

Développeur d'application - Java

Projet 6 :
Concevez une application web Java de A à Z

Badr Benmammar

bader.benmaamar@gmail.com

PLAN

- ❑ **Contexte du projet**
- ❑ **Besoins**
 - ❑ Fonctionnalités requises
- ❑ **Diagramme de classe UML**
- ❑ **Modèle physique de données**
- ❑ **Scripts SQL de base de données**
- ❑ **Solutions techniques**
 - ❑ Application Web avec Java et Spring Boot
 - ❑ Utilisation de contrôleurs pour gérer les requêtes HTTP
 - ❑ Spring Security pour sécuriser l'application web en gérant l'authentification
 - ❑ Spring Data JPA pour gérer la persistance des données
- ❑ **Couverture de test**
- ❑ **Diagrammes de cas d'utilisation**
- ❑ **Présentation de l'application**

Contexte du projet

☐ **Contexte :**

- ☐ Une start-up technologique qui fournit des solutions bancaires et financières.

☐ **Objectif :**

- ☐ Une application qui permettrait aux clients de transférer de l'argent pour gérer leurs finances ou payer leurs amis.

☐ **Problème à résoudre :** transactions d'argent peu pratiques :

- ☐ Les banques ont actuellement un processus long et peu pratique de configuration d'un transfert d'argent.
- ☐ Les transferts bancaires demandent trop de données (numéro de compte, code SWIFT, etc.).
- ☐ Il est difficile de rembourser ou de transférer de l'argent à des amis ou de la famille.
- ☐ Il est difficile de transférer de l'argent vers des comptes pour des achats.

☐ **Solution :**

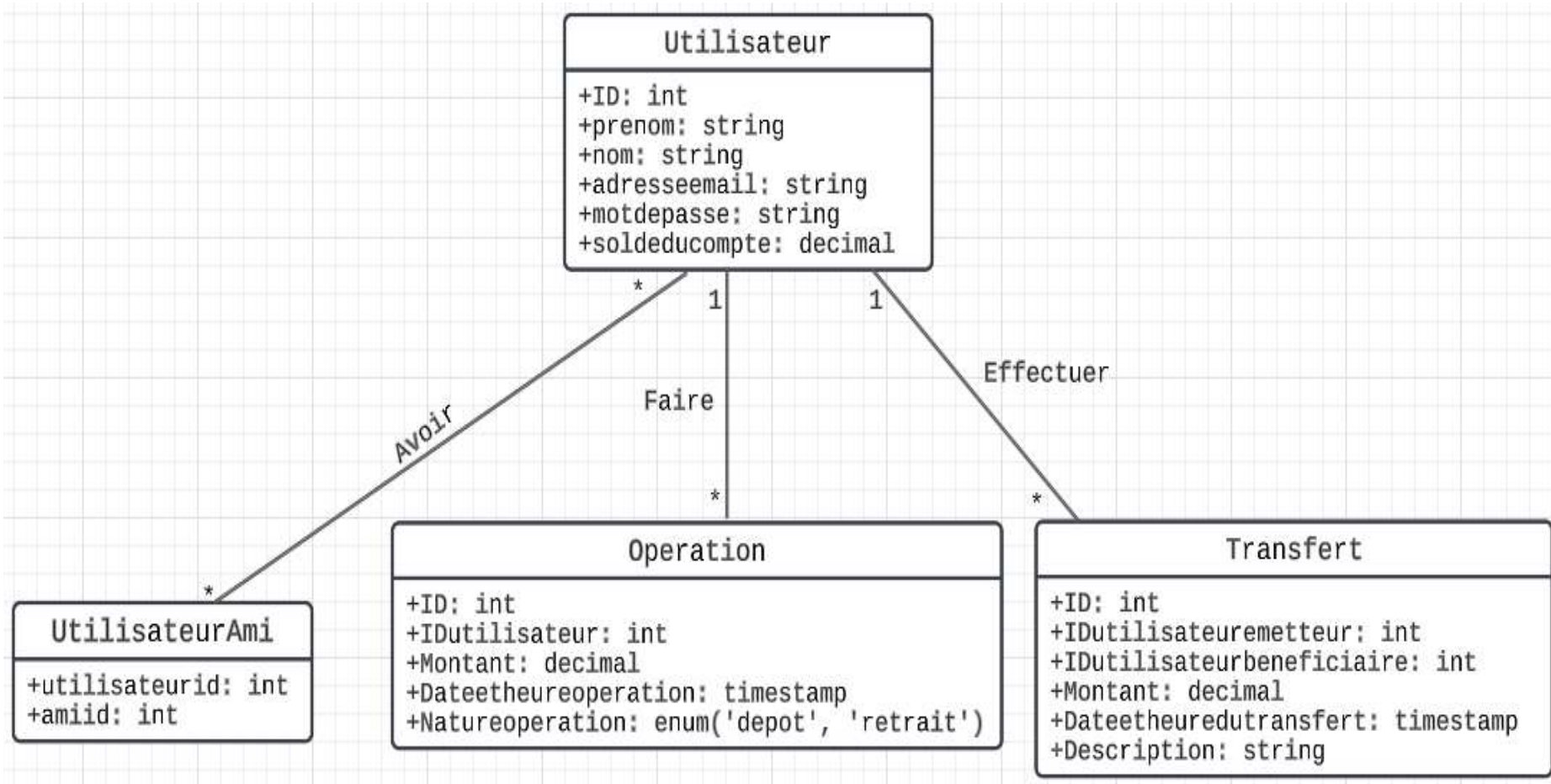
- ☐ Développer une application où les utilisateurs pourraient s'enregistrer facilement avec une adresse e-mail ou un compte de réseaux sociaux.
- ☐ Les utilisateurs peuvent ajouter des amis à leur réseau pour leur transférer de l'argent.
- ☐ Passer par une conception simple pour rationaliser la procédure et éviter les soucis.

Fonctionnalités requises

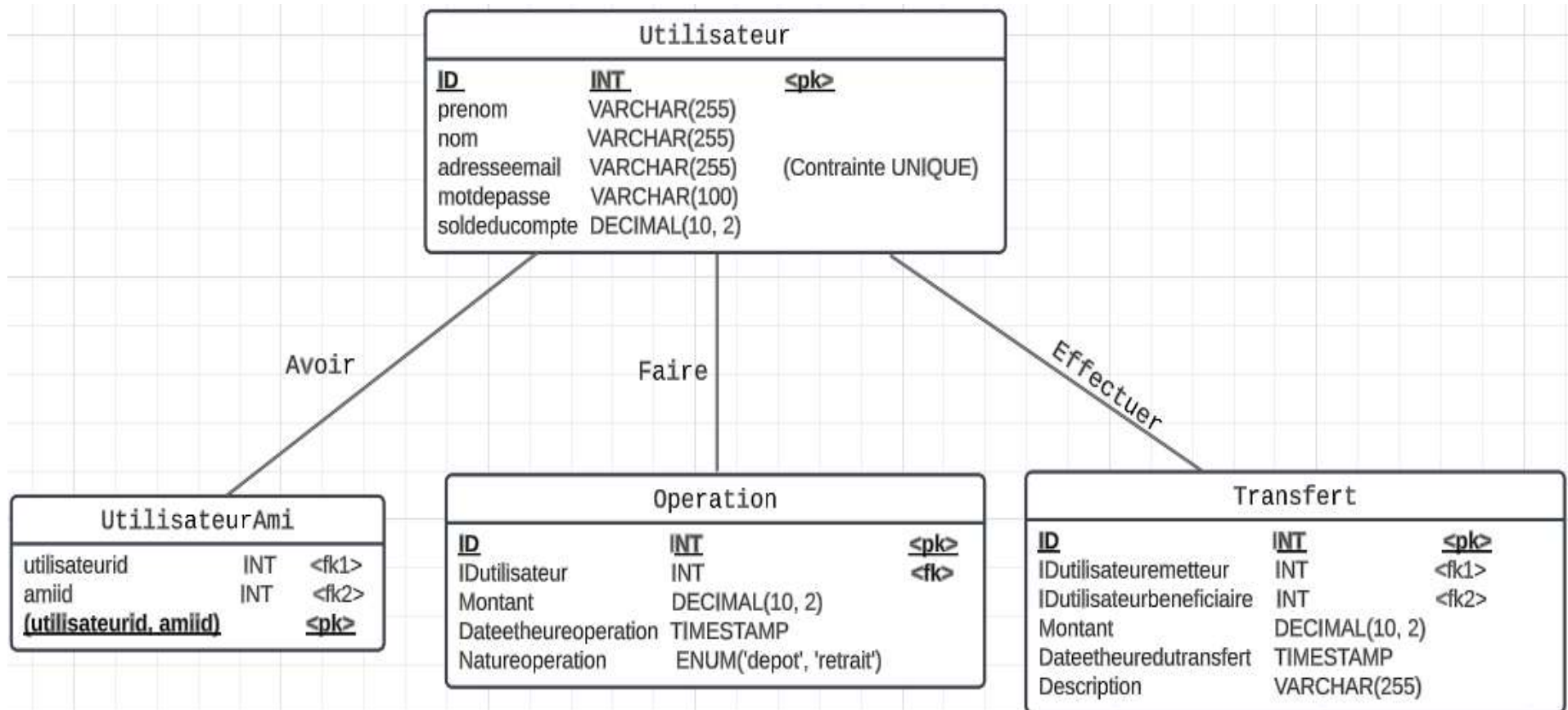
❑ Besoins/fonctionnalités :

- ❑ Les nouveaux utilisateurs doivent pouvoir s'enregistrer à l'aide d'un identifiant e-mail unique.
- ❑ Les utilisateurs doivent pouvoir se connecter à partir de leurs comptes dans la base de données.
- ❑ Après la connexion, les utilisateurs peuvent ajouter des personnes à leurs listes à partir de leur adresse e-mail (si la personne existe déjà dans la base de données).
- ❑ Un utilisateur peut verser de l'argent sur son compte dans notre application.
- ❑ À partir du solde disponible, les utilisateurs peuvent effectuer des paiements à leurs amis enregistrés sur l'application.
- ❑ À tout moment, les utilisateurs peuvent transférer l'argent vers leur compte bancaire.
- ❑ À chaque transaction, nous prélevons un pourcentage de 0,5 % pour monétiser l'application.

