

2021

Gestion d'un compte bancaire



Hafsa Nouri

MBD

1/17/2021

État de l'art et Modéliser

Conception :

Le montant totale : la somme de toutes les transactions effectuées, chaque fois que le client clique sur le bouton « Submit » on ajoute ce montant au montant total qui est initialement égale à zéro.

Crédit : en cliquant sur le bouton « + » un montant de 50 dh s'ajoute dans le champ du montant à ajouter

Débit : en cliquant sur le bouton « - » un montant de -50 dh se diminue dans le champ du montant à ajouter

Cas de 0 dh : en cas de 0 un message d'alerte s'affiche en bas de l'application « spécifier un montant s'il vous plaît »

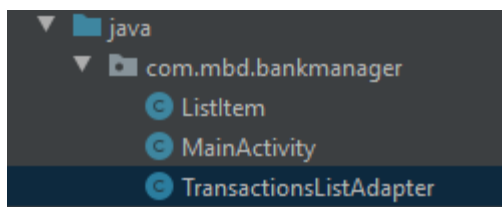
Historique des transactions : une liste des transactions se crée une fois que vous cliquez sur le bouton « Submit » et après la 1^{er} transaction vous aurez la plus récente en top de la liste, pour chaque transaction on aura des informations relatives que vous trouverez ci-dessous

- Montant effectué : la valeur si positive c'est un crédit si c'est négatif c'est un débit.
- Date d'affectation : le jour et l'heure d'où la transaction a été effectuée.
- Type de transaction : Crédit / débit.

Réalisation :

Le projet contient 3 classes Java et 3 layouts XML.

Les classes Java :



MainActivity.java : la class principale ou en fait les calculs du montant totale et on soumit les valeurs des transactions.

```
1 package com.mbd.bankmanager;
2
3 import ...
4
23
24 public class MainActivity extends AppCompatActivity {
25     int minteger = 0;
26     String transactionType;
27     ArrayList<ListItem> transaction = new ArrayList<>();
28     int fullSolde = 0;
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {...}
93     public ListItem getListData(int amount, String date, String transactionType) {...}
104
105     public String date() {...}
111     public String TransactionType(int amount) {...}
119     public void increase(View view) {...} public void decrease(View view) {...}
128
129     private void display(int number) {...}
133
134 }
```

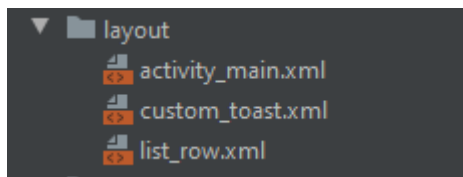
ListItem.java : cette classe contient les éléments de la liste (amonts, date, transaction) avec les getters et les setters.

```
1 package com.mbd.bankmanager;
2
3 public class ListItem {
4     private String amount;
5     private String date;
6     private String transaction;
7     public String getAmount() { return amount; }
10    public void setAmount(String amount) { this.amount = amount; }
13    public String getDate() { return date; }
16    public void setDate(String date) { this.date = date; }
19    public String getTransaction() { return transaction; }
22    public void setTransaction(String location) { this.transaction = location; }
25 }
26
```

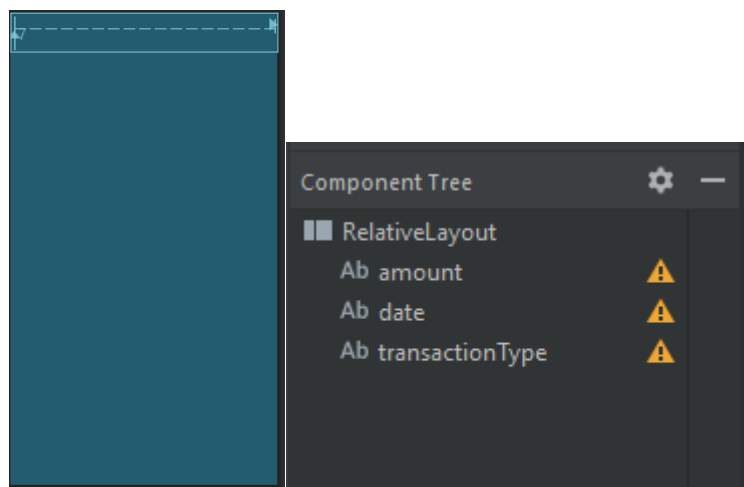
TransactionsListAdapter.java : la classe qui s'occupe d'actualiser au même temps de l'exécution la liste chaque fois l'utilisateurs ajoutes une nouvelle transaction.

```
1 package com.mbd.bankmanager;
2 import ...
10
11 public class TransactionsListAdapter extends BaseAdapter {
12     private ArrayList<ListItem> listData;
13     private LayoutInflater layoutInflater;
14     public TransactionsListAdapter(Context aContext, ArrayList<ListItem> listData) {...}
18
19 @ public TransactionsListAdapter(Context context, int textViewResourceId, ArrayList userList) {...}
23
24
25 @Override
26 public int getCount() { return listData.size(); }
29 @Override
30 public Object getItem(int position) { return listData.get(position); }
33 @Override
34 public long getItemId(int position) { return position; }
37 public View getView(int position, View v, ViewGroup vg) {...}
56 static class ViewHolder {...}
61
```

