

# NetSys Distributed Systems CW - ReadMe

NB: This file best read in a dedicated Markdown reader/renderer. A PDF file has been supplied.

## Prerequisites

This system requires:

- an installation of Python 3.6+, that you can invoke from a command prompt (ie a command such as `python` or `python3.6` exists in your PATH variable)
- an installation of the Pyro4 package for the version of Python one is planning to use
- an installation of the Postcodes IO API package for the version of Python one is planning to use

For instructions on how to install Pyro4, visit <https://pyro4.readthedocs.io/en/stable/install.html>

For instructions on how to install the Postcodes IO API package, visit <https://pypi.org/project/postcodes-io-api/>

## Included in the submission

- `README.MD`: what you're reading now!
- `README.pdf`: a PDF version of what you're reading now!
- `client.py`: the Just Hungry client
- `front.py`: the frontend server
- `back.py`: the backend server(s)
- `NetSysDSDiagram.pdf`: the System Design Diagram

## Running the system

Henceforth, assume each instruction to run a command requires its own process/terminal window/command prompt - whichever works easiest for you. I developed and tested this system on Windows 10, running many (so many...) Powershell windows, each to execute a different command. Additionally, I will:

- use `python` as my python command - replace as appropriate with your own, for example if you have `python3.6`
- use Windows-style file slashes (ie. `\` instead of `/`) - again, replace where appropriate, ie. with UNIX-style
- assume all commands are being invoked from one directory that contains all the source code - again, replace where appropriate with full filepaths.

First, run a Pyro4 nameserver with the command `pyro4-ns`. This will allow the frontend server and backend servers to bind themselves to the network.

Next, run at least one backend server with the command `python .\back.py`. Backend servers can be open and closed between method calls from the client/frontend, but at least one needs to be active at any one time for the frontend server to start, or to continue working.

Run exactly one frontend server with the command `python .\front.py`

Finally, run at least one client with the command `python .\client.py`

The system should now be ready to use. Interact with it from the client to (hopefully!) see changes reflected on the backend servers.

To test the replication transparency, you can try this:

- start the frontend server with one backend server running
- add a new item of food to the menu with the client's SUGGEST option
- start two new backend servers
- place an order with ADD and VIEW
- close the first backend server (eg. Ctrl-C in the terminal window where it is running (on Windows)), keeping the second and third open
- use ORDERS and MENU to see the correct, up-to date respective lists stored on the new servers

## External web services

Two services have been utilised for verifying postcodes in this system - if one is unavailable, the other will be used. They have been chosen for their free availability and simplicity of use - both lend themselves to a very simple True/False check, which is precisely how they have been implemented in the `back.py` source code.

Find out more about these two services below:

<http://postcodes.io/about> (provided under the MIT license)

<https://www.getthedata.com/open-postcode-geo-api> (provided under free use with attribution)

MIT License:

Copyright 2020 hrhq57

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.