Diagramme de classe UML



Modèle physique de données



Scripts SQL de base de données

```
CREATE TABLE projet.utilisateur (  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  prenom VARCHAR(255),  
  nom VARCHAR(255),  
  adresseemail VARCHAR(255),  
  motdepasse VARCHAR(100),  
  soldeducompte DECIMAL(10, 2),  
  CONSTRAINT uniqueemail UNIQUE (adresseemail)  
);
```

```
CREATE TABLE projet.utilisateurami (  
  utilisateurid INT,  
  amiid INT,  
  FOREIGN KEY (utilisateurid) REFERENCES projet.utilisateur(ID),  
  FOREIGN KEY (amiid) REFERENCES projet.utilisateur(ID),  
  CONSTRAINT pkutilisateurami PRIMARY KEY (utilisateurid, amiid)  
);
```

<https://github.com/baderbbm/Benmammar-Badr-projet6/blob/main/src/SQL/tables.sql>

```
CREATE TABLE projet.operation (  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  IDutilisateur INT,  
  Montant DECIMAL(10, 2),  
  Dateetheureoperation TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  Natureoperation ENUM('depot', 'retrait'),  
  FOREIGN KEY (IDutilisateur) REFERENCES utilisateur(ID)  
);
```

```
CREATE TABLE projet.transfert (  
  ID INT PRIMARY KEY AUTO_INCREMENT,  
  IDutilisateurmetteur INT,  
  IDutilisateurbeneficiaire INT,  
  Montant DECIMAL(10, 2),  
  Dateetheuredutransfert TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  Description VARCHAR(255),  
  FOREIGN KEY (IDutilisateurmetteur) REFERENCES utilisateur(ID),  
  FOREIGN KEY (IDutilisateurbeneficiaire) REFERENCES utilisateur(ID)  
);
```

Application Web avec Java et Spring Boot

- ❑ Pour développer cette application, nous avons choisi d'utiliser Java comme langage de programmation principal.
- ❑ Java est un langage populaire et puissant, bien adapté au développement d'applications web robustes et sécurisées. En utilisant Java, nous pouvons tirer parti de ses fonctionnalités orientées objet, sa portabilité et sa grande communauté de développeurs.
- ❑ De plus, nous avons opté pour le Framework Spring Boot pour faciliter le développement de l'application.
- ❑ Spring Boot offre un ensemble d'outils et de bibliothèques qui simplifient le processus de développement en fournissant des configurations par défaut et en automatisant certaines tâches courantes.
- ❑ Cela permet aux développeurs de se concentrer davantage sur la logique métier plutôt que sur la configuration.

Utilisation de contrôleurs pour gérer les requêtes HTTP

- ❑ Dans l'architecture REST (Representational State Transfer), les contrôleurs jouent un rôle essentiel. Ce sont des composants de l'application qui reçoivent les requêtes HTTP, traitent les données et renvoient les réponses appropriées. **Ils agissent comme une couche intermédiaire entre les clients (navigateurs, applications tierces) et les services métier de l'application.**
- ❑ En utilisant des contrôleurs, nous pouvons définir des points de terminaison (endpoints) pour chaque action spécifique que l'application doit prendre en charge. Cela permet une organisation claire du code et une séparation des responsabilités.
- ❑ Deux dépendances à mentionner :
 - ❑ **spring-boot-starter-web** : une dépendance qui fournit un ensemble de fonctionnalités de base pour développer des applications web avec Spring Boot. Cette dépendance inclut des composants essentiels pour le développement web, tels que **Spring MVC (Model-View-Controller)** pour la gestion des requêtes HTTP, **la prise en charge des templates (Thymeleaf, etc.)**, et la configuration de base pour gérer **les ressources statiques (HTML, CSS, JavaScript)**.
 - ❑ **spring-boot-starter-thymeleaf** : une dépendance qui sert à intégrer Thymeleaf, un moteur de template, dans une application Spring Boot.

Endpoints

- ❑ GET /enregistrerUtilisateur : renvoie la vue "enregistrer-utilisateur".
- ❑ POST /enregistrerUtilisateur : enregistrer un utilisateur.
- ❑ GET /ajouterAmi : renvoie la vue "ajouter-ami".
- ❑ POST /ajouterAmi : ajouter un ami.
- ❑ GET /effectuerDepot : renvoie la vue "effectuerDepot".
- ❑ POST /effectuerDepot : effectuer un dépôt.
- ❑ GET /effectuerRetrait : renvoie la vue "effectuerRetrait".
- ❑ POST /effectuerRetrait : effectuer un retrait.
- ❑ GET /effectuerVirement : renvoie la vue "effectuerVirement".
- ❑ POST /effectuerVirement : effectuer un virement.
- ❑ GET /historiqueOperations : renvoie la vue "historique-operations".
- ❑ GET /login : renvoie la vue "login" pour l'authentification.
- ❑ GET /home : renvoie la vue "home".

Packages du projet

```
src/main/java
├── com.openclassrooms.DataLayerSec
│   ├── DataLayerSecApplication.java
│   ├── com.openclassrooms.DataLayerSec.config
│   │   ├── CustomUserDetails.java
│   │   ├── CustomUserDetailsService.java
│   │   └── SecurityConfig.java
│   ├── com.openclassrooms.DataLayerSec.controller
│   │   └── UtilisateurController.java
│   ├── com.openclassrooms.DataLayerSec.dto
│   │   ├── OperationDTO.java
│   │   ├── TransfertDTO.java
│   │   └── UtilisateurDTO.java
│   ├── com.openclassrooms.DataLayerSec.exceptions
│   │   ├── EmailExistsException.java
│   │   └── InsufficientBalanceException.java
│   ├── com.openclassrooms.DataLayerSec.model
│   │   ├── NatureOperation.java
│   │   ├── Operation.java
│   │   ├── Transfert.java
│   │   └── Utilisateur.java
│   ├── com.openclassrooms.DataLayerSec.repository
│   │   ├── OperationRepository.java
│   │   ├── TransfertRepository.java
│   │   └── UtilisateurRepository.java
│   └── com.openclassrooms.DataLayerSec.service
│       ├── OperationService.java
│       ├── TransfertService.java
│       └── UtilisateurService.java
```

dto (Data Transfer Objects): contient des classes qui servent à transférer les données entre les couches de notre application

Couche DAL/référentiel

Spring Data JPA pour gérer la persistance des données en utilisant des objets Java pour représenter les données stockées dans les bases de données

Packages du projet

```
src/main/java
├── com.openclassrooms.DataLayerSec
│   ├── DataLayerSecApplication.java
│   ├── com.openclassrooms.DataLayerSec.config
│   │   ├── CustomUserDetails.java
│   │   ├── CustomUserDetailsService.java
│   │   └── SecurityConfig.java
│   ├── com.openclassrooms.DataLayerSec.controller
│   │   └── UtilisateurController.java
│   ├── com.openclassrooms.DataLayerSec.dto
│   │   ├── OperationDTO.java
│   │   ├── TransfertDTO.java
│   │   └── UtilisateurDTO.java
│   ├── com.openclassrooms.DataLayerSec.exceptions
│   │   ├── EmailExistsException.java
│   │   └── InsufficientBalanceException.java
│   ├── com.openclassrooms.DataLayerSec.model
│   │   ├── NatureOperation.java
│   │   ├── Operation.java
│   │   ├── Transfert.java
│   │   └── Utilisateur.java
│   ├── com.openclassrooms.DataLayerSec.repository
│   │   ├── OperationRepository.java
│   │   ├── TransfertRepository.java
│   │   └── UtilisateurRepository.java
│   └── com.openclassrooms.DataLayerSec.service
│       ├── OperationService.java
│       ├── TransfertService.java
│       └── UtilisateurService.java
```

dto (Data Transfer Objects): contient des classes qui servent à transférer les données entre les couches de notre application

Couche DAL/référentiel

Spring Data JPA pour gérer la persistance des données en utilisant des objets Java pour représenter les données stockées dans les bases de données

