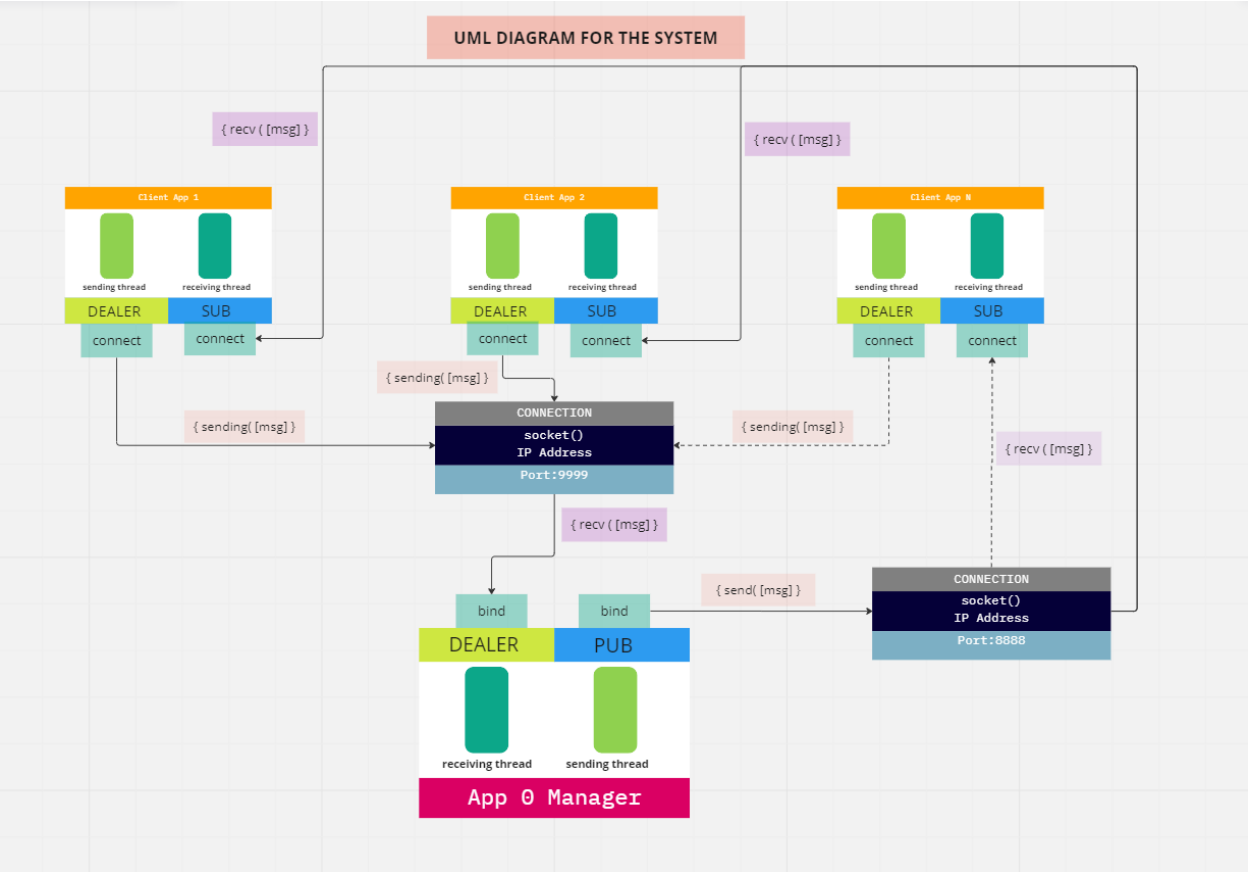
Documentation and User Guide

Link to Github Repository for the Technical Assessment: [Click here for the link.]



Assumption Made in the Design of the "App 0 Manager" - "Client App {N}" System.

1. When the "App 0 Manager" sends a message, every Client App connected to the App 0 Manager receives the same message at the exact same time. In other words, the App 0 Manager will be broadcasting its messages to the Client Apps connected to it – This will be a **Pub-Sub** data distribution pattern.
    a. *Alternative implementation to this is when the App Manager identifies the last client it received a message from and sends **THE SAME MESSAGE** to the last identified client whenever it needs to send a message. (This implementation can be achieved using the ROUTER design pattern) – **Fair-queued** approach (Not implemented in this solution)*
2. Given the following specifications
    a. App 0 Manager – should be able to send and receive messages with any client App {N}
    b. "Client App 0" – can send and receive messages from "App 0 Manager"
    c. "Client App 1" – can send and receive messages from "App 0 Manager"

    I concluded that;

    i) Any "Client App {N}" can send a message or messages to the "App 0 Manager" without an initial message from the "App 0 Manager"
    ii) Likewise "App 0 Manager" can send out a message or messages without receiving any initial message from any "Client App" connected to it

    Note:

    a. In the case that a connection has not been established between any client App and the "App 0 Manager", if a client sends a message to the "App 0 Manager", it will be received when the "App 0 Manager" binds to the port at a later time which the clients are connected to.
    b. In the case that a client is not connected to the "App 0 Manager", previous messages broadcasted/published out by the "App 0 Manager" will never be received by the client. Only subsequent messages broadcasted by the "App 0 Manager" will be received after a connection is established on the clients end.

Classes and Functions

Two application were made implemented namely

i. The AppManager application
ii. The ClientApp application

Note: *Only one Client App was created. However, multiple client program could be run at the same time, thus creating a single application for the client was optimum for this system.*

Classes and Methods

AppManager application	**ClientApp application**
AppManager -> Class	ClientApp -> Class
__init__() -> method	__init__() -> method
run() -> method	run() -> method
exit() -> method	exit() -> method
sending_messages() -> method	sending_messages() -> method
receiving_messages() -> method	receiving_messages() -> method

How to execute application.

Files needed:

1. AppManager.py
2. ClientApp.py
3. requirements.txt

This solution is Terminal-based and the following steps can be followed for execution.

Step1: Have python3 installed on your machine.

Step2: Create a virtual environment for the program to be executed

Link to creating a virtual environment on windows: CLICK HERE

Link to creating a virtual environment on Linux: CLICK HERE

i. activate the virtual environment
ii. run this following command to install pyzmp

```
pip install -r requirements.txt
```

```
or
```

```
pip3 install -r requirements.txt
```

iii. Open 3 terminals and change path to the location where both AppManager.py and ClientApp.py are installed. Type each of the following commands into the terminals to run the applications

```
python3 AppManager.py
```

```
python3 ClientApp.py
```

```
python3 ClientApp.py
```

Sample Demo