

Sommaire

Introduction	3
I- Travail à faire :	4
II- Conception	4
II- Architecture Rest API	5
III- Partie Backend	5
1- Initialisation du projet sur IntelliJ	5
2- la structure de l'application	6
3- Démonstration de l'API	8
III- Partie du Frontend avec Angular	11
1- Structure d'un projet Angular :	11
2- Structure du projet Digital Banking :	11
IV- Partie Mobile avec Flutter	17
1- Flutter ?	17
2- Structure de projet Digital Bank sous Flutter ?	17
2- Démonstration de l'application	18
Conclusion.....	24

Introduction

Dans le cadre de notre formation de Spring et Angular chez Monsieur Mohamed Youssfi, nous sommes amenés à réaliser une application web rendu coté client basé sur Spring et Angular afin d'appliquer nos connaissances théoriques et pratiquer les notions étudiées durant ce cours.

Cette application permet de gérer les comptes bancaires. Chaque compte appartient à un client il existe deux types de comptes : Courant et Epargnes. Chaque Compte peut subir des opérations de types Débit ou crédit.

L'application se compose des couches suivantes :

- Couche DAO (Entités JPA et Repositories)
- Couche Service définissant les opérations suivantes :
Ajouter des comptes, ajouter des clients, effectuer un débit (Retrait), effectuer un crédit (Versement), effectuer un virement, consulter un compte
- La couche DTO
- Mappers (DTO <=>Entities)
- La couche Web (Rest Controllers)
- Couche sécurité (Spring Security avec JWT)

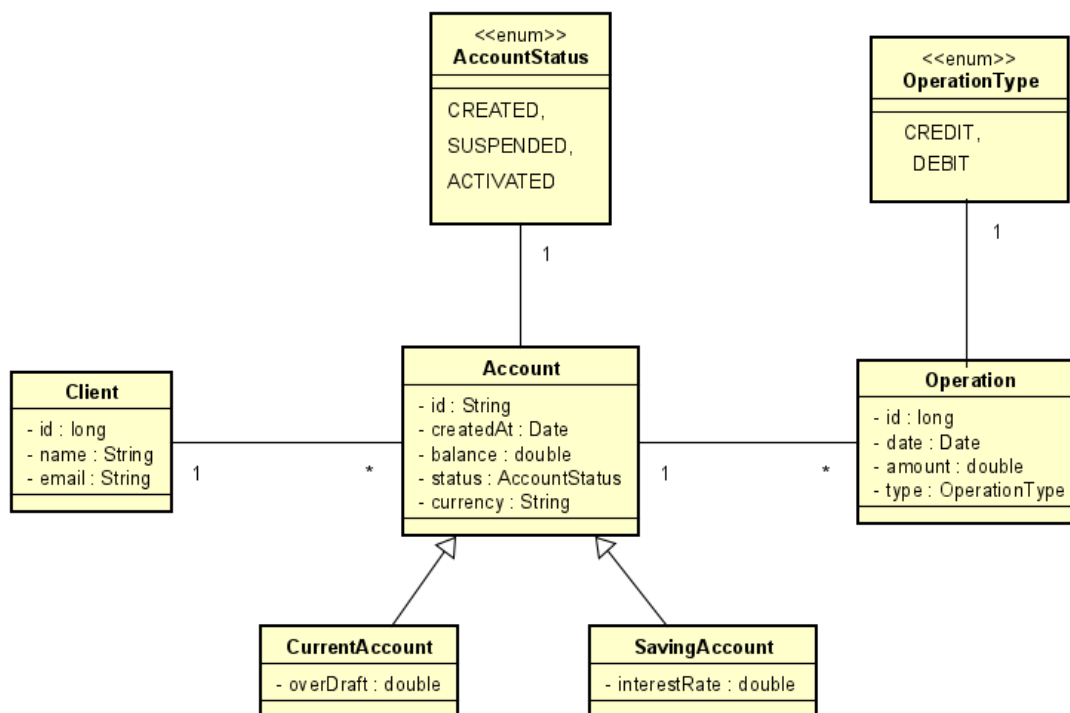
Dans ce rapport je présente le travail que j'ai réalisé afin de réussir ce projet, en montrant la conception, les étapes de développements et la démonstration à partir des captures d'écran.

I- Travail à faire :

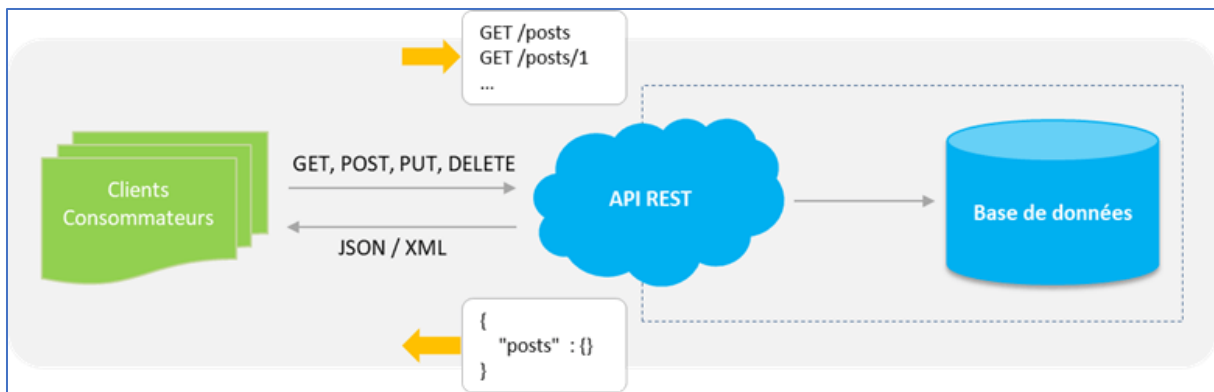
- Créer et tester la couche DAO
- Créer et tester la couche service
- Créer et tester la couche Web (Rest Controller)
- Modifier la couche service et la couche web en utilisant les DTO
- Créer un service d'authentification séparé basé sur Spring Security et JWT
- Sécuriser l'application Digital Banking en utilisant Spring Security et JWT
- Créer la partie Frontend Web en utilisant Angular
- Créer la partie Frontend Mobile avec Flutter