Packages du projet

```
src/main/java
├── com.openclassrooms.DataLayerSec
│   ├── DataLayerSecApplication.java
│   ├── com.openclassrooms.DataLayerSec.config
│   │   ├── CustomUserDetails.java
│   │   ├── CustomUserDetailsService.java
│   │   └── SecurityConfig.java
│   ├── com.openclassrooms.DataLayerSec.controller
│   │   └── UtilisateurController.java
│   ├── com.openclassrooms.DataLayerSec.dto
│   │   ├── OperationDTO.java
│   │   ├── TransfertDTO.java
│   │   └── UtilisateurDTO.java
│   ├── com.openclassrooms.DataLayerSec.exceptions
│   │   ├── EmailExistsException.java
│   │   └── InsufficientBalanceException.java
│   ├── com.openclassrooms.DataLayerSec.model
│   │   ├── NatureOperation.java
│   │   ├── Operation.java
│   │   ├── Transfert.java
│   │   └── Utilisateur.java
│   ├── com.openclassrooms.DataLayerSec.repository
│   │   ├── OperationRepository.java
│   │   ├── TransfertRepository.java
│   │   └── UtilisateurRepository.java
│   └── com.openclassrooms.DataLayerSec.service
│       ├── OperationService.java
│       ├── TransfertService.java
│       └── UtilisateurService.java
```

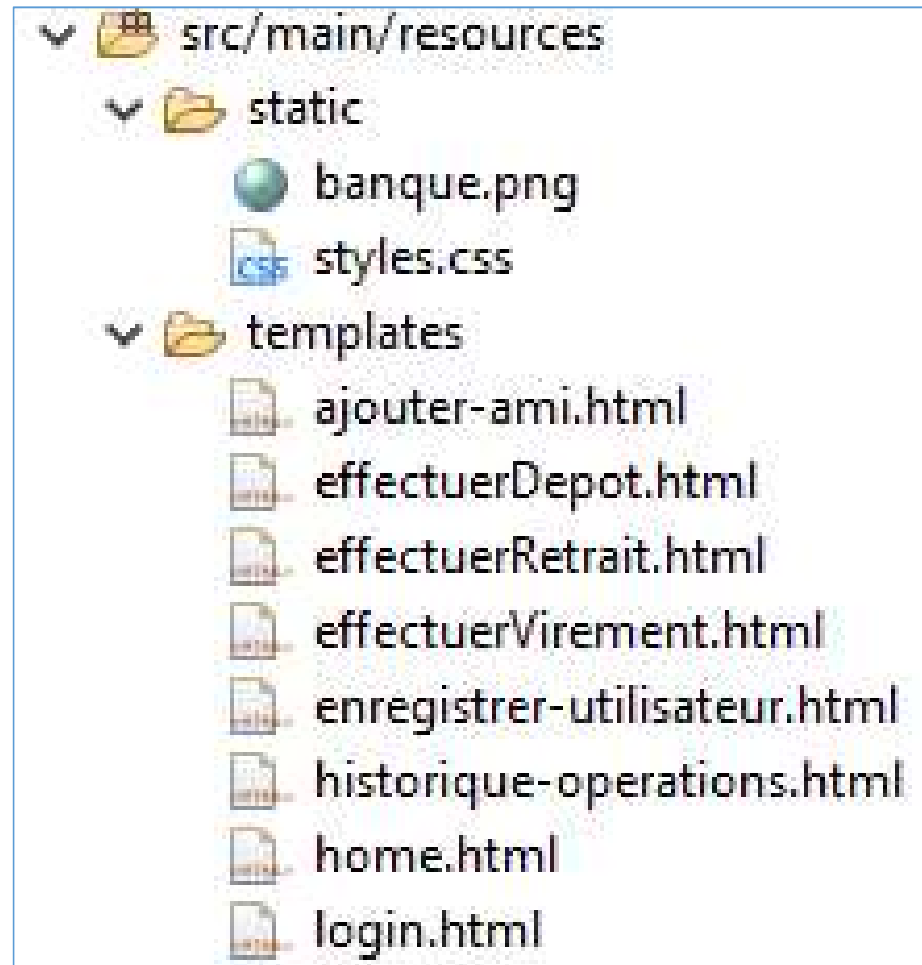
Spring Security pour sécuriser l'application web en gérant l'authentification

dto (Data Transfer Objects): contient des classes qui servent à transférer les données entre les couches de notre application

Couche DAL/référentiel

Spring Data JPA pour gérer la persistance des données en utilisant des objets Java pour représenter les données stockées dans les bases de données

Interface web



application.properties

❑ Utilisation de variables d'environnement :

```
spring.datasource.url=${SPRING_DATASOURCE_URL}
spring.datasource.username=${SPRING_DATASOURCE_USERNAME}
spring.datasource.password=${SPRING_DATASOURCE_PASSWORD}
spring.jpa.properties.hibernate.dialect=${SPRING_JPA_PROPERTIES_HIBERNATE_DIALECT}
```















SPRING_DATASOURCE_PASSWORD	rootroot
SPRING_DATASOURCE_URL	jdbc:mysql://localhost:3306/projet?serverTimezone=UTC
SPRING_DATASOURCE_USERNAME	root
SPRING_JPA_PROPERTIES_HIBERNATE_DIALECT	org.hibernate.dialect.MySQL5Dialect

mvn test et mvn jacoco:report

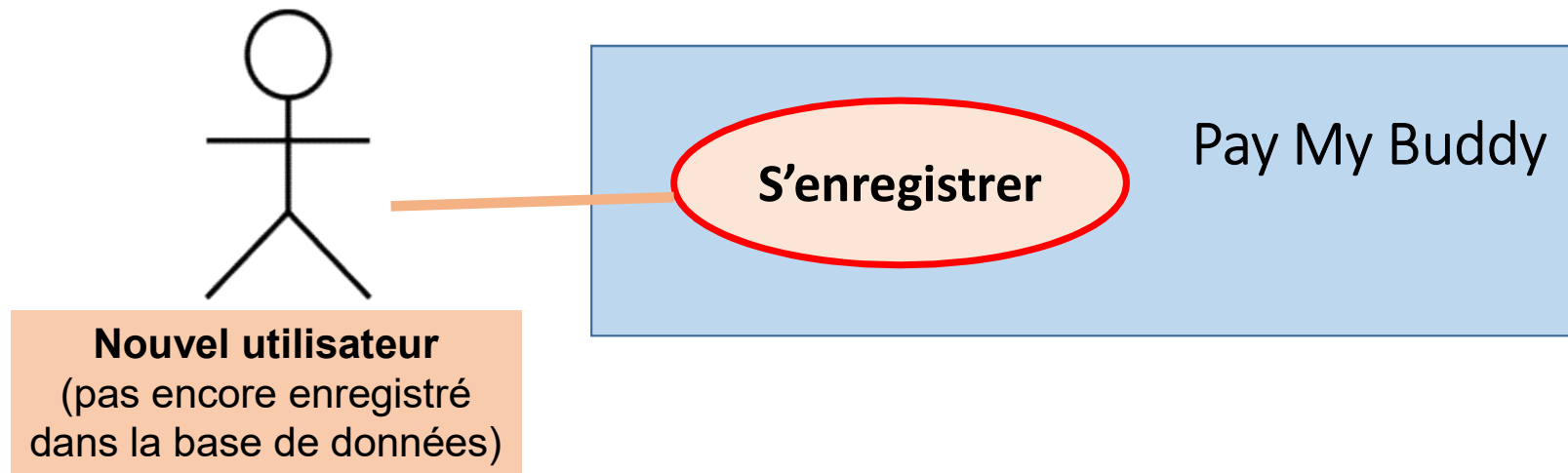
```
[INFO] Results:
[INFO] Tests run: 36, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 14.270 s
[INFO] Finished at: 2023-09-19T16:40:47+02:00
[INFO]
ntic@DESKTOP-10DQFQK /cygdrive/c/Users/ntic/Documents/workspace-spring-tool-suite-4-4.19.1.RELEASE/DataLayerSec
$ mvn jacoco:report
[INFO] Scanning for projects...
[INFO]
[INFO] < com.openclassrooms:DataLayerSec >
[INFO] Building DataLayerSec 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO]
[INFO] [ jar ]
[INFO]
[INFO] --- jacoco:0.8.7:report (default-cli) @ DataLayerSec ---
[INFO] Loading execution data file C:\Users\ntic\Documents\workspace-spring-tool-suite-4-4.19.1.RELEASE\DataLayerSec\target\jacoco.exec
[INFO] Analyzed bundle 'DataLayerSec' with 17 classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.245 s
[INFO] Finished at: 2023-09-19T16:40:55+02:00
[INFO]
```


