: Nhập tiêu đề

Mô tả giao dịch: Thêm mô tả

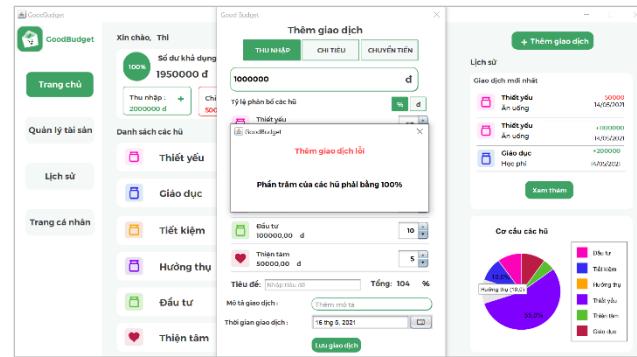
Thời gian giao dịch: 16/05/2021

Lưu giao dịch

- S'il y a une erreur comme si l'utilisateur n'a pas entré le montant ou si le pourcentage total des "pots" est supérieur à 100%, cela donnera un message d'erreur.

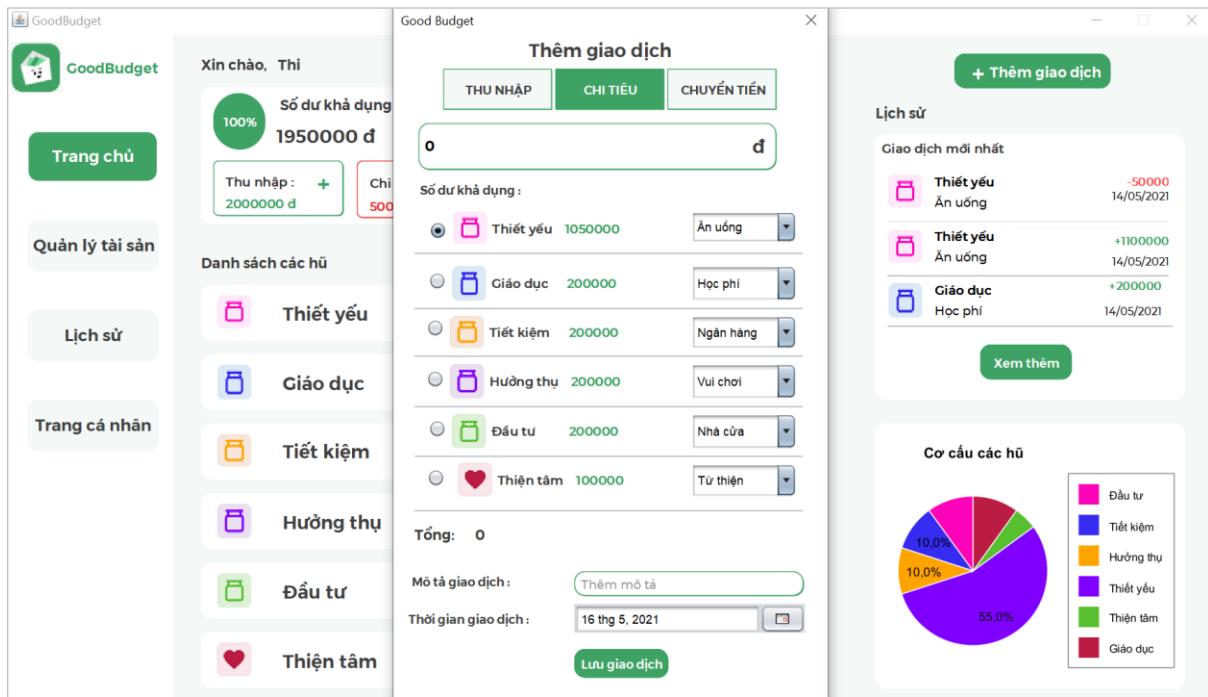


L'erreur de l'utilisateur n'a entré le



L'erreur de pourcentage total des

2. Interface l'écran d'exécution la fonctionnalité d'ajout de transactions de dépenses.



- Similaire à la fonctionnalité ci-dessus, les utilisateurs entrent d'abord le montant dans la zone de saisie.
- Ensuite, choisissez le type de "pot" que vous souhaitez dépenser, remplissez les informations nécessaires. Enfin, cliquez sur le bouton « Enregistrer la transaction ».