II- Conception

Diagramme de classe : Digital Banking

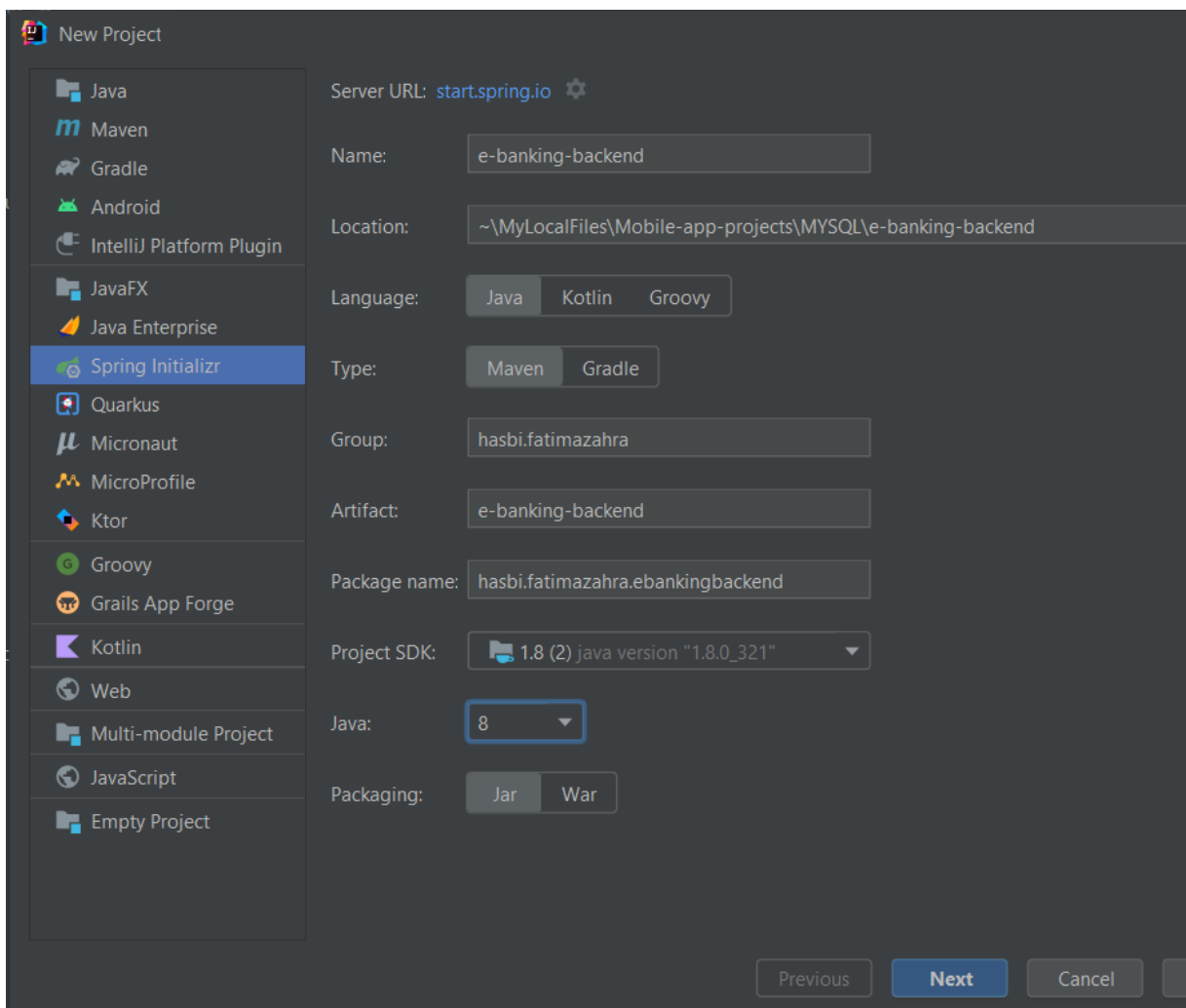


II- Architecture Rest API

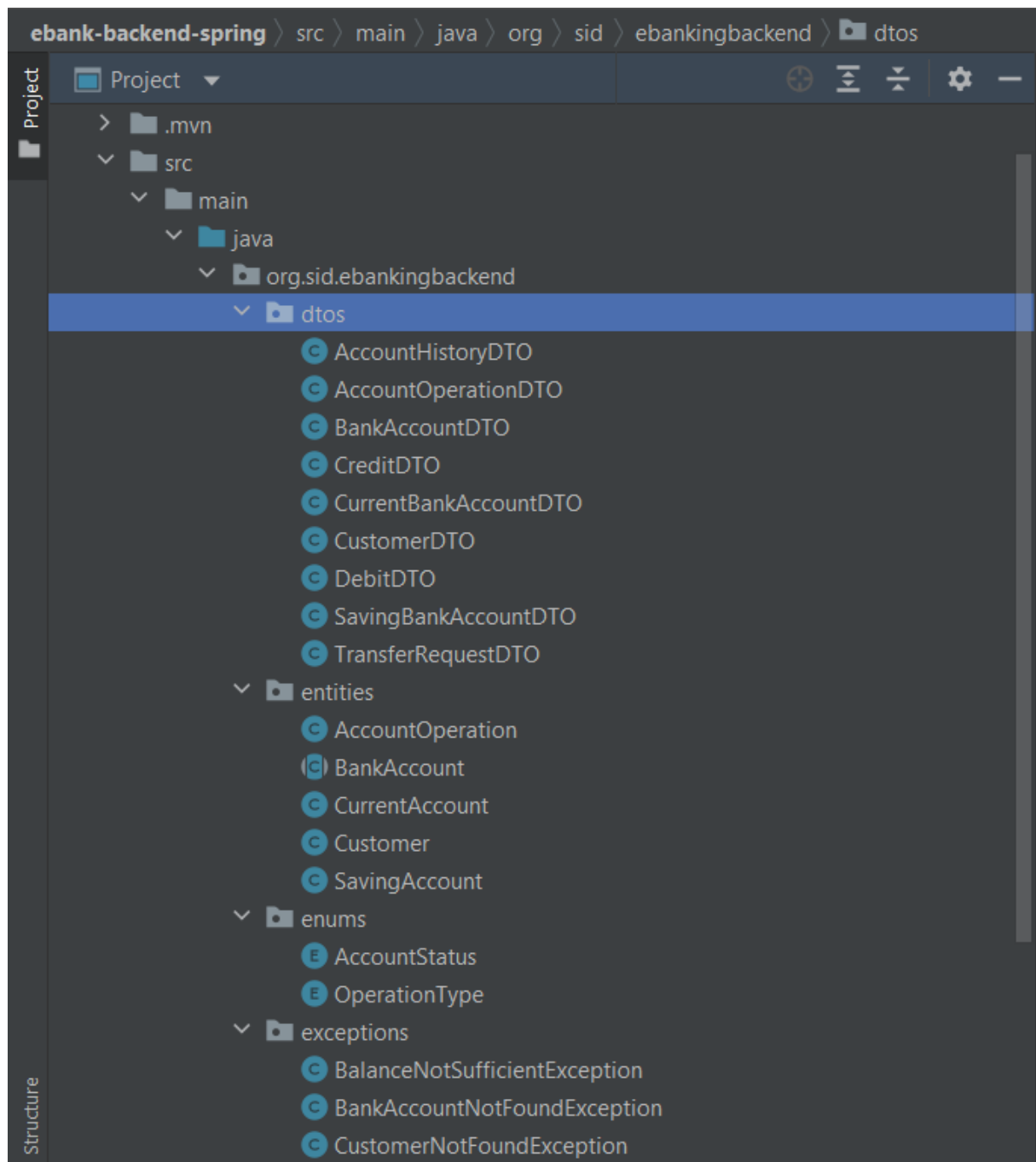


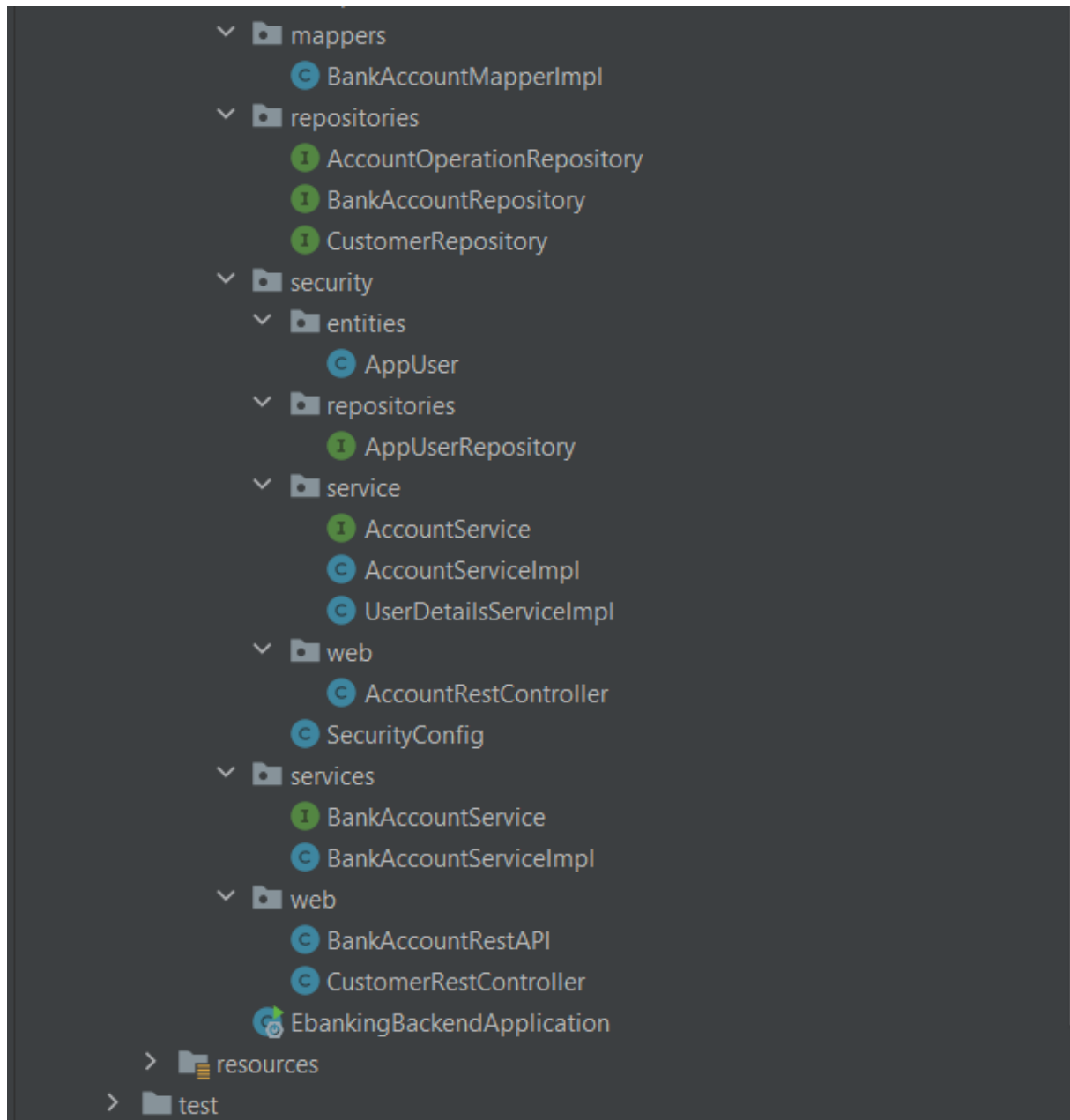
III- Partie Backend

1- Initialisation du projet sur IntelliJ



2- la structure de l'application





Le lien du code source du backend et de frontend avec Angular de l'application sur GitHub :