Couverture de test

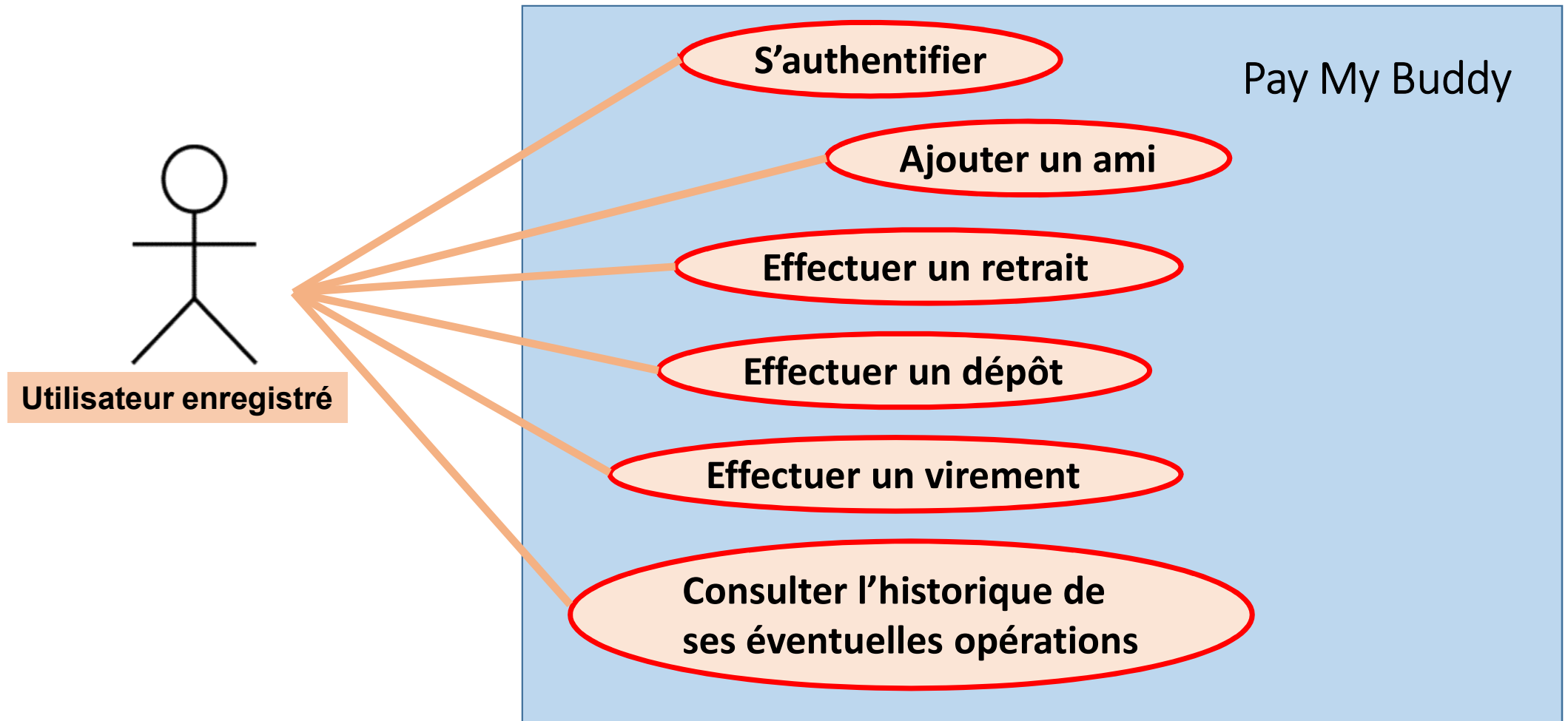
DataLayerSec

Element	Missed Instructions	Cov.
 com.openclassrooms.DataLayerSec.config		100 %
 com.openclassrooms.DataLayerSec.exceptions		100 %
 com.openclassrooms.DataLayerSec.service		93 %
 com.openclassrooms.DataLayerSec.model		88 %
 com.openclassrooms.DataLayerSec.controller		87 %
 com.openclassrooms.DataLayerSec.dto		84 %
 com.openclassrooms.DataLayerSec		44 %
Total	110 of 1 179	90 %

Diagrammes de cas d'utilisation



Diagrammes de cas d'utilisation

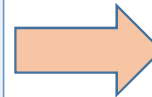


Un nouvel utilisateur doit pouvoir s'enregistrer à l'aide d'un identifiant e-mail unique

localhost:8080/enregistrerUtilisateur

Enregistrement Utilisateur

Nom:	<input type="text" value="ben"/>
Prénom:	<input type="text" value="badr"/>
Adresse Email:	<input type="text" value="badr@gmail.com"/>
Mot de passe:	<input type="password" value="....."/>
Solde du Compte:	<input type="text" value="5000"/>



localhost:8080/enregistrerUtilisateur

Enregistrement Utilisateur

Nom:	<input type="text"/>
Prénom:	<input type="text"/>
Adresse Email:	<input type="text"/>
Mot de passe:	<input type="password"/>
Solde du Compte:	<input type="text"/>

L'utilisateur a été enregistré avec succès.

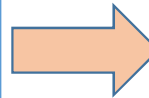
	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
▶	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21...	5000.00

Un nouvel utilisateur doit pouvoir s'enregistrer à l'aide d'un identifiant e-mail unique

localhost:8080/enregistrerUtilisateur

Enregistrement Utilisateur

Nom:	<input type="text" value="ben"/>
Prénom:	<input type="text" value="badr"/>
Adresse Email:	<input type="text" value="badr@gmail.com"/>
Mot de passe:	<input type="password" value="....."/>
Solde du Compte:	<input type="text" value="5000"/>



localhost:8080/enregistrerUtilisateur

Enregistrement Utilisateur

Nom:	<input type="text"/>
Prénom:	<input type="text"/>
Adresse Email:	<input type="text"/>
Mot de passe:	<input type="password"/>
Solde du Compte:	<input type="text"/>

L'email existe déjà.

Inscription de plusieurs utilisateur

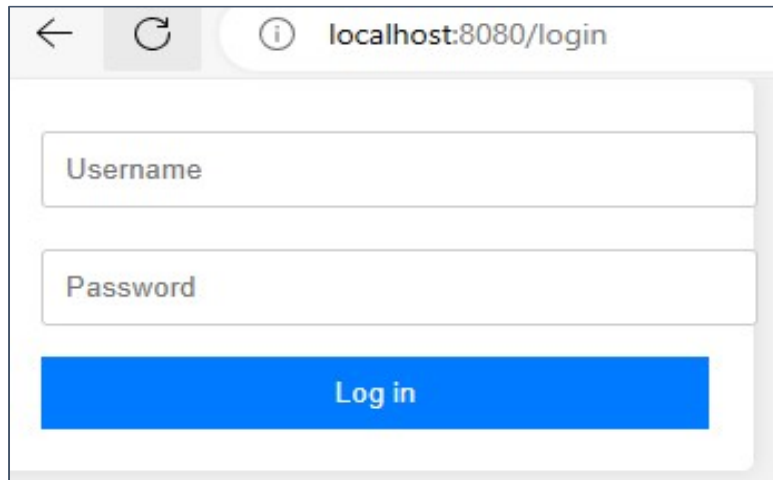
	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
▶	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKuNarC6SSkZ21QaK...	5000.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	6000.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

← → ↻ ⓘ localhost:8080/enregistrerUtilisateur

Enregistrement Utilisateur

Nom:	<input type="text" value="ben"/>
Prénom:	<input type="text" value="ines"/>
Adresse Email:	<input type="text" value="ines@gmail.com"/>
Mot de passe:	<input type="password" value="....."/>
Solde du Compte:	<input type="text" value="7000"/>

Un utilisateur doit pouvoir se connecter à partir de son compte dans la base de données

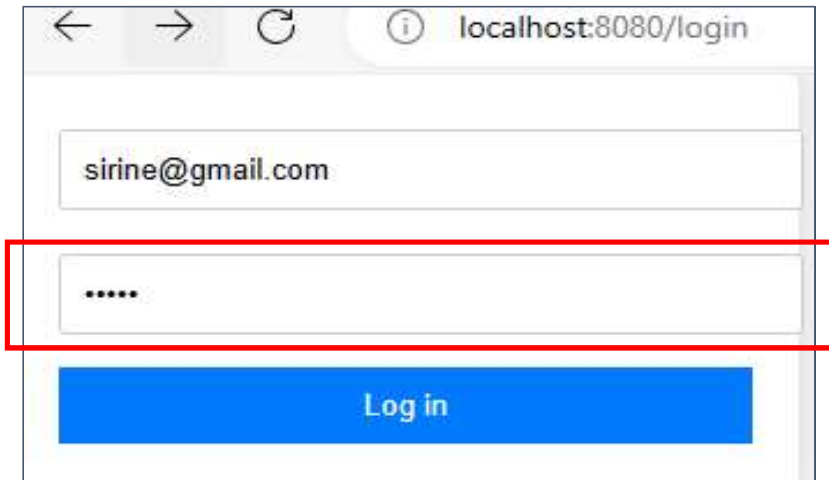


localhost:8080/login

Username

Password

Log in

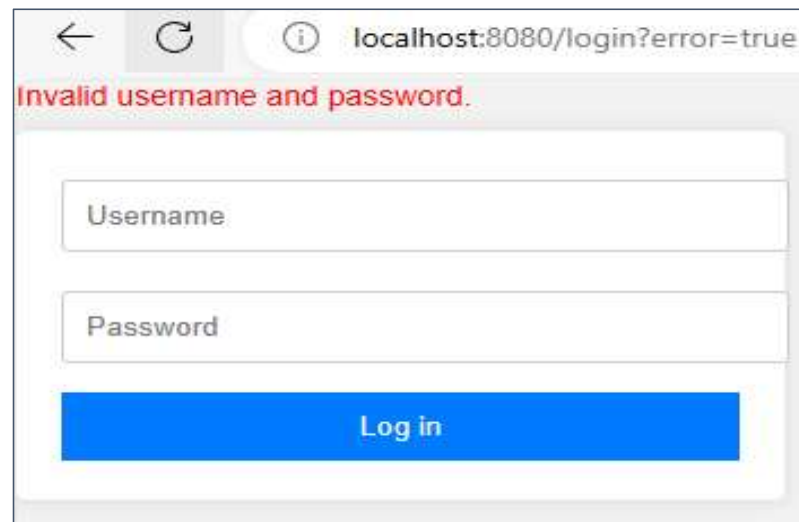


localhost:8080/login

sirine@gmail.com

.....