Số dư khả dụng :

<input checked="" type="radio"/> Thiết yếu 1050000	Ăn uống
<input type="radio"/> Giáo dục 200000	Học phí
<input type="radio"/> Tiết kiệm 200000	Ngân hàng
<input type="radio"/> Hưởng thụ 200000	Vui chơi
<input type="radio"/> Đầu tư 200000	Nhà cửa
<input type="radio"/> Thiện tâm 100000	Tư thiện

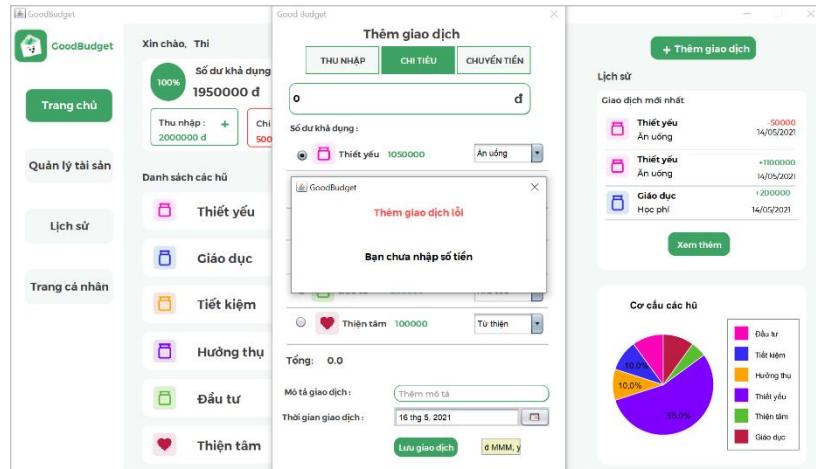
Tổng: 0

Mô tả giao dịch:

Thời gian giao dịch:

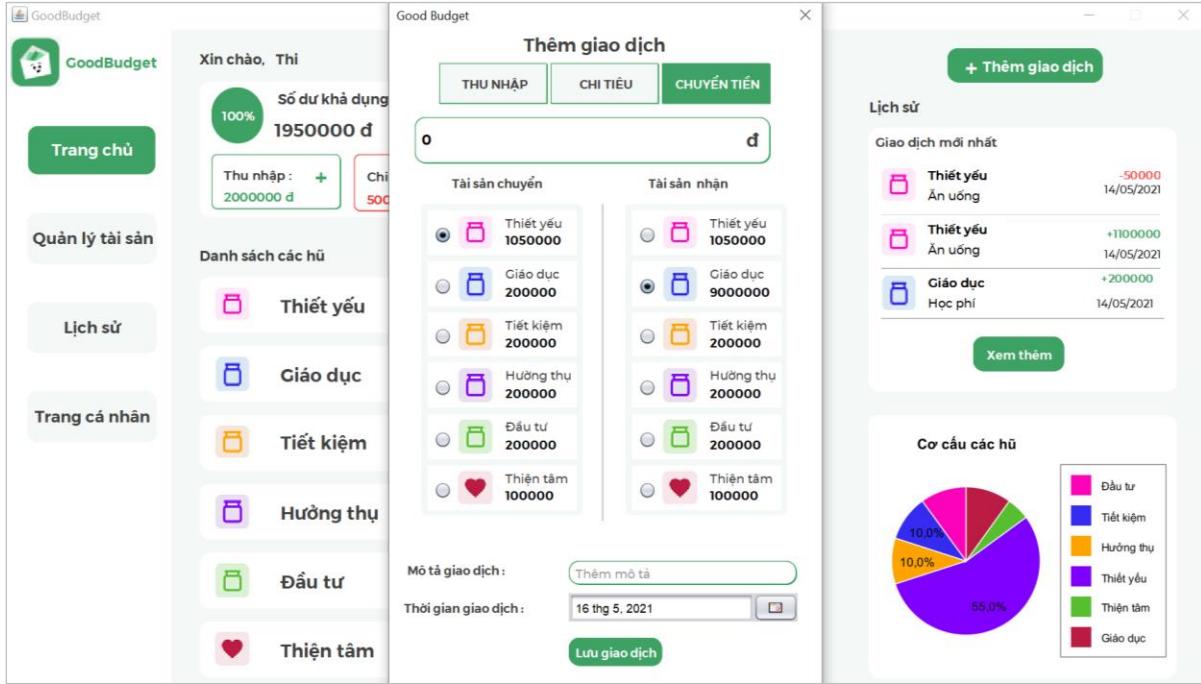
Lưu giao dịch

- S'il y a une erreur comme si l'utilisateur n'a pas entré le montant ou si le montant entré par l'utilisateur est supérieur au montant disponible du "pot", cela donne un message d'erreur.