<https://github.com/FatimaZahraHASBI/Projet-Digital-Banking-Spring-Angular.git>

3- Démonstration de l'API

```
localhost:8085

1 // 20220529005754
2 // http://localhost:8085/
3
4 [
5   {
6     "id": 1,
7     "username": "Fatima Zahra HASBI",
8     "password": "$2a$10$EfQ39nEZWkT/.wRMXj60HuQMLCMKBrnXNsBYJ5P79s.VvxTyv3Mc2",
9     "role": "ADMIN"
10  }
11 ]
```

```
DigitalBankingWeb | FatimaZahraHASBI/Projet | localhost:8085/customers | +
localhost:8085/customers

1 // 20220529005830
2 // http://localhost:8085/customers
3
4 [
5   {
6     "id": 1,
7     "name": "FatimaZahra",
8     "email": "FatimaZahra@gmail.com"
9   },
10  {
11    "id": 2,
12    "name": "Hasnaa",
13    "email": "Hasnaa@gmail.com"
14  },
15  {
16    "id": 3,
17    "name": "Achraf",
18    "email": "Achraf@gmail.com"
19  }
20 ]
```



```
localhost:8085/accounts

1 // 20220529005859
2 // http://localhost:8085/accounts
3
4 [
5   {
6     "type": "CurrentAccount",
7     "id": "00e25fa5-0c11-47f5-895e-00d8a5e4ba3e",
8     "balance": 852374.9128190153,
9     "createdAt": "2022-05-28T23:41:54.000+00:00",
10    "status": null,
11    "customerDTO": {
12      "id": 2,
13      "name": "Hasnaa",
14      "email": "Hasnaa@gmail.com"
15    },
16    "overDraft": 9000.0
17  },
18  {
19    "type": "SavingAccount",
20    "id": "30a6a710-861a-47eb-8f44-33765666293e",
21    "balance": 676644.4959706266,
22    "createdAt": "2022-05-28T23:41:54.000+00:00",
23    "status": null,
24    "customerDTO": {
25      "id": 2,
26      "name": "Hasnaa",
27      "email": "Hasnaa@gmail.com"
28    },
29    "interestRate": 5.5
30  },
31  {
```

```
localhost:8085/accounts/00e25fa5-0c11-47f5-895e-00d8a5e4ba3e

1 // 20220529010008
2 // http://localhost:8085/accounts/00e25fa5-0c11-47f5-895e-00d8a5e4ba3e
3
4 {
5   "type": "CurrentAccount",
6   "id": "00e25fa5-0c11-47f5-895e-00d8a5e4ba3e",
7   "balance": 852374.9128190153,
8   "createdAt": "2022-05-28T23:41:54.000+00:00",
9   "status": null,
10  "customerDTO": {
11    "id": 2,
12    "name": "Hasnaa",
13    "email": "Hasnaa@gmail.com"
14  },
15  "overDraft": 9000.0
16 }
```

```
localhost:8085/accounts/00e25fa5-0c11-47f5-895e-00d8a5e4ba3e/operations

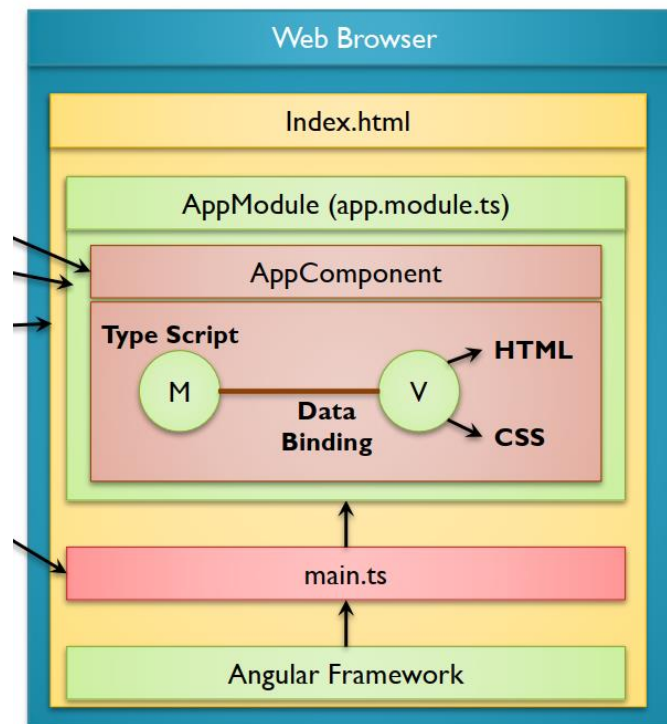
1 // 20220529010116
2 // http://localhost:8085/accounts/00e25fa5-0c11-47f5-895e-00d8a5e4ba3e/operations
3
4 [
5   {
6     "id": 1,
7     "operationDate": "2022-05-28T23:41:54.000+00:00",
8     "amount": 17999.65280288101,
9     "type": "CREDIT",
10    "description": "Credit"
11  },
12  {
13    "id": 2,
14    "operationDate": "2022-05-28T23:41:54.000+00:00",
15    "amount": 3238.0021663189614,
16    "type": "DEBIT",
17    "description": "Debit"
18  },
19  {
20    "id": 3,
21    "operationDate": "2022-05-28T23:41:54.000+00:00",
22    "amount": 95610.69753039457,
23    "type": "CREDIT",
24    "description": "Credit"
25  },
26  {
27    "id": 4,
28    "operationDate": "2022-05-28T23:41:54.000+00:00",
29    "amount": 4100.794090135647,
30    "type": "DEBIT",
31    "description": "Debit"
32  }
33 ]
```

```
localhost:8085/customers/1/accounts

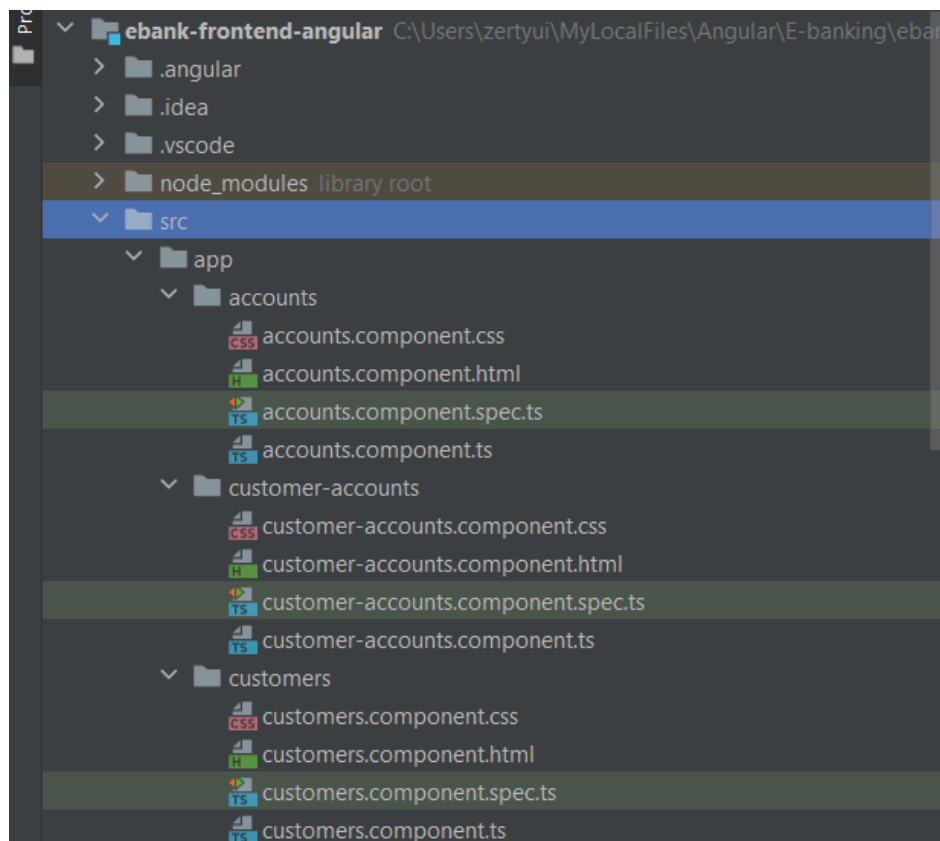
1 // 20220529010151
2 // http://localhost:8085/customers/1/accounts
3
4 [
5   {
6     "type": "SavingAccount",
7     "id": "5c27ee78-3031-4955-b806-8e520dee13cb",
8     "balance": 766619.774658905,
9     "createdAt": "2022-05-28T23:41:54.000+00:00",
10    "status": null,
11    "customerDTO": {
12      "id": 1,
13      "name": "FatimaZahra",
14      "email": "FatimaZahra@gmail.com"
15    },
16    "interestRate": 5.5
17  },
18  {
19    "type": "CurrentAccount",
20    "id": "5ff2d0c1-04d0-4e25-840d-63f9fc4836d7",
21    "balance": 958873.3212285051,
22    "createdAt": "2022-05-28T23:41:53.000+00:00",
23    "status": null,
24    "customerDTO": {
25      "id": 1,
26      "name": "FatimaZahra",
27      "email": "FatimaZahra@gmail.com"
28    }
29  }
30 ]
```