Log in



localhost:8080/login?error=true

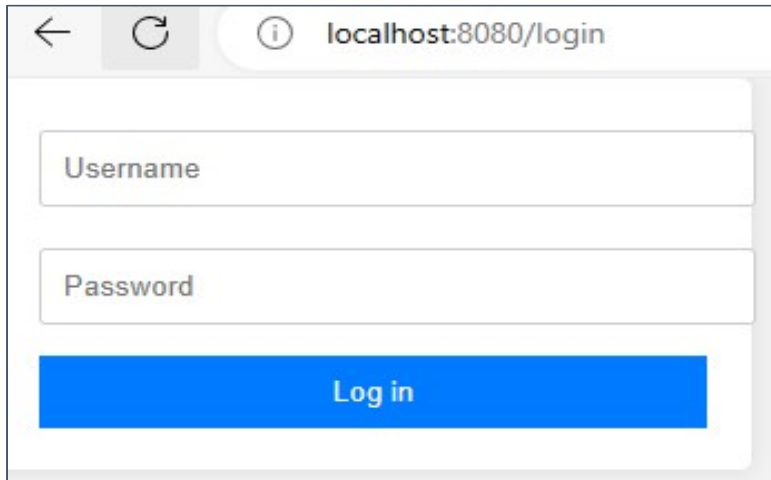
Invalid username and password.

Username

Password

Log in

Un utilisateur doit pouvoir se connecter à partir de son compte dans la base de données

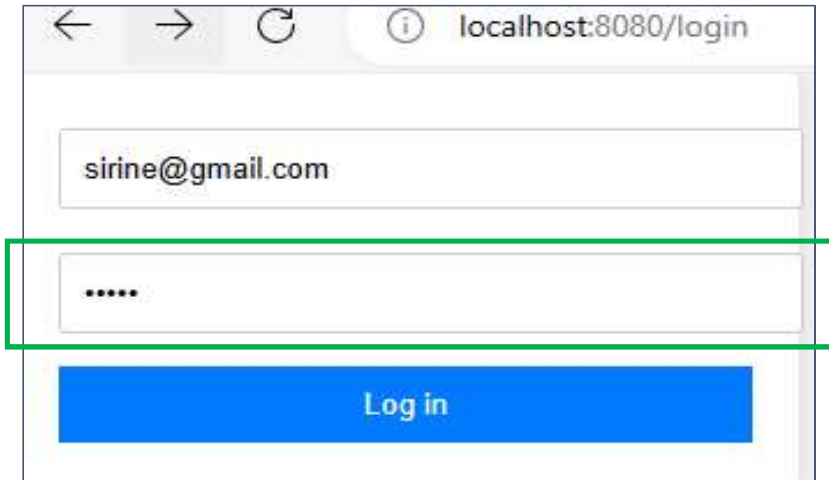


← ↻ ⓘ localhost:8080/login

Username

Password

Log in



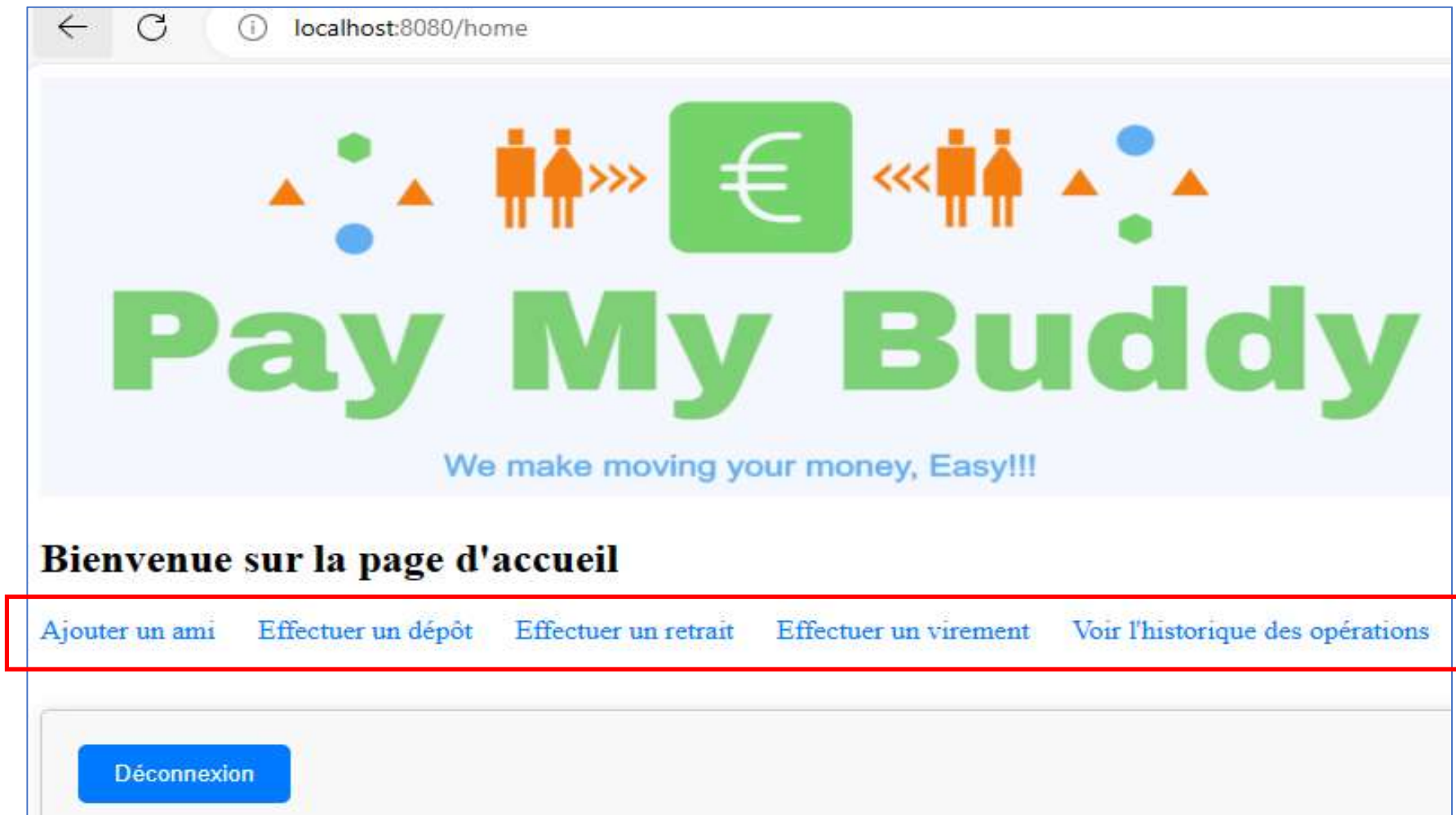
← → ↻ ⓘ localhost:8080/login

sirine@gmail.com

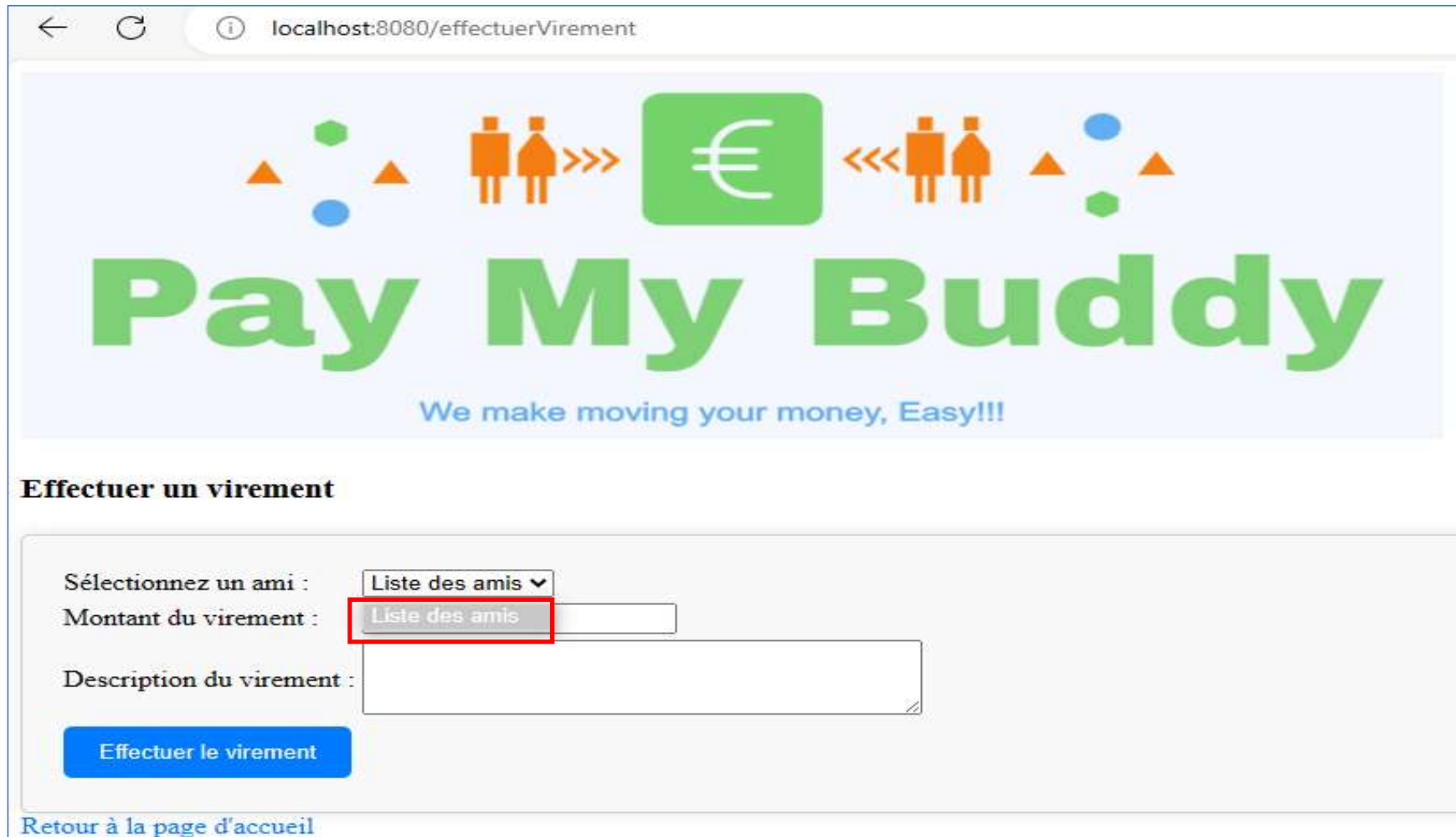
.....

