In the name of Allah

List of services are as follow:

* Bank-service : Access right to database
* Atm-service: Access forbidden to database and make request through bank-service
* Service-registry: register all services via eureka
* Cloud-gateway: an interface between all services and user request
* Config-server: save all common settings in this service in order to change settings in only one place.

Port numbers for services are as follow:

* bank-service: 9001
* atm-service: 9002
* cloud-gateway: 9191
* config-server: 9010
* service-registry: 8761

Database:

* PostgreSQL
* Create a database name it as bankdb.

Java version:

* 12

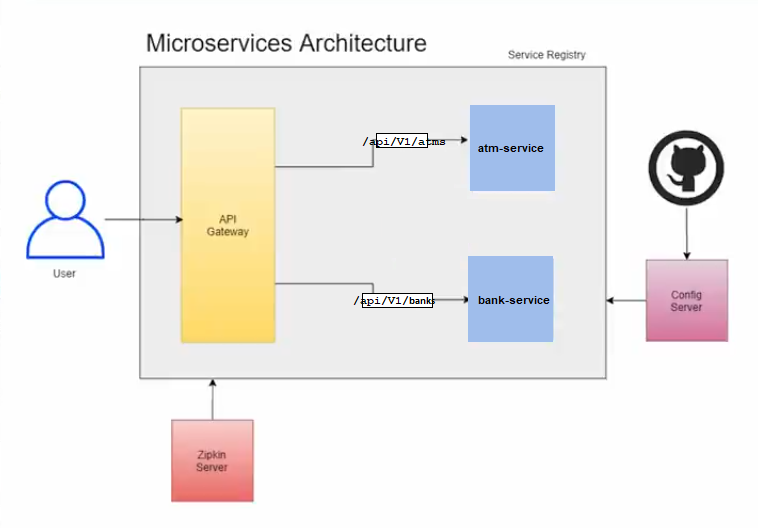
Comments:

* I have added some comments for documentation as an example (In the bank-service, all AccountEntity\*.java files, as well as PrintOutput.java and PasswordType.java files).
* Swagger for bank-service and atm-service is added. (It is better to limit access from bank-Api, but here I allowed user to do actions via bank-Api).
* JWT token authorization is implemented.
* Test in some parts of bank-service is implemented.
* As in requirement, there were no need to implement some Api’s as well as adding new user, account or card info, but in order to make more convenient, I have added some of them to easy work with swagger.
* Services can run in docker
* It is better to use common classes in one library and then share between services, but due to the small size of this project, I have not do that.
* Only Get and Post requests have been used.
* Exception managing is done only in bank-service as example.

How to Start:

* Clone the project in your system directory
  + Git clone <https://github.com/HesamoddinMonfared/SimpleATM>
* Go to atm-service directory and run the following command
  + docker build -t atm-service .
* Go to bank-service directory and run the following command
  + docker build -t bank-service .
* Go to cloud-gateway directory and run the following command
  + docker build -t cloud-gateway .
* Go to config-server directory and run the following command
  + docker build -t config-server .
* Go to service-registry directory and run the following command
  + docker build -t service-registry .
* Go to project root directory and run the following command
  + docker compose up
* Visit the bank-service swagger Api via pasting the following link into your browser
  + localhost:9001/swagger-ui.html
* Visit the atm-service swagger Api via pasting the following link into your browser
  + localhost:9002/swagger-ui.html

The architecture of project is as follow:



The order of running services is as follow:

* service-registry
* config-server
* cloud-gateway
* zipkin-server in order to run monitoring system via running the following command:
  + java -jar [zipkin-server-jar-file-name].jar
  + the address and port is usually: http://127.0.0.1:9411
* bank-service
* atm-service