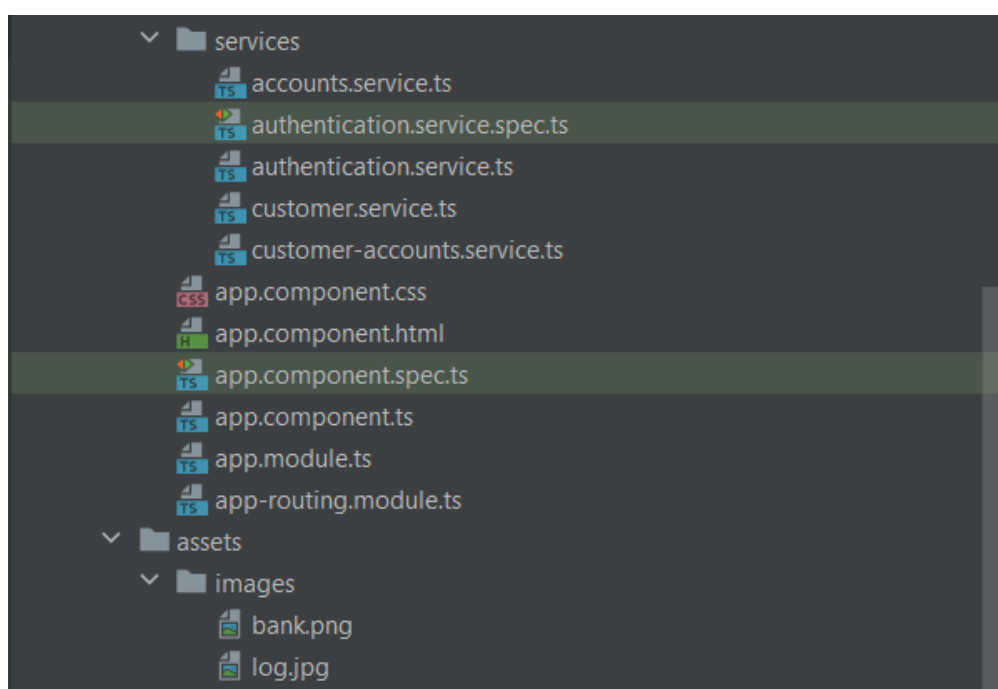
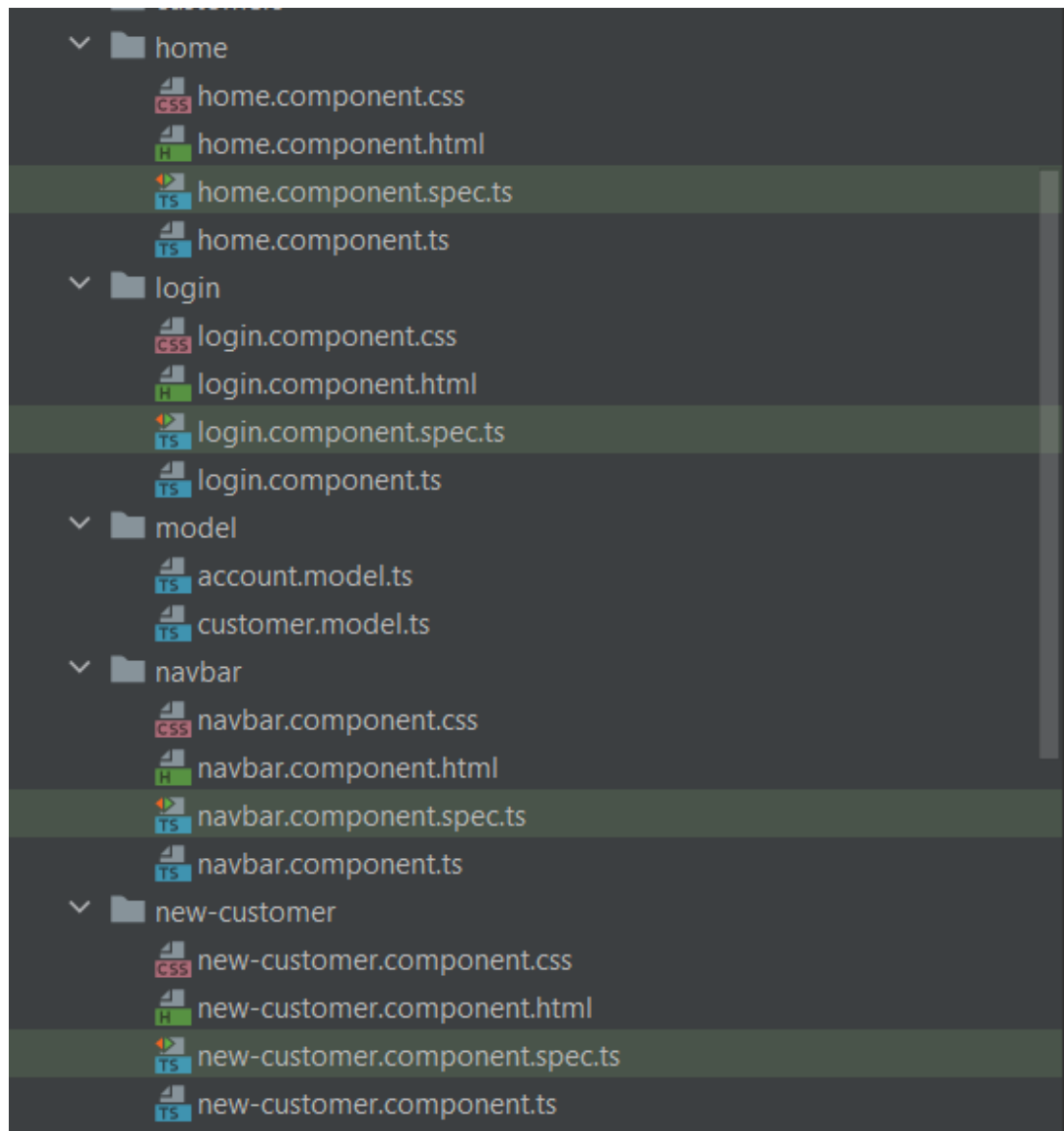
III- Partie du Frontend avec Angular

1- Structure d'un projet Angular :

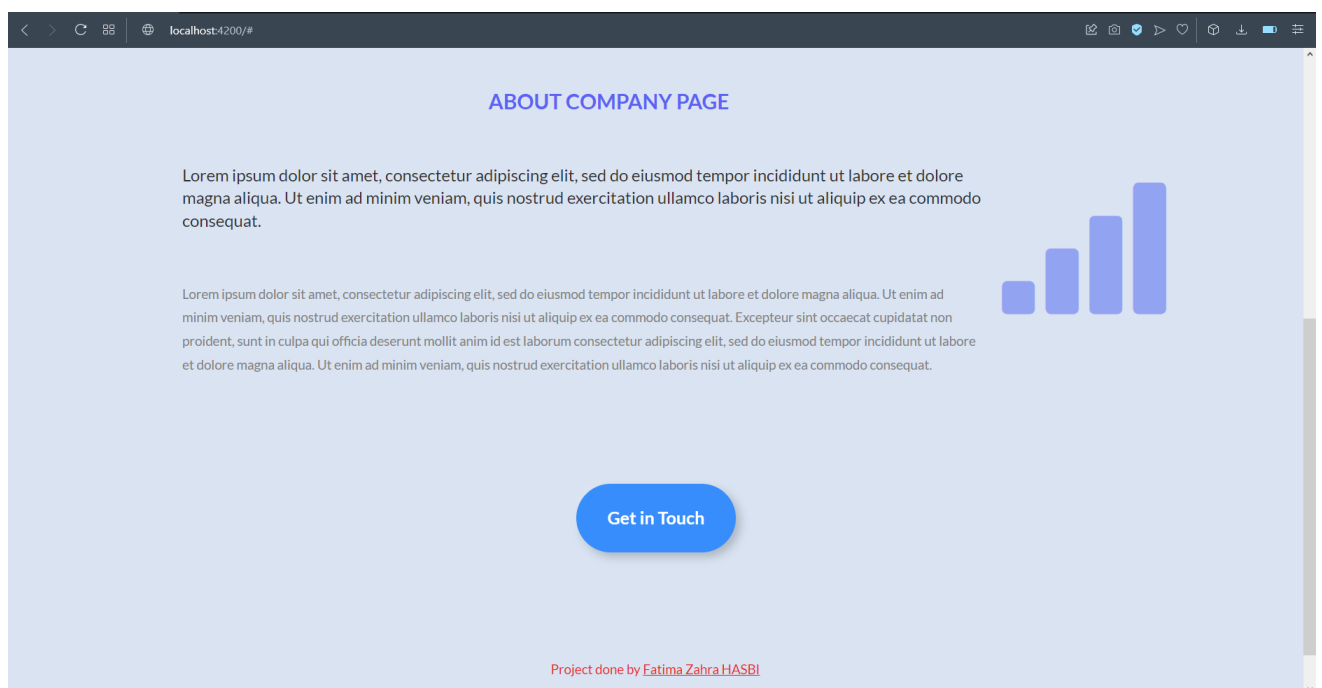
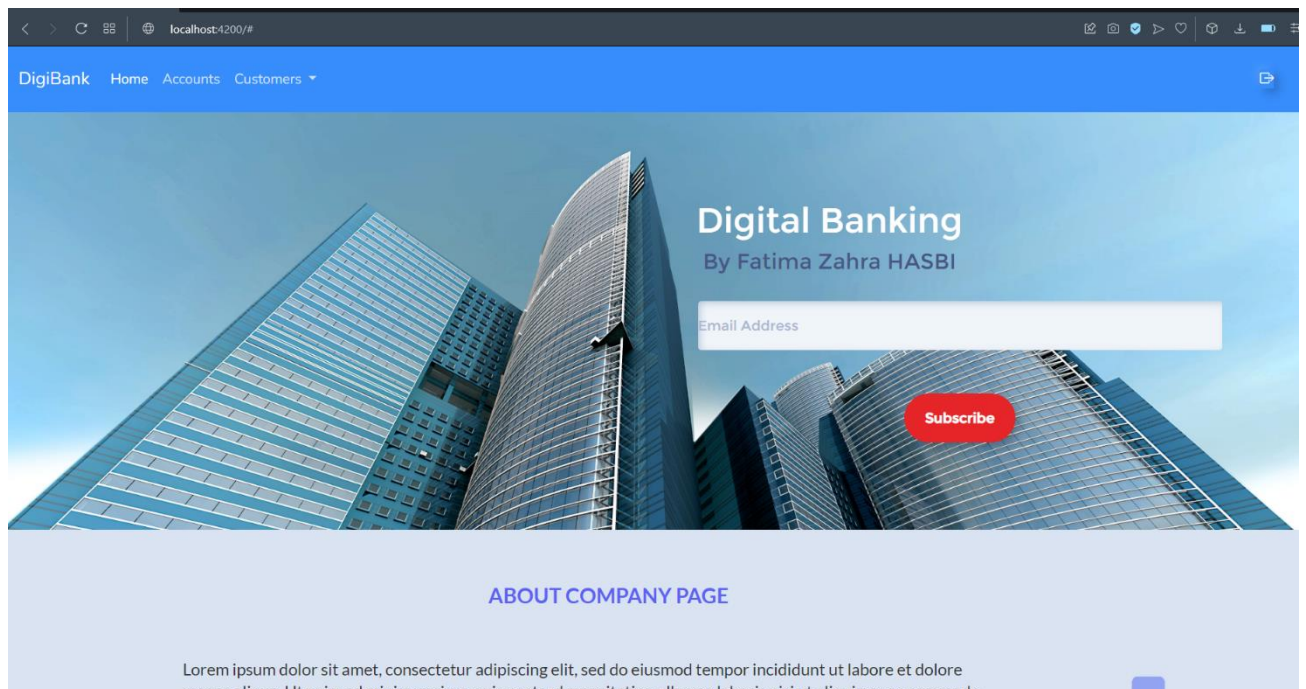