Log in

Un utilisateur doit pouvoir se connecter à partir de son compte dans la base de données




Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/effectuerVirement'. The page features a header with a logo consisting of stylized orange and blue shapes, a green square with a white Euro symbol (€), and the text 'Pay My Buddy' in large green letters, followed by the tagline 'We make moving your money, Easy!!!' in blue. Below the header, the section is titled 'Effectuer un virement'. It contains three input fields: 'Sélectionnez un ami :', 'Montant du virement :', and 'Description du virement :'. The first field has a dropdown menu with 'Liste des amis' selected. The second field has a text input with 'Liste des amis' entered, which is highlighted by a red rectangle. The third field is a larger text area. A blue button labeled 'Effectuer le virement' is positioned below the input fields. At the bottom left, there is a link 'Retour à la page d'accueil'.

← ↻ ⓘ localhost:8080/effectuerVirement


Pay My Buddy
We make moving your money, Easy!!!

Effectuer un virement

Sélectionnez un ami :

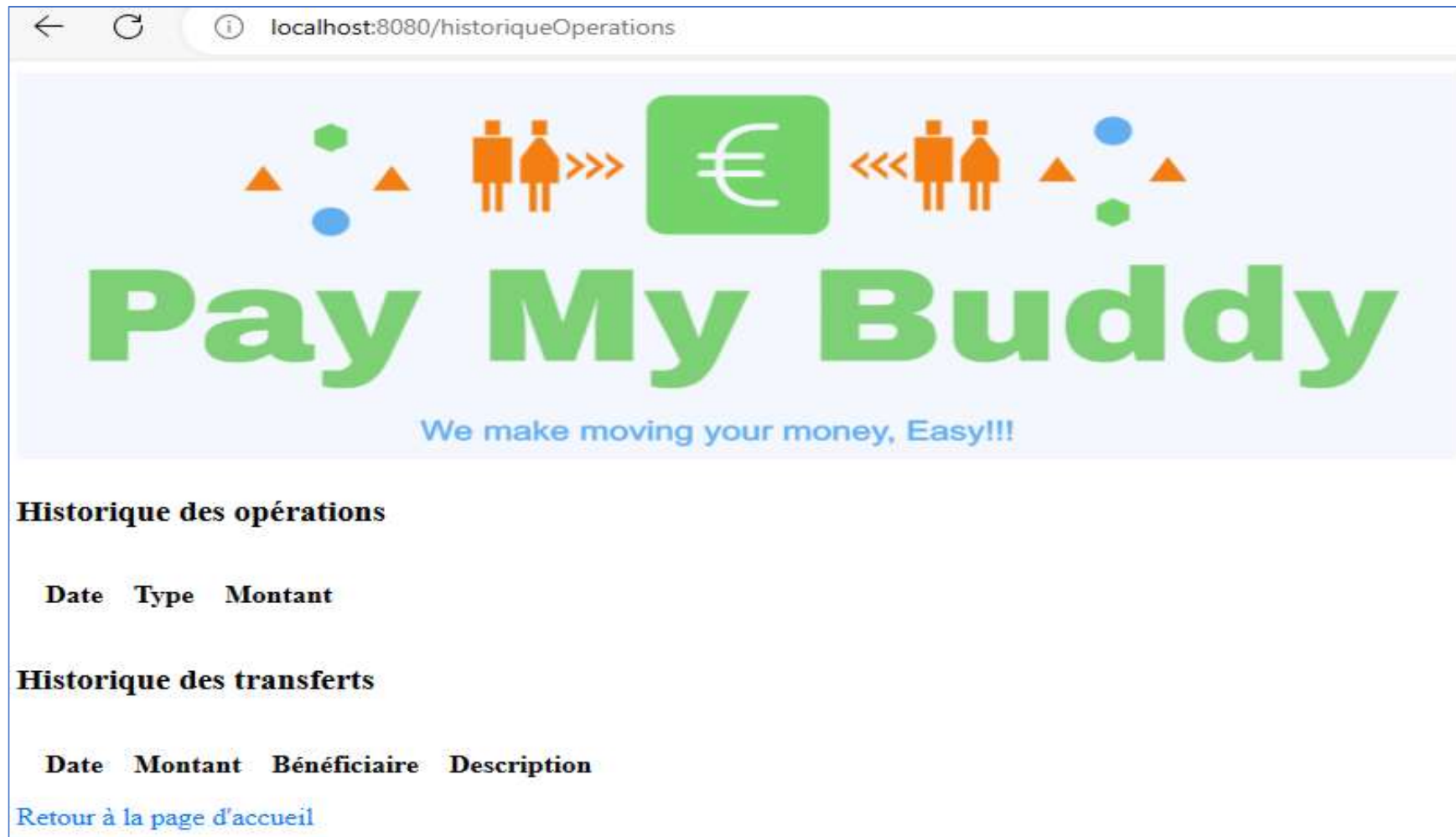
Montant du virement :

Description du virement :

[Retour à la page d'accueil](#)

[Effectuer le virement](#)


Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application



Un utilisateur peut ajouter des personnes à sa liste à partir de leur adresse e-mail



← ↻ ⓘ localhost:8080/ajouterAmi



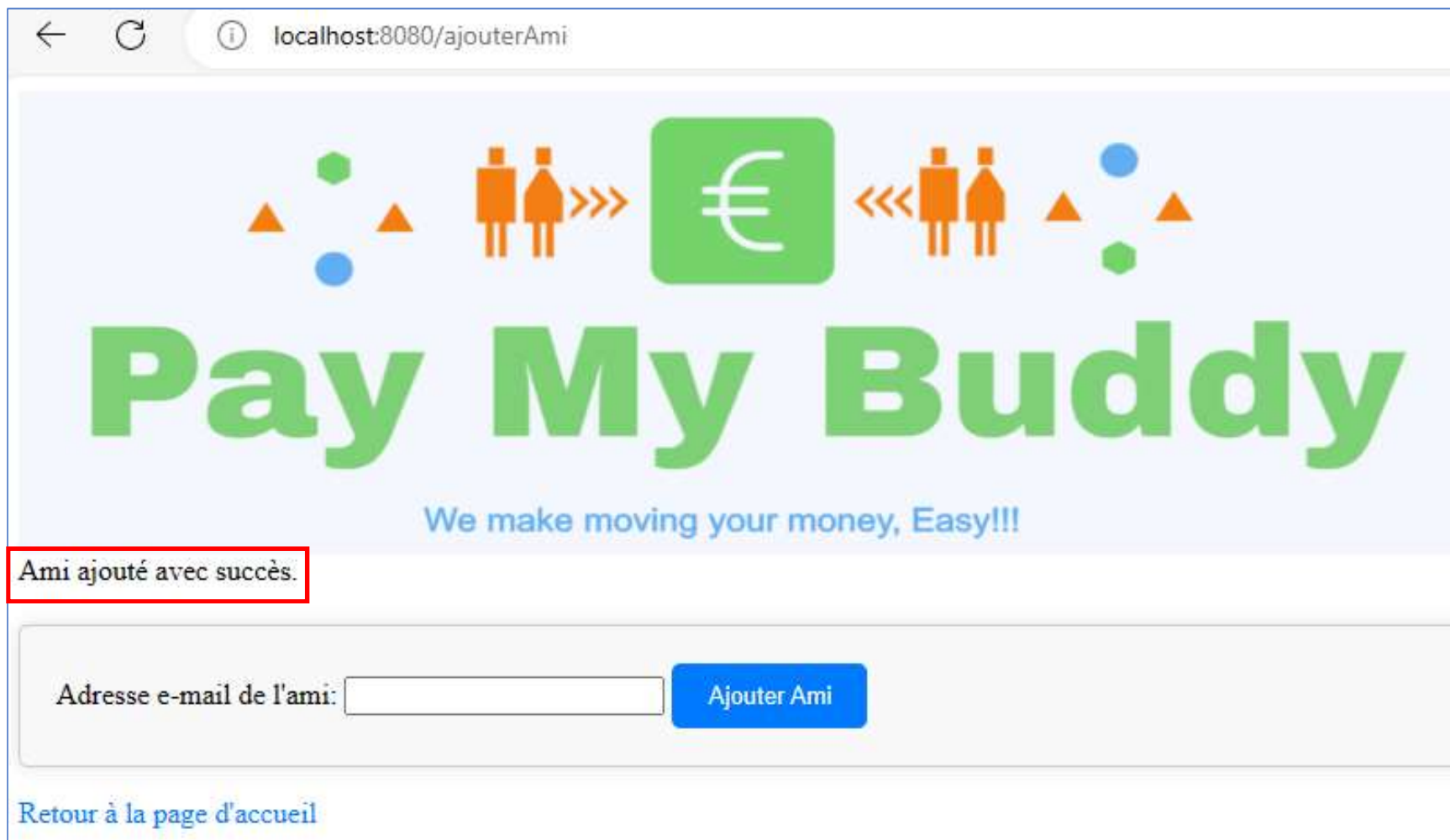
Pay My Buddy

We make moving your money, Easy!!!