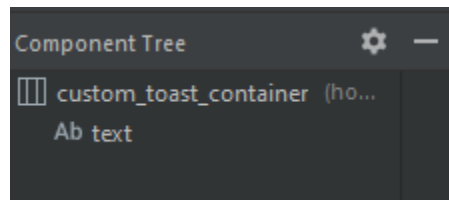
Les layout XML :



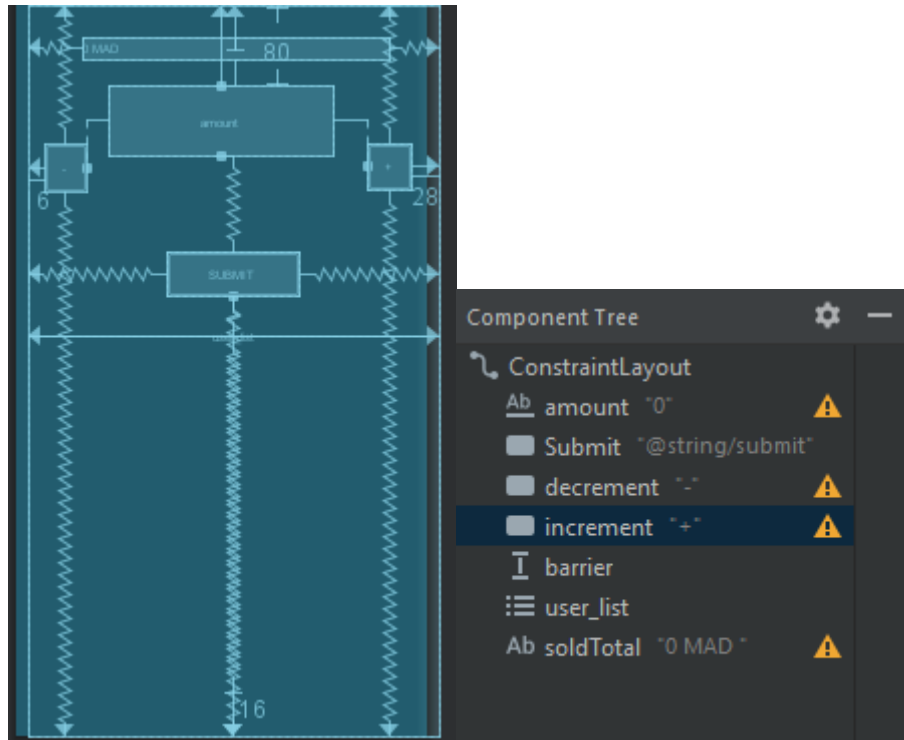
list_row.xml : cette partie contient les lignes de la liste selon les 3 variables qu'on veut afficher