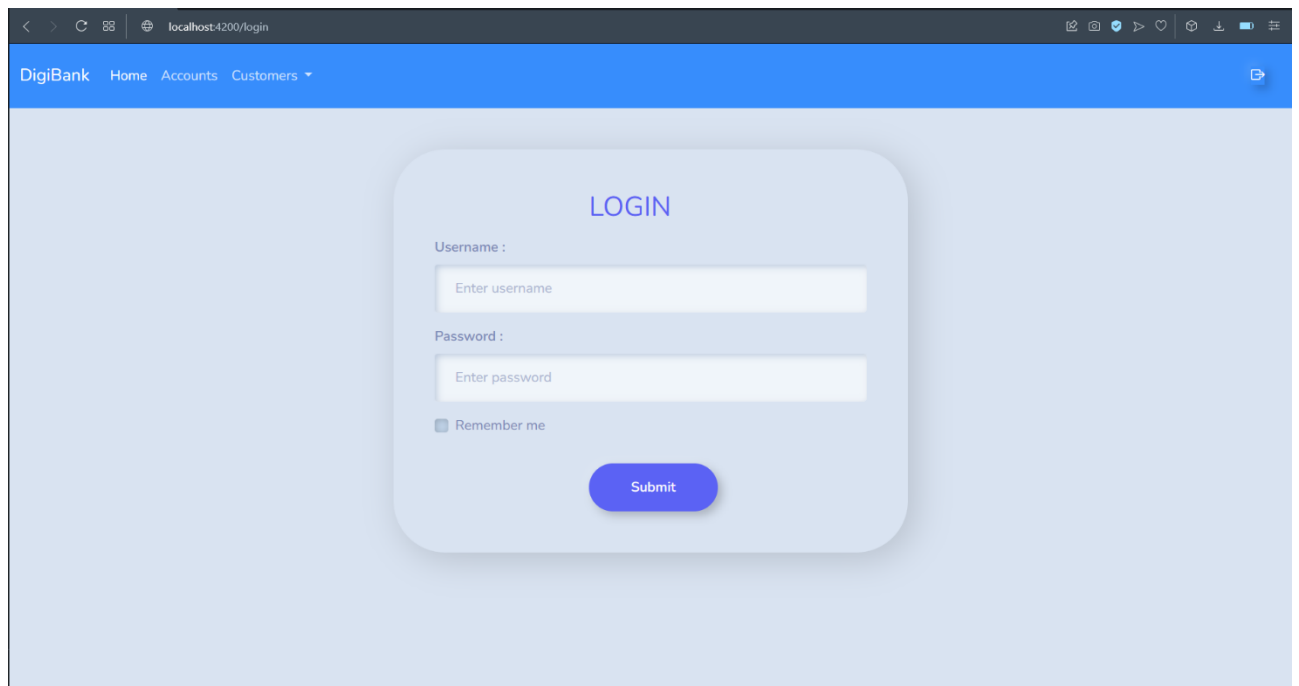
2- Structure du projet Digital Banking :





3- Démonstration de l'application web :





localhost:4200/login

DigiBank Home Accounts Customers

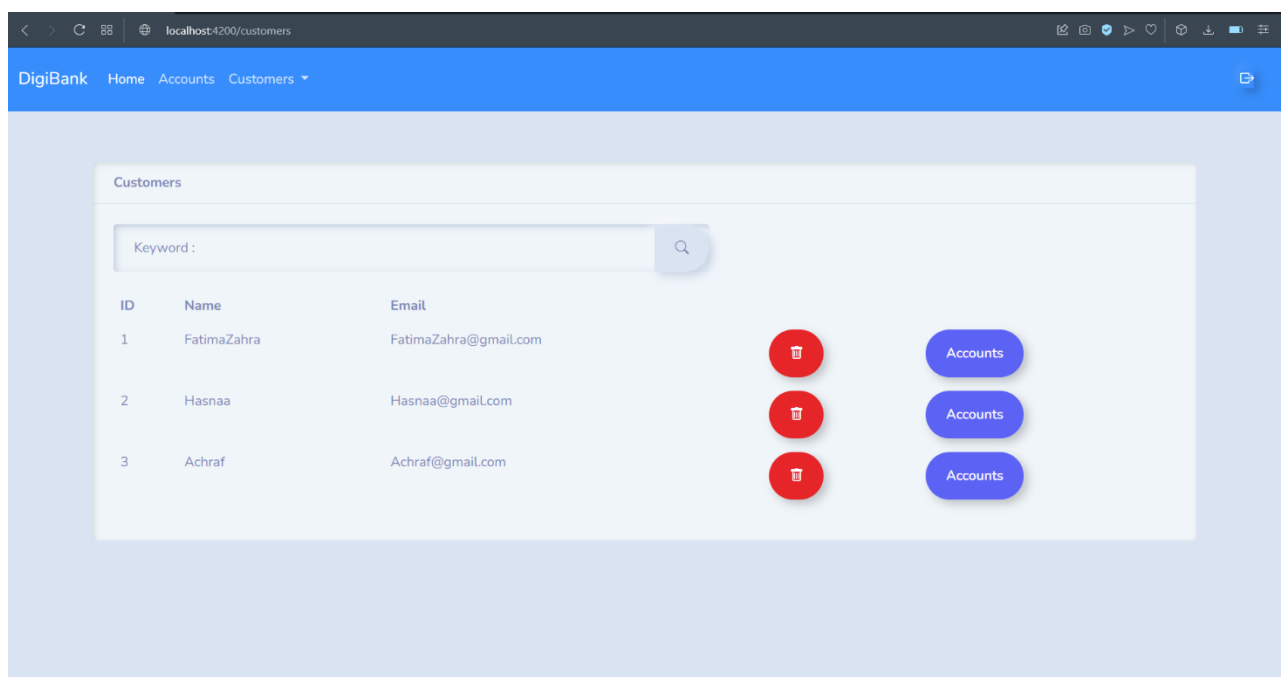
LOGIN

Username :

Password :

☐ Remember me

Submit



localhost:4200/customers

DigiBank Home Accounts Customers

Customers

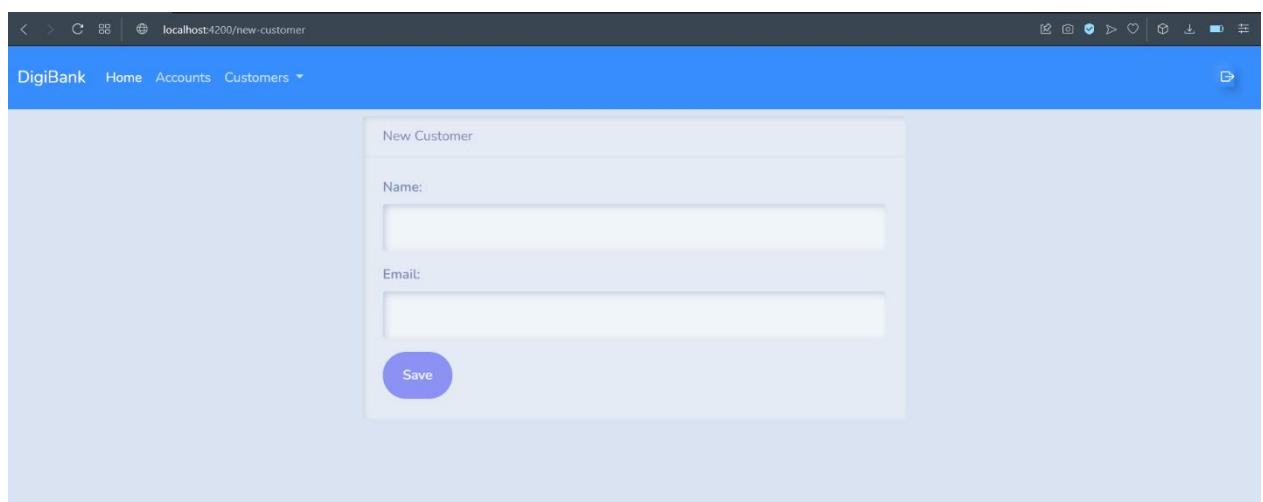
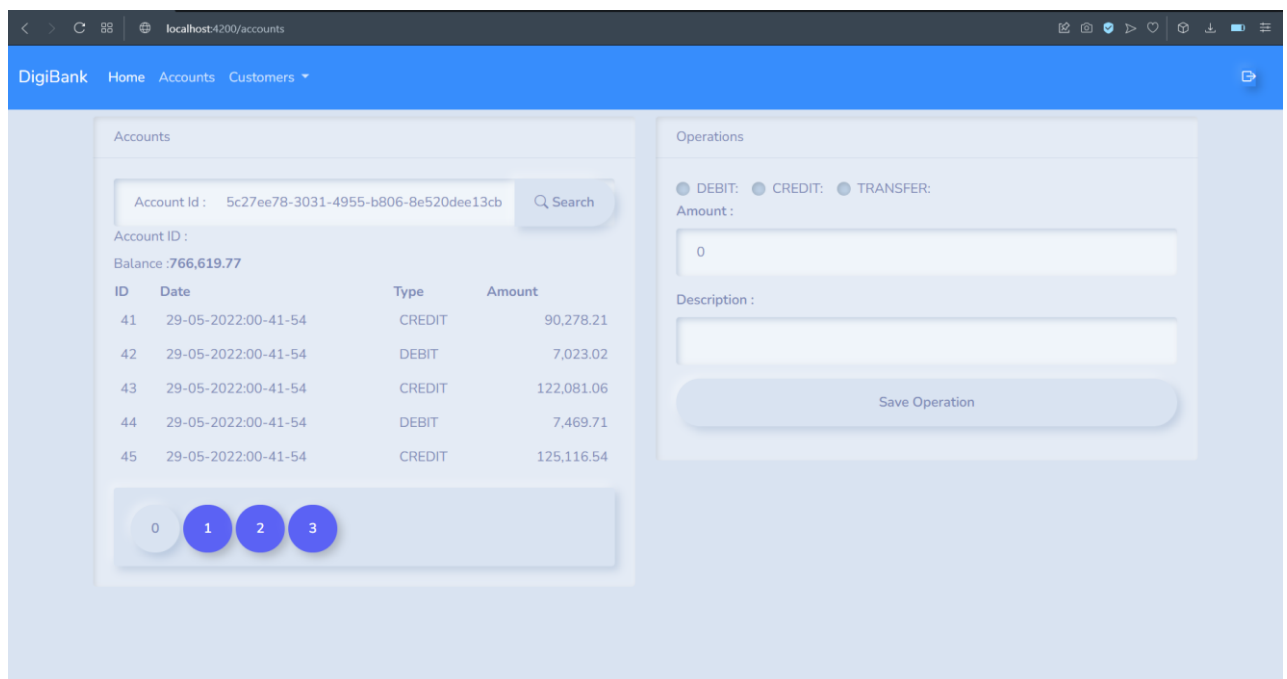
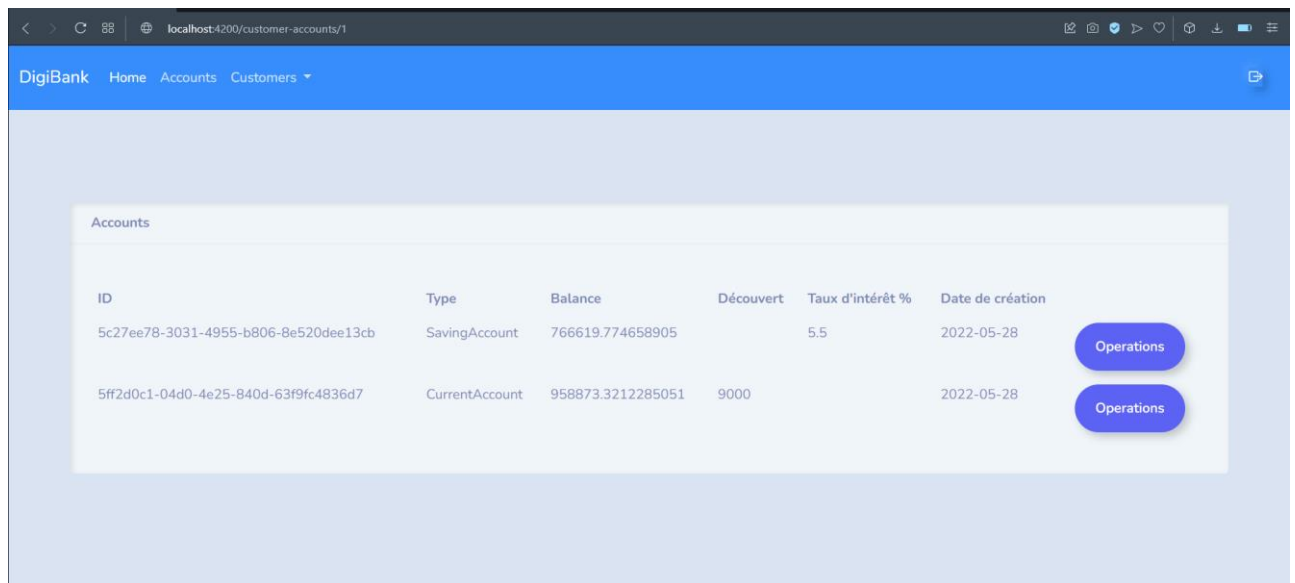
Keyword :