Adresse e-mail de l'ami:


[Retour à la page d'accueil](#)

Un utilisateur peut ajouter des personnes à sa liste à partir de leur adresse e-mail



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ajouterAmi'. The page features a header with a logo consisting of two groups of stylized orange figures with arrows pointing towards a central green square containing a white Euro symbol (€). Below the logo, the text 'Pay My Buddy' is written in a large, bold, green font, followed by the tagline 'We make moving your money, Easy!!!' in a smaller, blue font. A red-bordered box highlights a green message: 'Ami ajouté avec succès.' Below this, there is a form with the label 'Adresse e-mail de l'ami:' followed by a text input field and a blue button labeled 'Ajouter Ami'. At the bottom left, there is a blue link that says 'Retour à la page d'accueil'.

← ↻ ⓘ localhost:8080/ajouterAmi



Pay My Buddy

We make moving your money, Easy!!!

Ami ajouté avec succès.

Adresse e-mail de l'ami:

[Ajouter Ami](#)


[Retour à la page d'accueil](#)

Un utilisateur peut ajouter des personnes à sa liste à partir de leur adresse e-mail

ID	prenom	nom	adresseemail	motdepasse	soldeducompte
27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	5000.00
29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	6000.00
30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOAljPUKrnkQmBn...	7000.00


utilisateurid	amiid
29	27
29	30

Un utilisateur peut ajouter des personnes à sa liste à partir de leur adresse e-mail



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ajouterAmi'. The page has a light blue header with a decorative graphic of two groups of people (represented by orange stick figures) connected by arrows to a central green square containing a white Euro symbol (€). Below this graphic, the text 'Pay My Buddy' is written in large, bold, green letters. Underneath, the tagline 'We make moving your money, Easy!!!' is displayed in smaller blue text. The main content area is a light gray box containing a form. The form has a label 'Adresse e-mail de l'ami:' followed by a text input field containing the email 'badr@gmail.com'. To the right of the input field is a blue button with the text 'Ajouter Ami'. At the bottom left of the page, there is a blue link that says 'Retour à la page d'accueil'.

localhost:8080/ajouterAmi



Pay My Buddy

We make moving your money, Easy!!!


Adresse e-mail de l'ami: Ajouter Ami

[Retour à la page d'accueil](#)

Un utilisateur peut ajouter des personnes à sa liste à partir de leur adresse e-mail



← ↻ ⓘ localhost:8080/ajouterAmi



Pay My Buddy

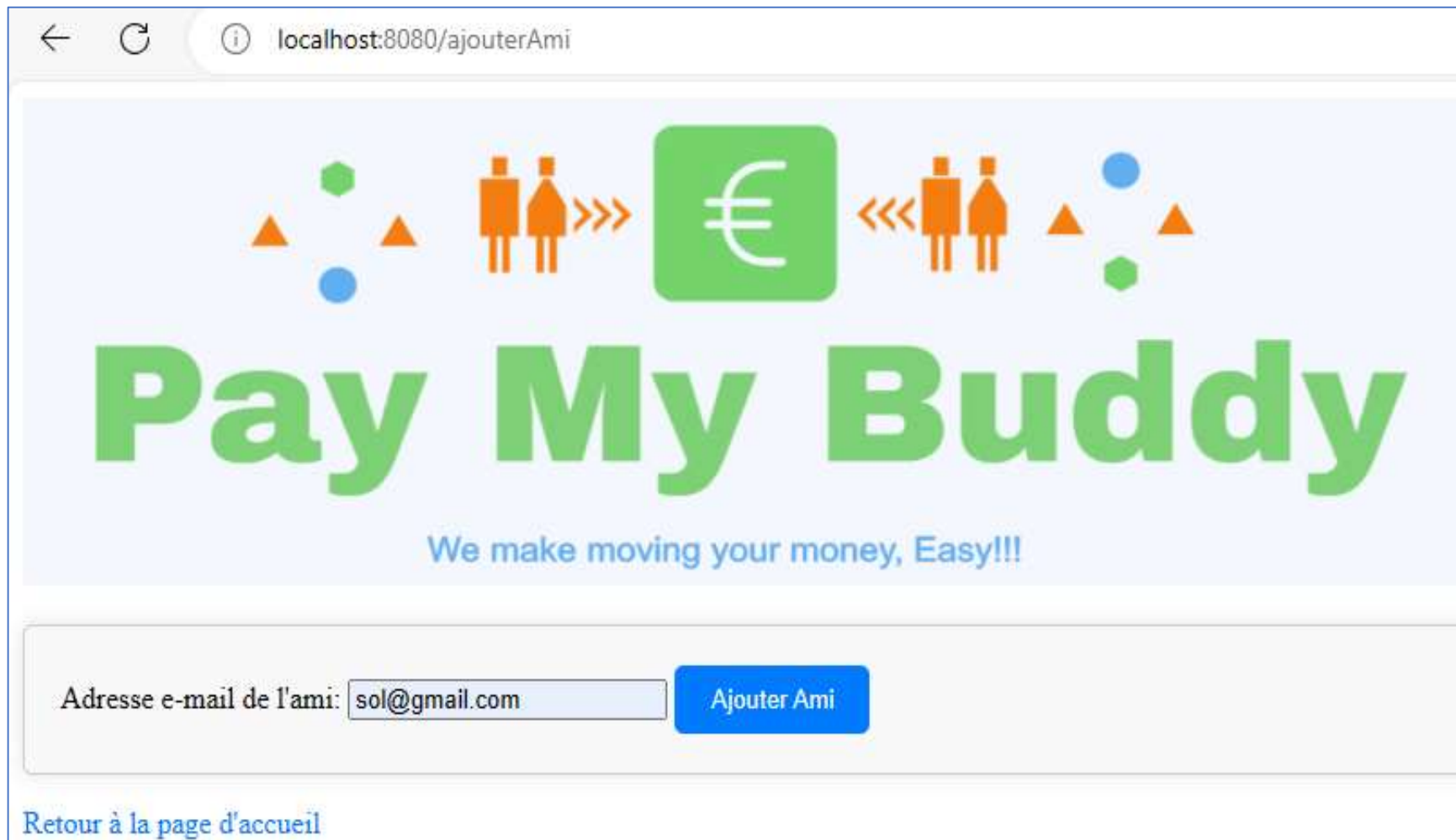
We make moving your money, Easy!!!

Cet utilisateur est déjà dans votre liste d'amis.

Adresse e-mail de l'ami: Ajouter Ami


[Retour à la page d'accueil](#)

Un utilisateur peut ajouter des personnes à sa liste à partir de leur adresse e-mail



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/ajouterAmi'. The page has a light blue header with a decorative graphic of two groups of people (represented by orange stick figures) connected by arrows, with a green square containing a white Euro symbol (€) in the center. Below the graphic, the text 'Pay My Buddy' is written in large, bold, green letters. Underneath, the tagline 'We make moving your money, Easy!!!' is displayed in smaller blue text. The main content area is a light gray box containing a form. The form has a label 'Adresse e-mail de l'ami:' followed by a text input field containing 'sol@gmail.com'. To the right of the input field is a blue button with the text 'Ajouter Ami'. At the bottom left of the page, there is a blue link that says 'Retour à la page d'accueil'.

← ↻ ⓘ localhost:8080/ajouterAmi



Pay My Buddy

We make moving your money, Easy!!!

Adresse e-mail de l'ami:

[Retour à la page d'accueil](#)

Un utilisateur peut ajouter des personnes à sa liste à partir de leur adresse e-mail



localhost:8080/ajouterAmi



Pay My Buddy

We make moving your money, Easy!!!

Aucun utilisateur trouvé avec cette adresse e-mail.

Adresse e-mail de l'ami:

[Ajouter Ami](#)

[Retour à la page d'accueil](#)

Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/effectuerVirement'. The page features a header with a logo consisting of two stylized figures, a green square with a white Euro symbol, and another set of stylized figures, all flanked by colorful geometric shapes. Below the logo, the text 'Pay My Buddy' is written in a large, green, sans-serif font, followed by the tagline 'We make moving your money, Easy!!!' in a smaller, blue font.

The main content area is titled 'Effectuer un virement' in bold black text. It contains a form with three input fields: 'Sélectionnez un ami :', 'Montant du virement :', and 'Description du virement :'. The 'Sélectionnez un ami :' field is a dropdown menu with a red border, showing a list of friends: 'Liste des amis', 'Liste des amis', 'badr@gmail.com', and 'ines@gmail.com'. The 'Montant du virement :' field is a text input box. The 'Description du virement :' field is a text area. Below the form is a blue button labeled 'Effectuer le virement'.