custom_toast.xml : il contient de design du toast personnalisé

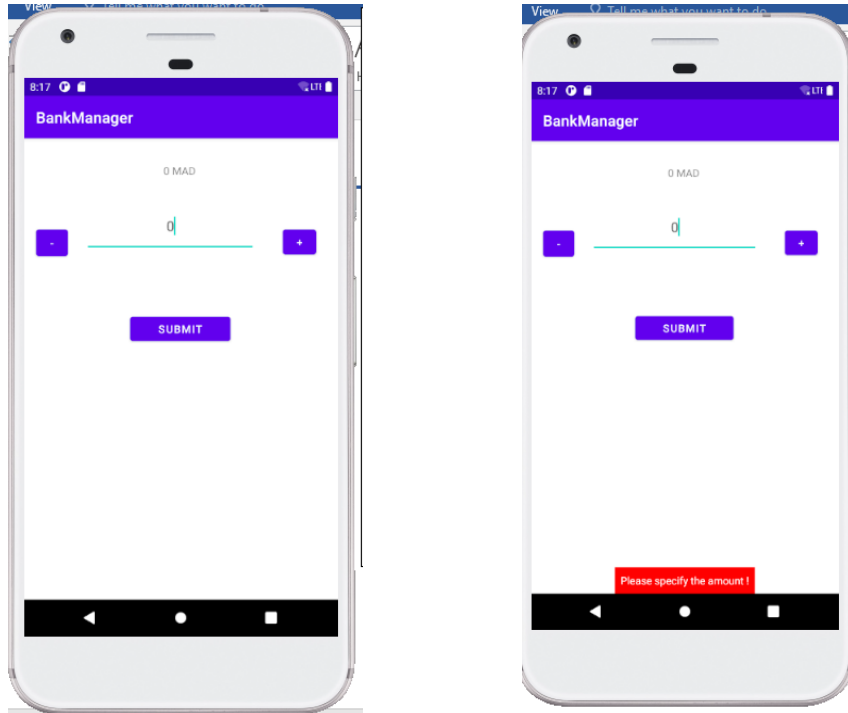


activity_main.xml : contient tous les champs et les boutons dans la fenêtre principale

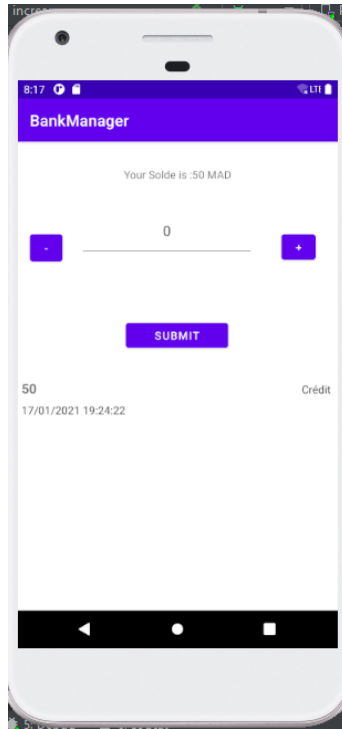


Produit finale

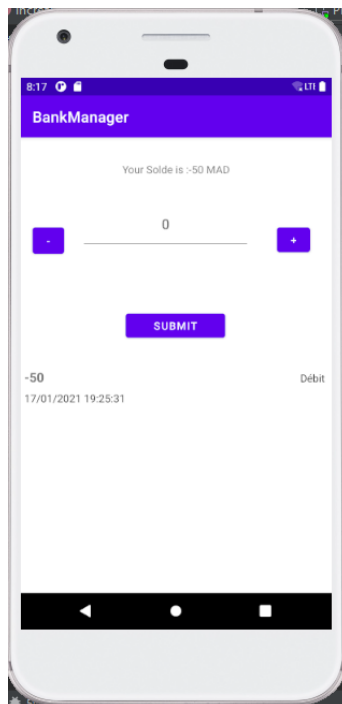
Démo du cas de zéro :



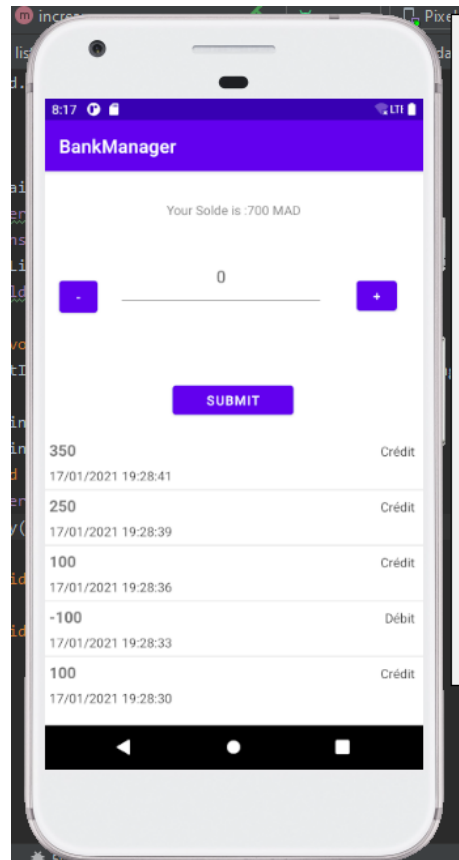
Démo d'un crédit :



Démo cas de débit :



Démo cas d'historique des transactions :



Montant totale :

