A link to your demo video.

( !!!!)

How to compile and run the code

Before running our Python code, you will need first to have the Rabbit MQ docker running with the following command:

*docker run -it --rm --name rabbitmq -p 5672:5672 -p 15672:15672 rabbitmq:3.11-management*

You will need Python3 so ensure that it is installed and that pip is installed with it.

To install the required packages please use the following command:

*pip install -r .\requirements.txt*

The command will just install the pika package (1.3.1).

To run the program, you just need to run the main.py script with Python.

*py main.py*

or:

*python3 main.py*

Depending on the installation of python.

Design documentation. Make sure to describe your concurrency strategies, failure handling, and consistency.

The main design idea uses the ComEntity class which is responsible for all the communications between the funding agencies, the university and the researchers using RabbitMQ.

Each ComEntity is a thread and only communicate which other parts of the program using RabbitMQ after being started. The ComEntity is an abstract class used to implement the ability to send and receive messages (with the sendMsg and the receiveResponse methods). And by ensuring that each child can implement a behaviour by overriding the behaviour method.  
The ComEntity class also provide minor utility functions.

We have a TimeKeeper class that extends ComEntity useful to count the time. It sends time update via RabbitMQ to the funding agencies and the universities.

After that we have the funding agency (FundingAgency class) that extends ComEntity that will receive via RabbitMQ the research proposals of the researchers and will decide if it is accepted or not based on the amount of money it has left and other criteria.  
If the proposal is accepted the researcher will be notified and the university will also be notified.

The University class that extends ComEntity will keep track of the money the researcher has on the fund that have been allocated when a research proposal has been accepted.  
The university can deal with the researcher withdrawing money from their account, adding and removing member to their team and managing new research account and the ability to delete them if needs be.

And finally, the Researcher class that extends ComEntity will make research proposal until one is accepted to a funding agency and if the researcher has an accepted proposal they will now work on it (meaning withdrawing money from their account from time to time).

The research class is used to manage in a tidy way the interactions between the university and the research accounts.

Since we use RabbitMQ there is no concurrency issues since the threads do not interact directly with each other’s.

Our program has some trouble dealing with crashes or start/stop cycles without cleaning the messages stored in RabbitMQ.

A plot/figure and a description of a concurrency, performance or scalability study you did that shows how the system performs with many clients/requests and/or deals with failure.

What aspects of the project you got working / not working / partly working

The following is working:  
  
We managed to implement 3 researchers (can be more) dealing with 2 funding agencies (can be more) and one university.

The research proposals are made of an ID (the researcher’s ID), a title, a project description and the requested amount of funding.

The funding agency has a budget, a minimum and a maximum for the application fund amount to be accepted and it keeps track of how much it has left.

The funding agency will respond to the researcher if the proposal is accepted or not and will transmit the relevant information to the university if the research proposal is accepted.

The university does store all information related to research accounts, users, and the transactions made and delete the account if the deadline is crossed or if there is no more money on the account.

The researcher who gets a project approved can add and remove researchers. Withdraw funds if there is still money in the account.

The following is not implemented:

The researchers cannot access the details of their account or list the transactions made on that account.

Clovis GILLES:

In this assignment I don't have learnt a lot of things. Indeed, I have already worked with the language Python in a company. And it uses some multithreading notions and therefore, some concurrency notions. Despite this, I have learned how to use and work with RabbitMQ with Python and note that, I have to relearn the particularities of the language like for example the syntax of a ternary. Like I said previously, we use some multithreading notions like the class Thread that our class ComEntity inherited. Therefore, we have to use some notion of concurrency. For the protocol of communication hardly all the exchanges/communications in the program are with the TCP schema: a request implies a response. Note that it can also be linked to the lectures material.

My part in this program was the development of the following:

* University Class
* FundingAgency Class
* Researcher Class
* Research Class
* ResearcherAccount Class
* The protocol of communication for every exchange
* The generation of subjects of research and names of the objects

If we have to recode this project, I think that I would make a better communication protocol but also a better behaviour for our researchers. The possibility to have multiple Universities (this is linked with the behaviour of the researchers) and maybe an output to the research, with a percentage of success which can be refunded by the Funding Agency or some consequences with the end of a research project.

Max 0.5 (half) page personal reflections from each student:

1. what did you learn?
2. how does it relate to course material?
3. What work did you contribute?
4. What would you do differently next time?