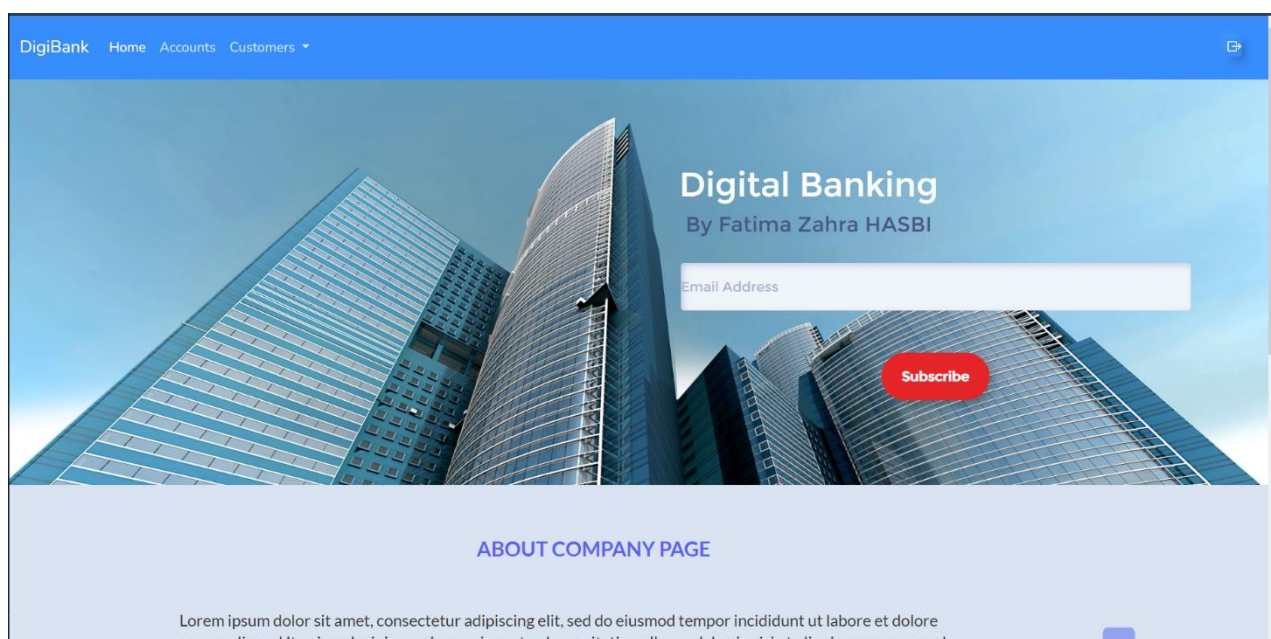
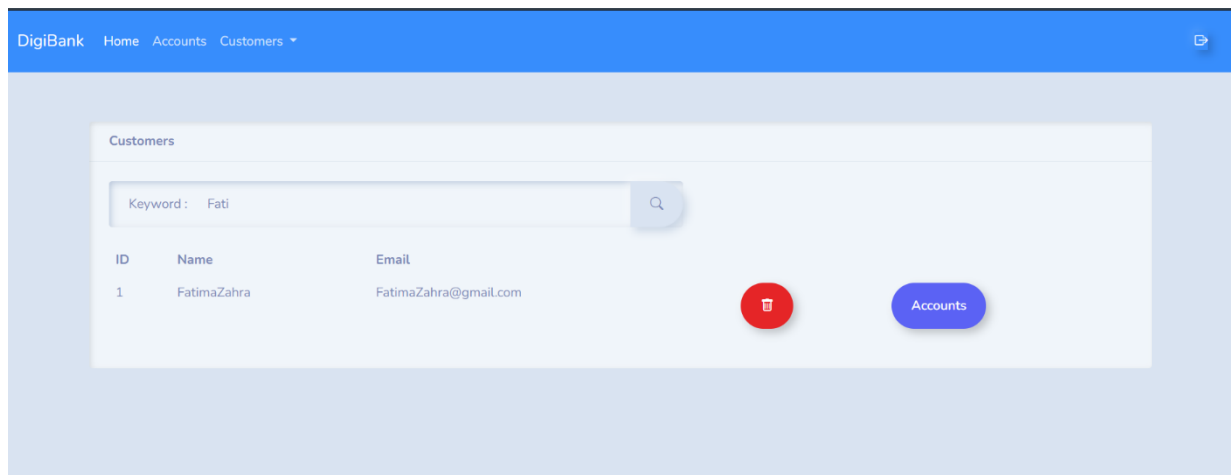
ID	Name	Email
1	FatimaZahra	FatimaZahra@gmail.com
2	Hasnaa	Hasnaa@gmail.com
3	Achraf	Achraf@gmail.com

Accounts

Accounts

Accounts





Le lien du code source du backend et de frontend avec Angular de l'application sur GitHub :

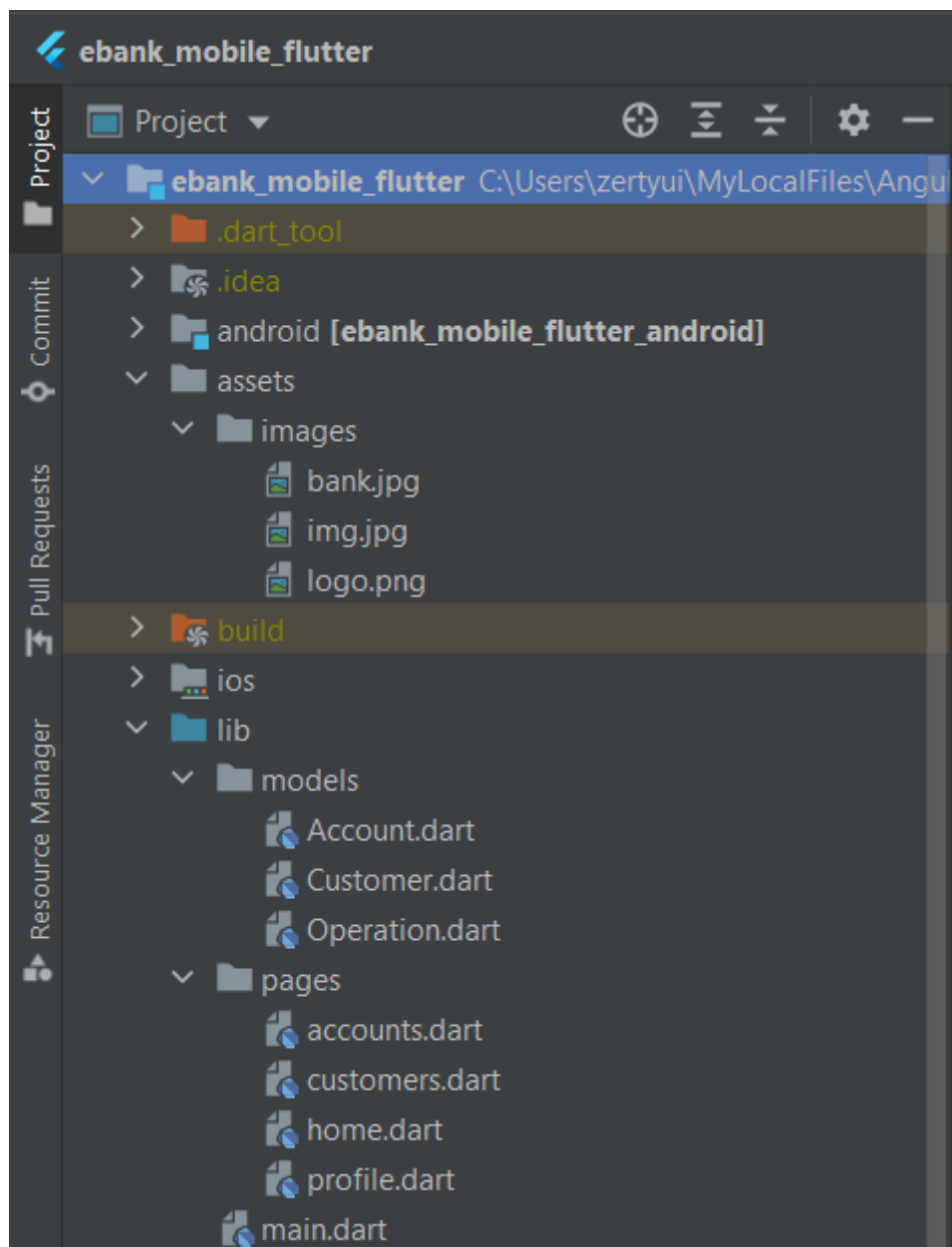
<https://github.com/FatimaZahraHASBI/Projet-Digital-Banking-Spring-Angular.git>

IV- Partie Mobile avec Flutter

1- Flutter ?