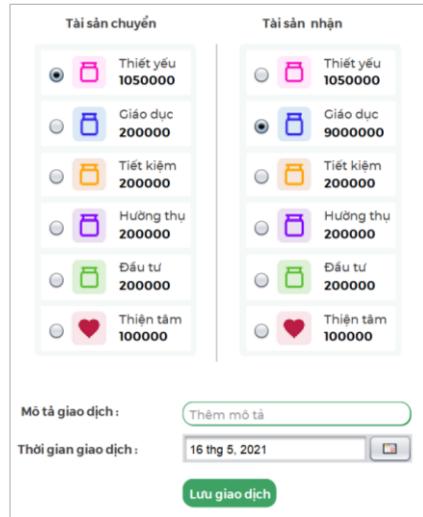
3. Interface l'écran d'exécution la fonctionnalité d'ajout de transactions de transfert d'argent.

- Similaire à les fonctionnalités ci-dessus, les utilisateurs entrent d'abord le montant dans la

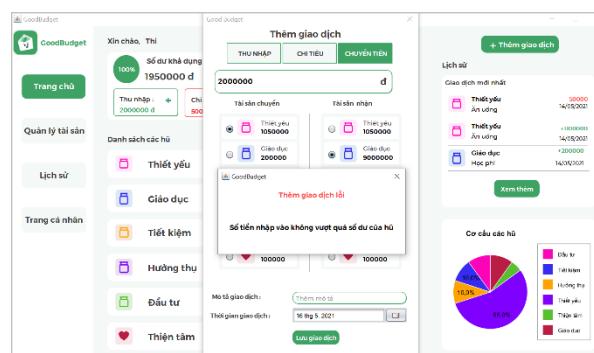
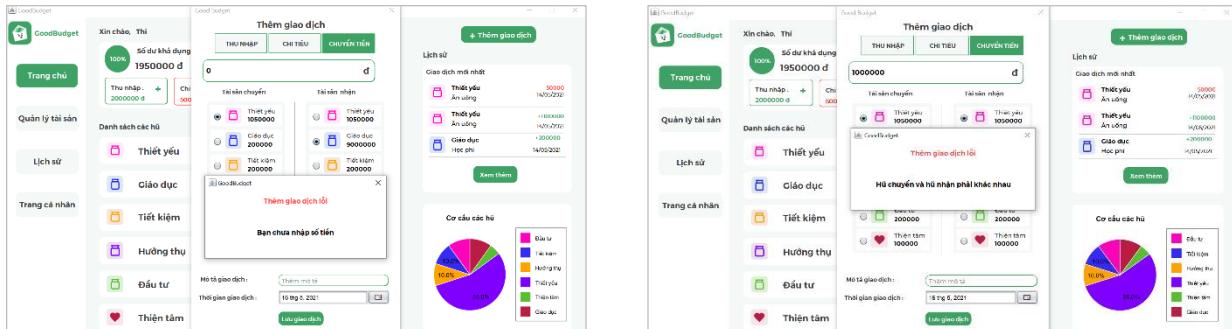


zone de saisie.

- Ensuite, l'utilisateur sélectionne le type "jar" pour transférer de l'argent et sélectionne le type "jar" pour recevoir l'argent. Enfin, remplir les informations nécessaires et cliquer sur le bouton « Enregistrer la transaction ».



- S'il y a une erreur comme si l'utilisateur n'entre pas le montant ou choisit le « pot » pour transférer et le « pot » pour recevoir le même, ou le montant entré est plus grand que le montant disponible dans le «pot», il donnera un message d'erreur.