At the bottom of the page, there is a link labeled 'Retour à la page d'accueil'.

Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application

Effectuer un virement


Sélectionnez un ami :

Montant du virement : 

Description du virement :

[Effectuer le virement](#)

[Retour à la page d'accueil](#)

ID	prenom	nom	adresseemail	motdepasse	soldeducompte
27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	5000.00
29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	6000.00 
30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application

Effectuer un virement

Solde insuffisant pour effectuer le virement.

Sélectionnez un ami :

Liste des amis ▼

Montant du virement :

Description du virement :


Effectuer le virement

[Retour à la page d'accueil](#)

Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application

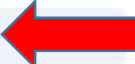
Effectuer un virement

Sélectionnez un ami :

Montant du virement : 

Description du virement :

[Retour à la page d'accueil](#)

ID	prenom	nom	adresseemail	motdepasse	soldeducompte
27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	5000.00
29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	6000.00 
30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application

Effectuer un virement

Virement effectué avec succès.

Sélectionnez un ami :

Montant du virement :

Description du virement :

Effectuer le virement

[Retour à la page d'accueil](#)

Prélèvement de 0.5% pour monétiser l'application

ID	prenom	nom	adresseemail	motdepasse	soldeducompte
27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	5000.00
29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	6000.00
30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
▶	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	6990.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	4000.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application

Effectuer un virement

Virement effectué avec succès.

Sélectionnez un ami :

Montant du virement :

Description du virement :

Effectuer le virement

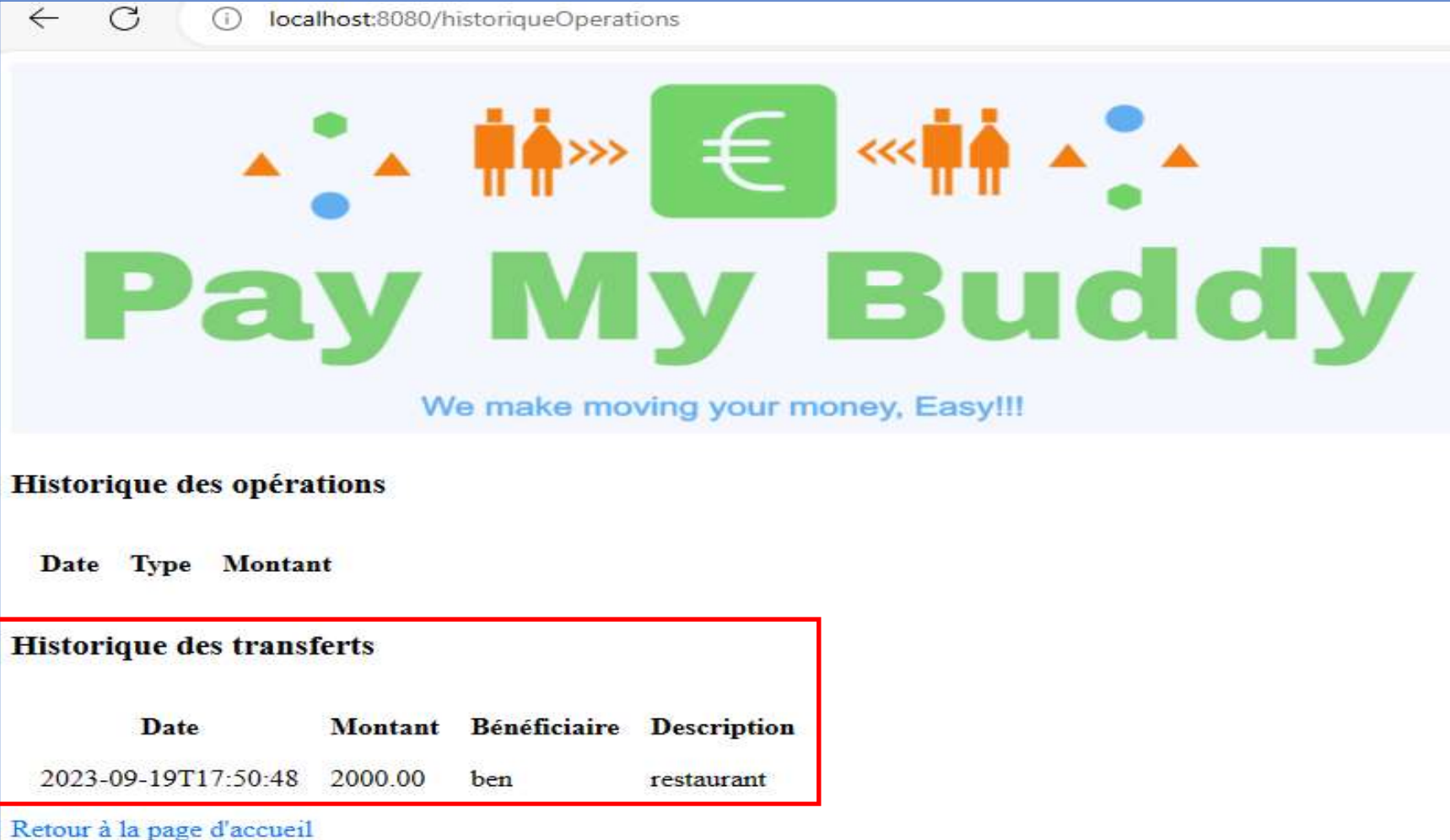
[Retour à la page d'accueil](#)

Prélèvement de 0.5% pour monétiser l'application

	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
►	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	6990.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	4000.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

ID	IDutilisateurremetteur	IDutilisateurbeneficiaire	Montant	Dateetheuredutransfert	Description
5	29	27	2000.00	2023-09-19 15:50:48	restaurant

Un utilisateur peut effectuer des paiements à ses amis enregistrés sur l'application



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/historiqueOperations'. The application header features a logo with two stylized figures and a large green Euro symbol, followed by the text 'Pay My Buddy' and the tagline 'We make moving your money, Easy!!!'. Below the header, there is a section titled 'Historique des opérations' with a table containing columns for 'Date', 'Type', and 'Montant'. A red rectangular box highlights a sub-section titled 'Historique des transferts', which contains a table with columns for 'Date', 'Montant', 'Bénéficiaire', and 'Description'. The table lists a single transaction: a payment of 2000.00 to 'ben' at a 'restaurant' on 2023-09-19T17:50:48. At the bottom of the page, there is a link labeled 'Retour à la page d'accueil'.

Historique des opérations

Date	Type	Montant
------	------	---------

Historique des transferts

Date	Montant	Bénéficiaire	Description
2023-09-19T17:50:48	2000.00	ben	restaurant

[Retour à la page d'accueil](#)

Un utilisateur peut verser de l'argent sur son compte dans notre application

Effectuer un dépôt

Montant du dépôt : [Effectuer le dépôt](#)

[Retour à la page d'accueil](#)

Effectuer un dépôt

Dépôt effectué avec succès.

Montant du dépôt : [Effectuer le dépôt](#)

[Retour à la page d'accueil](#)

	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
▶	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	6990.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	4000.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
▶	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	6990.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	7980.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

ID	IDutilisateur	Montant	Dateetheureoperation	Natureoperation
20	29	3980.00	2023-09-19 16:05:20	depot

Prélèvement de 0.5% pour monétiser l'application

Un utilisateur peut transférer l'argent vers son compte bancaire

	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
►	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	6990.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	7980.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

Effectuer un retrait

Montant du retrait : 8000

Effectuer le retrait

[Retour à la page d'accueil](#)

Effectuer un retrait

Montant du retrait :

Effectuer le retrait

Solde insuffisant pour effectuer le retrait.

[Retour à la page d'accueil](#)

Un utilisateur peut transférer l'argent vers son compte bancaire

	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
▶	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	6990.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	7980.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

Effectuer un retrait

Montant du retrait :

[Effectuer le retrait](#)

[Retour à la page d'accueil](#)

Effectuer un retrait

Montant du retrait :

[Effectuer le retrait](#)

Retrait effectué avec succès.

[Retour à la page d'accueil](#)

	ID	prenom	nom	adresseemail	motdepasse	soldeducompte
	27	badr	ben	badr@gmail.com	\$2a\$10\$VOVeV.KLYC1Qk/iGJhBKiuNarC6SSkZ21QaK...	6990.00
	29	sirine	ben	sirine@gmail.com	\$2a\$10\$O9vxHyvW94WYPT4DDFUmCuyfPssEfy3to...	2010.00
	30	ines	ben	ines@gmail.com	\$2a\$10\$N65gr3GkXtK1YkzKCO0tYOALjPUKrnkQmBn...	7000.00

	ID	IDutilisateur	Montant	Dateetheureoperation	Natureoperation
	20	29	3980.00	2023-09-19 16:05:20	depot
	21	29	5970.00	2023-09-19 16:11:28	retrait

Prélèvement de 0.5% pour monétiser l'application

Un utilisateur peut consulter l'historique de ses transactions

Historique des opérations

Date	Type	Montant
2023-09-19T18:05:20	depot	3980.00
2023-09-19T18:11:28	retrait	5970.00

Historique des transferts

Date	Montant	Bénéficiaire	Description
2023-09-19T17:50:48	2000.00	ben	restaurant