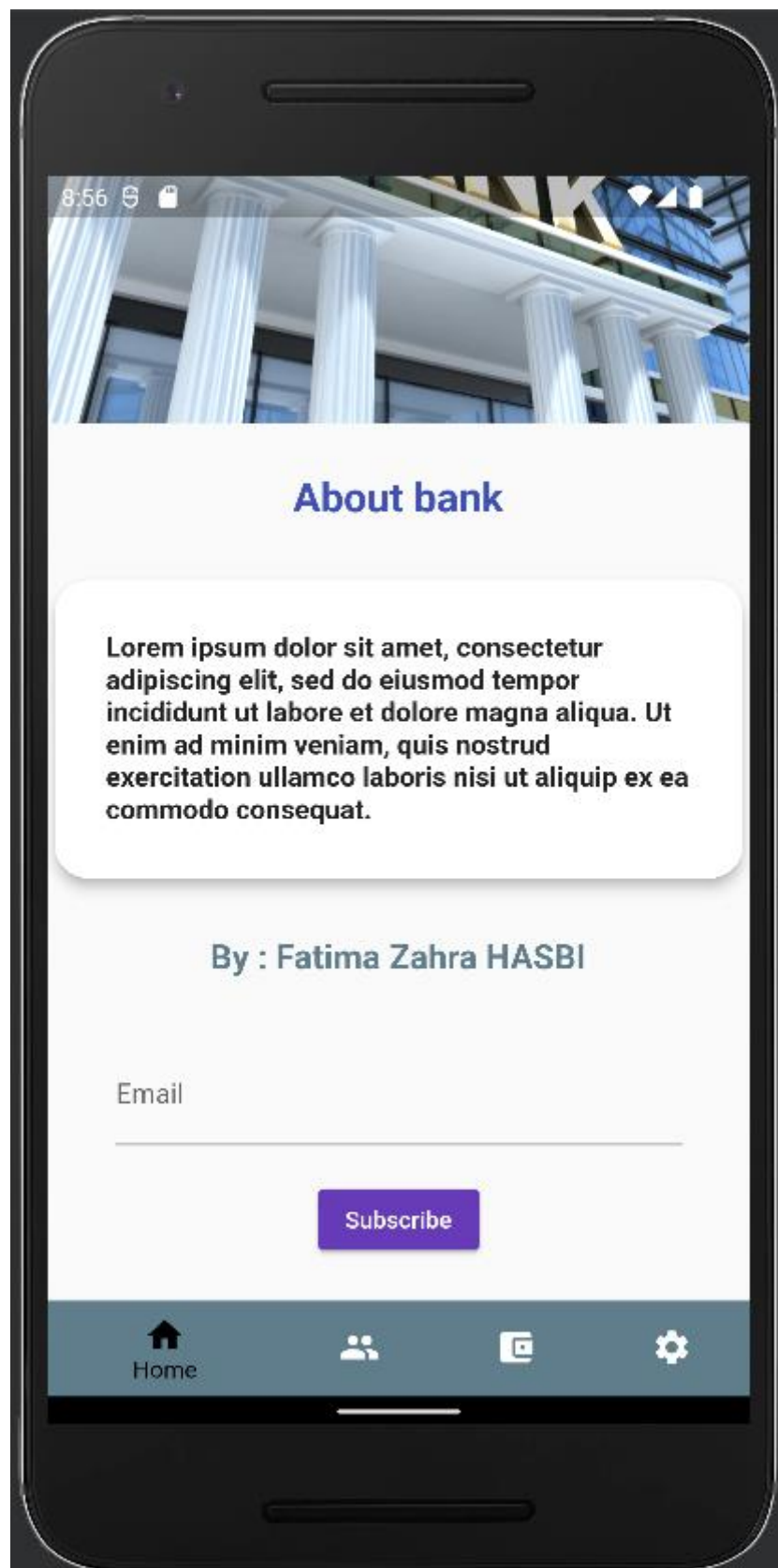
Flutter est un kit de développement logiciel d'interface utilisateur open-source créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code.

2- Structure de projet Digital Bank sous Flutter ?