III. Interface de l'écran de gestion de la propriété.

1. Statistiques sur les actifs des utilisateurs.

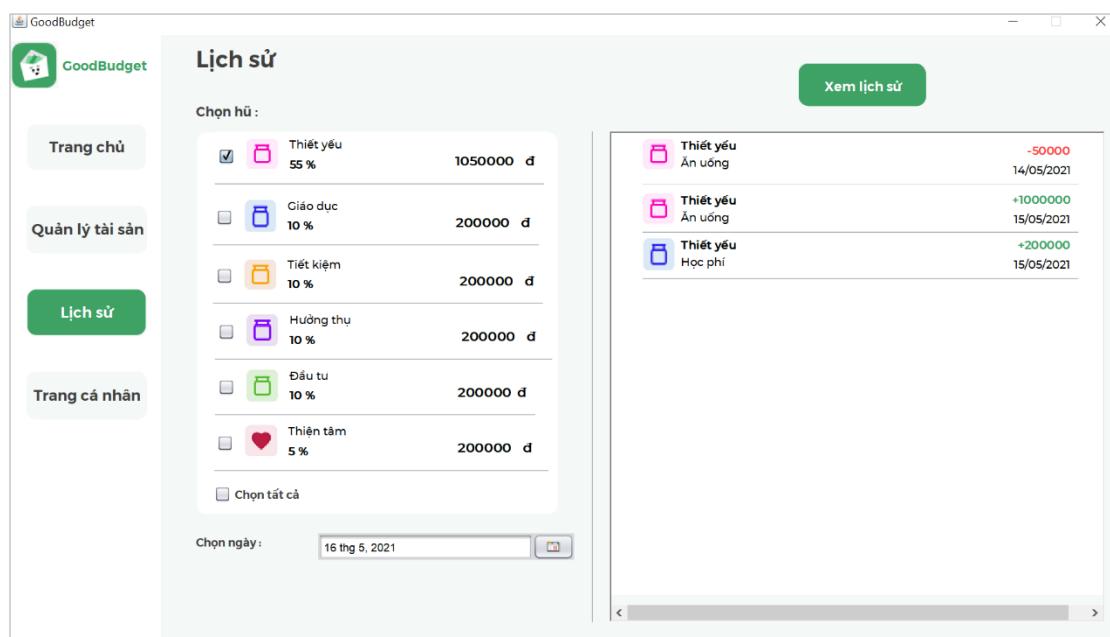
Afficher des graphiques pour les statistiques des actifs des utilisateurs par mois.



2. Afficher le graphique montrant la structure des "pots".



IV. Interface de l'écran de l'historique des transactions



- Les utilisateurs sélectionnent les options pour lesquelles ils souhaitent afficher l'historique et appuient sur le bouton « Afficher l'historique »

Chọn hủ :

<input checked="" type="checkbox"/>	 Thiết yêu	55 %	1050000 ₫
<input type="checkbox"/>	 Giáo dục	10 %	200000 ₫
<input type="checkbox"/>	 Tiết kiệm	10 %	200000 ₫
<input type="checkbox"/>	 Hưởng thụ	10 %	200000 ₫
<input type="checkbox"/>	 Đầu tư	10 %	200000 ₫
<input type="checkbox"/>	 Thiện tâm	5 %	200000 ₫
<input type="checkbox"/>	Chọn tất cả		

Chọn ngày :

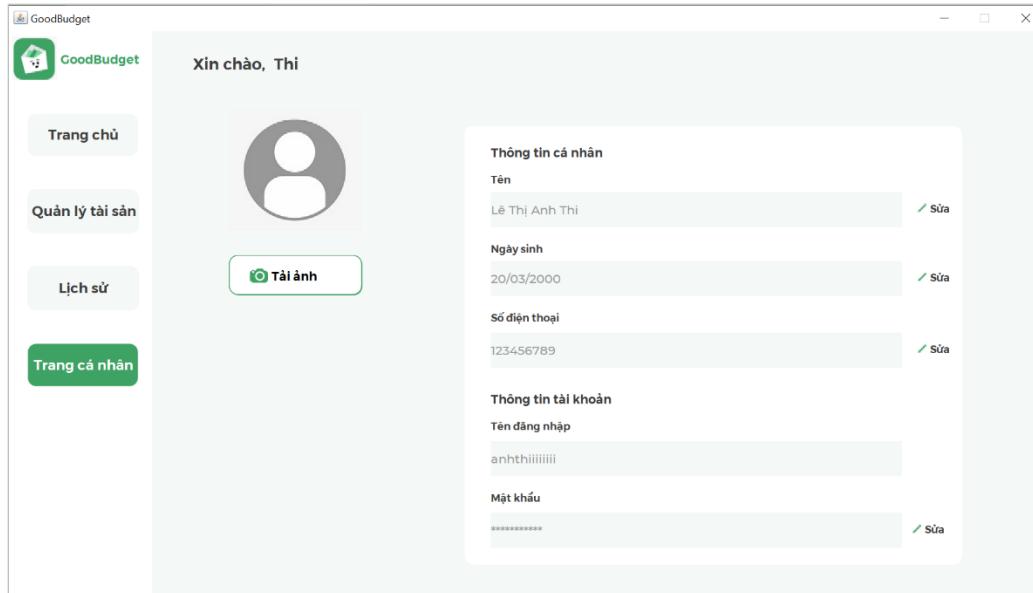
Xem lịch sử

- Et on obtient la sortie suivante:

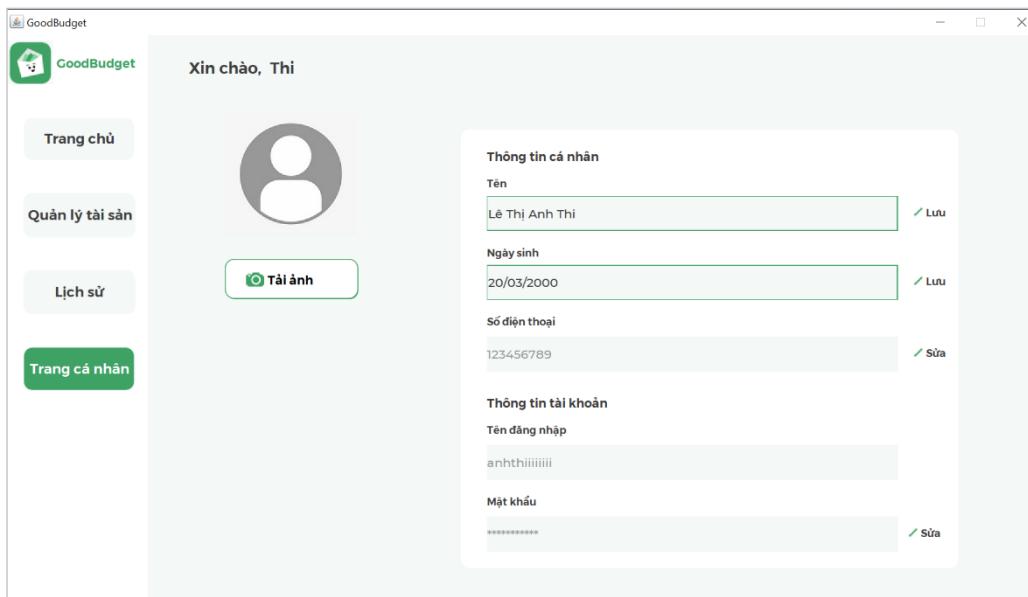
 Thiết yêu	Án uống	-50000	14/05/2021
 Thiết yêu	Án uống	+1000000	15/05/2021
 Thiết yêu	Học phí	+200000	15/05/2021

V. Interface d'écran de profil utilisateur.

- Affiche les informations personnelles des utilisateurs. En outre, l'application permet également aux utilisateurs de modifier leurs informations ici.



- Lorsque l'utilisateur clique sur le bouton « Modifier », il affichera ce qui suit :



Désormais, l'utilisateur peut modifier les informations dans les champs de saisie. Cliquer ensuite sur le bouton « Enregistrer » pour enregistrer les modifications.

Résultat

À propos de la base de données : Nous construisons une base de données afin que l'application puisse fonctionner de la meilleure façon en utilisant et en appliquant les connaissances acquises dans les matières. Grâce à cela, nous avons plus d'expérience dans les projets à venir.

À propos de la structure organisationnelle de la POO : à partir de la description des exigences de l'application, nous pouvons modéliser les implémentations et les relations entre elles. Nous avons appliqué les connaissances OOP que nous avons acquises dans le projet de cours.

À propos de l'interface utilisateur(GUI) : nous avons l'interface utilisateur complète. L'interface répond aux exigences suivantes: simple, facile à utiliser, rapide et claire. Il répond donc aux exigences données dans la description.

- Avantages :

- L'application n'est pas trop compliquée et peut proposer des designs rapidement.

- Désavantages :

- Difficulté à concevoir des interfaces avec Swing, sujettes à des erreurs de mise en page et difficile à assurer responsive.

- La base de données est difficile à partager, il y a donc des insuffisances lors du travail sur des projets entre les membres de l'équipe.

Dans le travail futur, nous voulons développer plus de fonction de gestion de la dette pour les utilisateurs. De plus, nous souhaitons développer une version Android et Web utilisable. L'application simple fait partie de nos capacités

Références

<https://netbeans.apache.org/kb/docs/java/quickstart-gui.html>

<https://www.javatpoint.com/java-swing>

http://www.java2s.com/Tutorial/Java/0240__Swing/Catalog0240__Swing.htm

<https://www.tutorialspoint.com/java/index.htm>

<https://www.w3schools.com/java/>

<https://www.javatpoint.com/java-tutorial>