**Problem Statement**

Build a distributed banking system.

**Goal**

Build a distributed banking system that allows multiple customers to withdraw or deposit money from the multiple branches in the bank.

Propagate a deposit /withdraw at a branch to all other branches, so that the balance at each branch is in sync.

Implement communication between processes using gRPC.

**Setup**

Python 3.8.0

pip 21.0.1

grpcio

grpcio-tools

**Implementation Process**

Declared a service with its interface functions and also service request and response message structures are declared in **service.proto** .

*service Bank {*

*rpc MsgDelivery (RequestMsg) returns (ReplyMsg) {}*

*}*

MsgDelivery function gets request from customer and other branch and supports operations query, deposit and withdraw. When it gets withdraw/deposit requests from customer it also propagates request to other branches so that they can update their balance.

Generated python client and server classes using **protoc** compiler and service.proto as input. Below are the two output files from compilation.

*service\_pb2.py & service\_pb2\_grpc.py*

Crated a class Branch.py by extending the generated class service\_pb2\_grpc.BankServicer, provided implementation for the service function MsgDelivery().

Created another class Customer.py, which is used to create all the customer processes. It has below functions

createStub()- creates client/stub for branch service.

executeEvents() – executes all the events provided in input

Created main.py, which is the main script to run and test the application.

Added a static configuration in all the files, which has port details for branch process.

bank \_config = [(1, 50051),(2, 50052),(3, 50053)].

**Results**

Input used in test, Input.txt

[

{ "id" : 1,"type" : "customer", "events" : [{ "id": 1, "interface":"query", "money": 400 },{ "id": 2, "interface":"withdraw", "money": 100 }] },

{ "id" : 2, "type" : "customer", "events" : [{ "id": 1, "interface":"query", "money": 400 },{ "id": 2, "interface":"withdraw", "money": 100 },{ "id": 3, "interface":"deposit", "money": 10 }] },

{"id" : 3, "type" : "customer","events" : [{ "id": 1, "interface":"query", "money": 400 },{ "id": 2, "interface":"withdraw", "money": 100 },{ "id": 3, "interface":"deposit", "money": 10 }] },

{ "id" : 1,"type" : "branch", "balance" : 400 },

{ "id" : 2, "type" : "branch", "balance" : 400 },

{"id" : 3,"type" : "branch","balance" : 400 }

]

Output form the customer processes

{'id': 2, 'recv': [{'interface': 'query', 'result': 'Success', 'money': 400.0}, {'interface': 'withdraw', 'result': 'Success'}, {'interface': 'deposit', 'result': 'Success'}, {'interface': 'query', 'result': 'Success', 'money': 120.0}]}

{'id': 1, 'recv': [{'interface': 'query', 'result': 'Success', 'money': 400.0}, {'interface': 'withdraw', 'result': 'Success'}, {'interface': 'query', 'result': 'Success', 'money': 120.0}]}

{'id': 3, 'recv': [{'interface': 'query', 'result': 'Success', 'money': 400.0}, {'interface': 'withdraw', 'result': 'Success'}, {'interface': 'deposit', 'result': 'Success'}, {'interface': 'query', 'result': 'Success', 'money': 120.0}]}

Below records show initial balance and all the customer transactions and the final balance at each branch.

400 - All the branches started with balance 400

-300 - Each customer withdrawn 100

+10 - Customer#2 deposited 10

+10 - Customer#3 deposited 10

120 - Final balance at each branch.