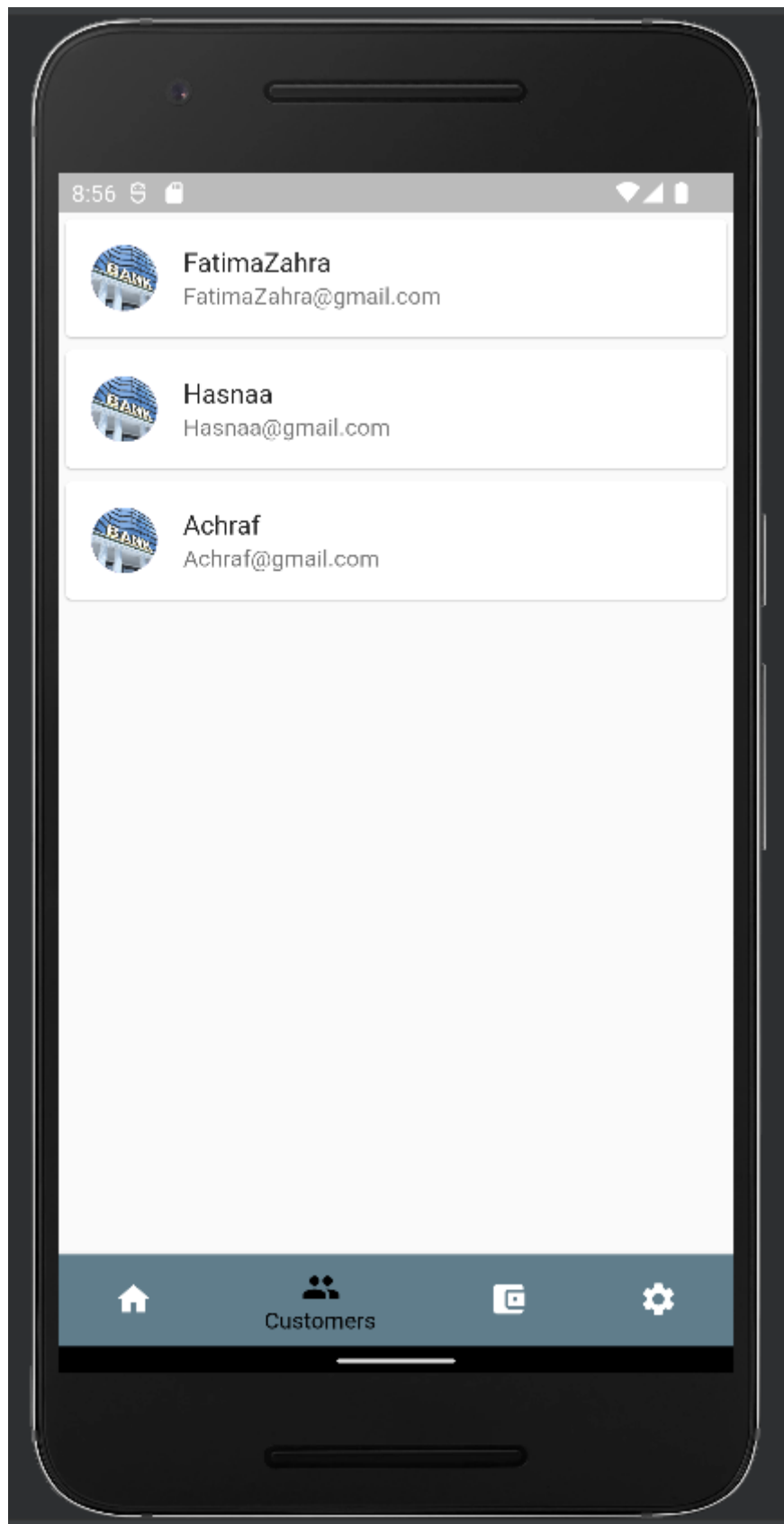


2- Démonstration de l'application





Liste de Customers depuis l'API



Le code source de la récupération des customers de l'API

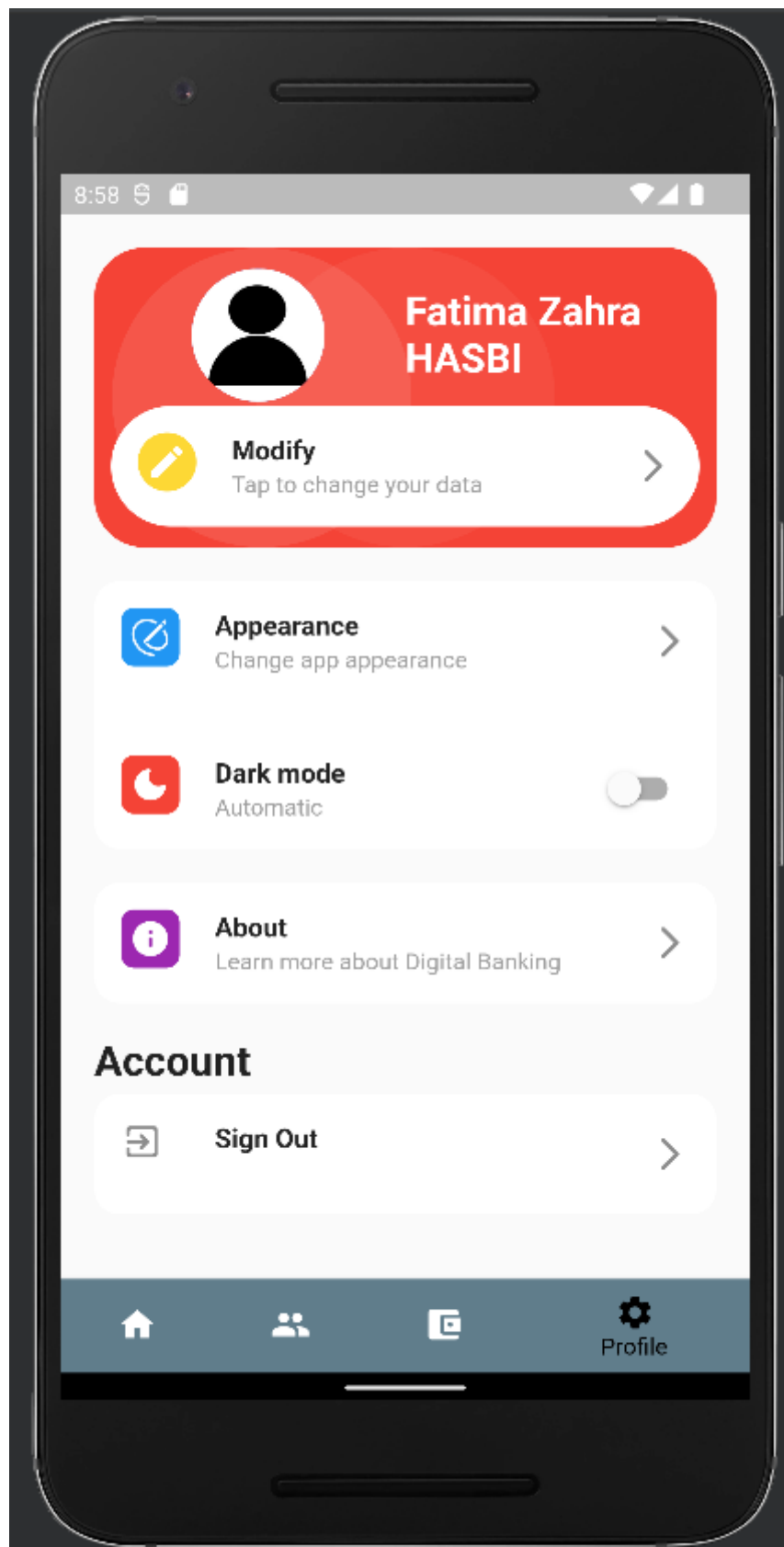
```
import 'package:ebank_mobile_flutter/models/Customer.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';

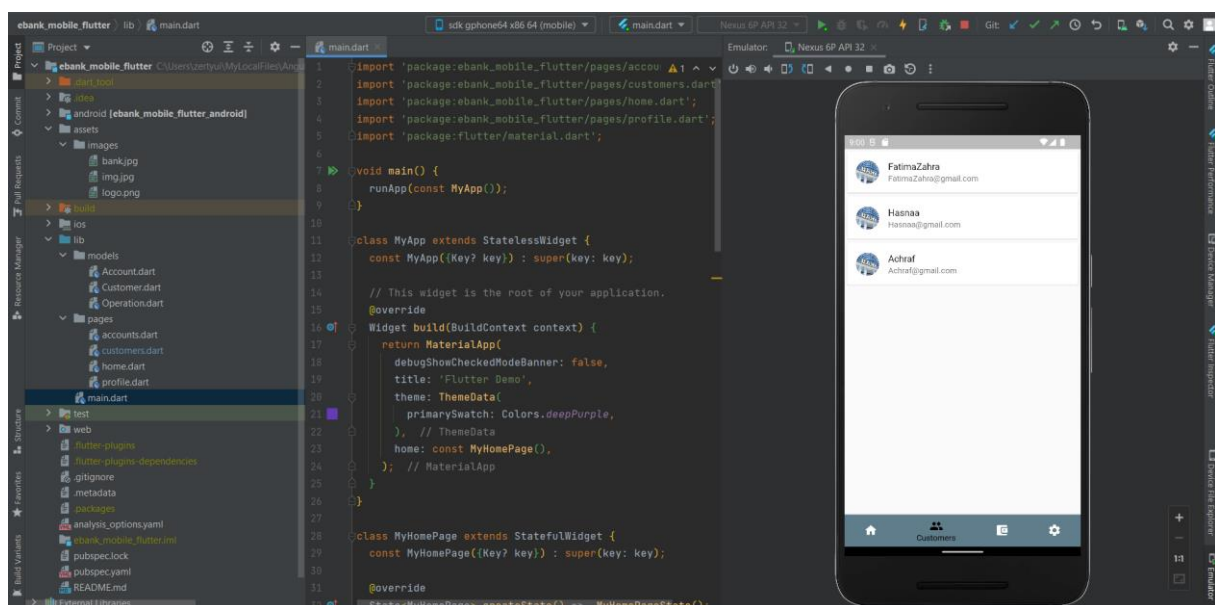
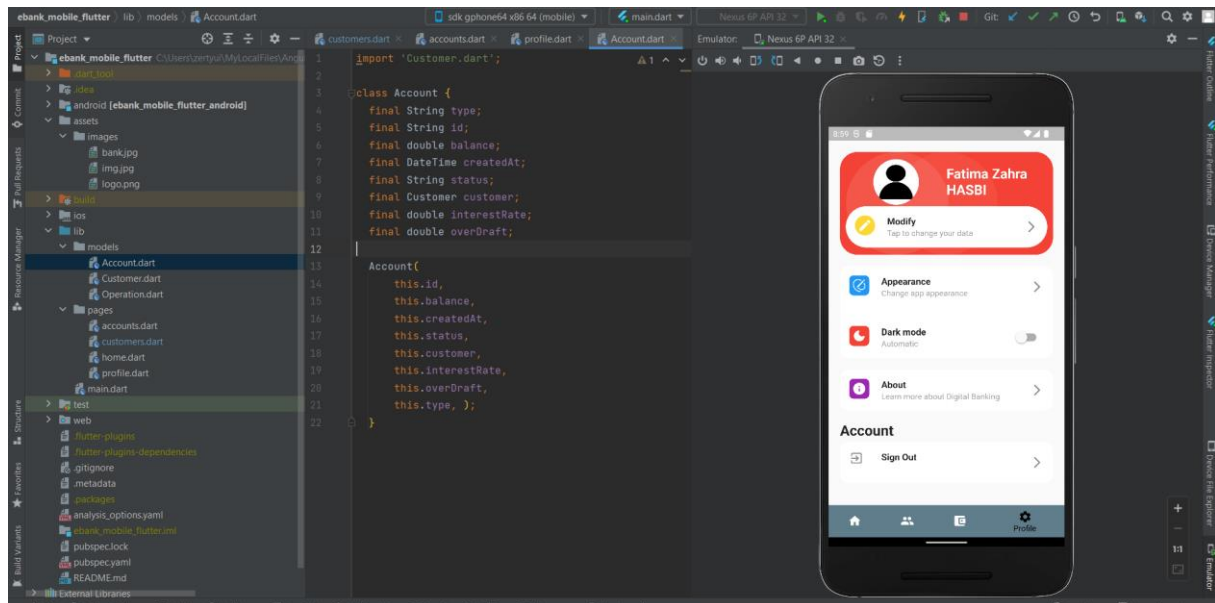
class Customers extends StatefulWidget {
  const Customers({Key? key}) : super(key: key);
  @override
  State<Customers> createState() => _CustomersState();
}

class _CustomersState extends State<Customers> {
  List<Customer> customers = [];
  Future<List<Customer>> _getCustomers() async {
    var data = await
http.get(Uri.parse('http://192.168.56.1:8085/customers'));
    var jsonData = json.decode(data.body);

    for(var c in jsonData){
      Customer customer = Customer(
        id: c['id'].toString(),
        name: c['name'],
        email: c['email'],
      );
      customers.add(customer);
    }
    return customers;
  }
  @override
  Widget build(BuildContext context) {
    return Center(
      child: FutureBuilder(
        future: _getCustomers(),
        builder: (BuildContext context, AsyncSnapshot snapshot){
          if(snapshot.data == null){
            return const Center(
              child: Text("Loading...")
            );
          } else {
            return ListView.builder(
              itemCount: snapshot.data.length,
              itemBuilder: (BuildContext context, int id) {
                return Card (child: ListTile(
                  leading: const CircleAvatar (backgroundImage:
                    AssetImage ("assets/images/bank.jpg")),
                  title: Text (snapshot.data[id].name),
                  subtitle: Text (snapshot.data[id].email),
                  onTap: () {},
                ));
              },
            );
          }
        },
      ),
    );
  }
}
```

```
}  
}
```





Le lien du code source du backend, de frontend avec Angular, et la partie mobile avec Flutter sur GitHub :

<https://github.com/FatimaZahraHASBI/Projet-Digital-Banking-Spring-Angular.git>

Conclusion

Ce projet vise à concevoir et implémenter une application web utilisant l'architecture web JEE et le Framework Spring avec le Framework Angular pour appliquer une approche "côté client" au lieu de l'approche classique "côté serveur", et réaliser par la suite la partie Frontend Mobile avec Flutter.

À cette fin, j'ai adopté une approche réfléchie et j'ai appliqué les bonnes pratiques de développement pour mettre en œuvre cette application, que ce rapport résumera pour vous.

Dans ce travail, j'ai appris à créer une application Web full stack pour gérer les clients bancaires en utilisant l'architecture JEE et le Framework Spring et le Framework javascript Angular.