



IS5006 INTELLIGENT SYSTEM DEPLOYMENT

HOW-TO GUIDE

Prepared by:

Group 7

Name	Matriculation Number
Ge Xiaomeng	A0112747H
Goh Khai Hong	A0212197H
Shen Siyuan	A0112489B
Xue Bin	A0039717X
Yong Chee Xian Matthew	A0090988J
Zou Yang	A0070179A

Semester 2, AY2020/2021

NATIONAL UNIVERSITY OF SINGAPORE

Table of Content

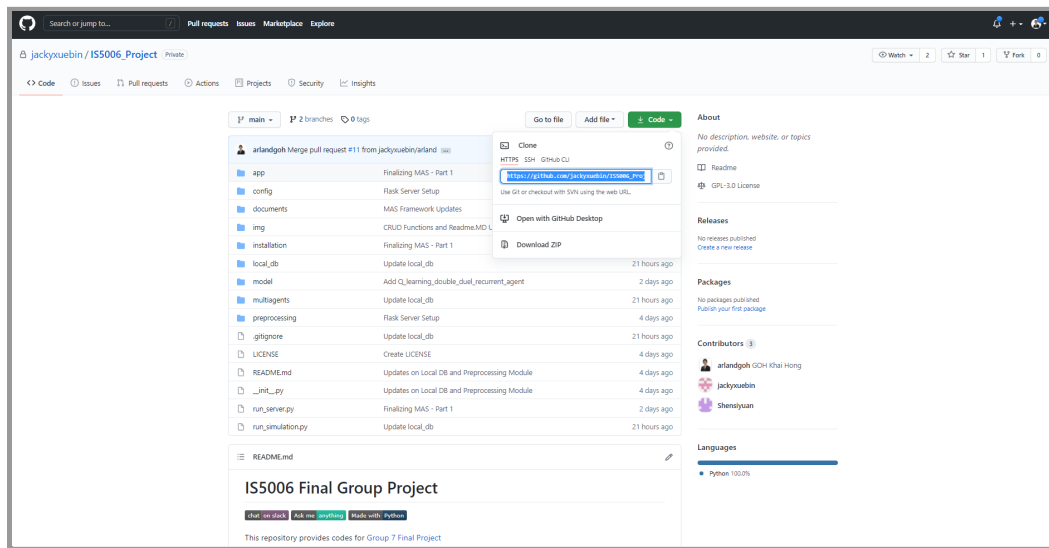
1. Clone the Project Repository to Your Local Machine	2
2. Setup the Environment for Group 7 Multi-agent System	3
2.1 Windows User	3
2.2 Ubuntu User	4
2.3 Mac User	5
3. Quick Start [Without Running Flask Server]	7
4. Start the Multi-agent System in Flask Server	8
5. Flow Chart for Group 7 Multi-agent System	9
6. Contact US	10

Group 7 Multi-agent System Installation Guide

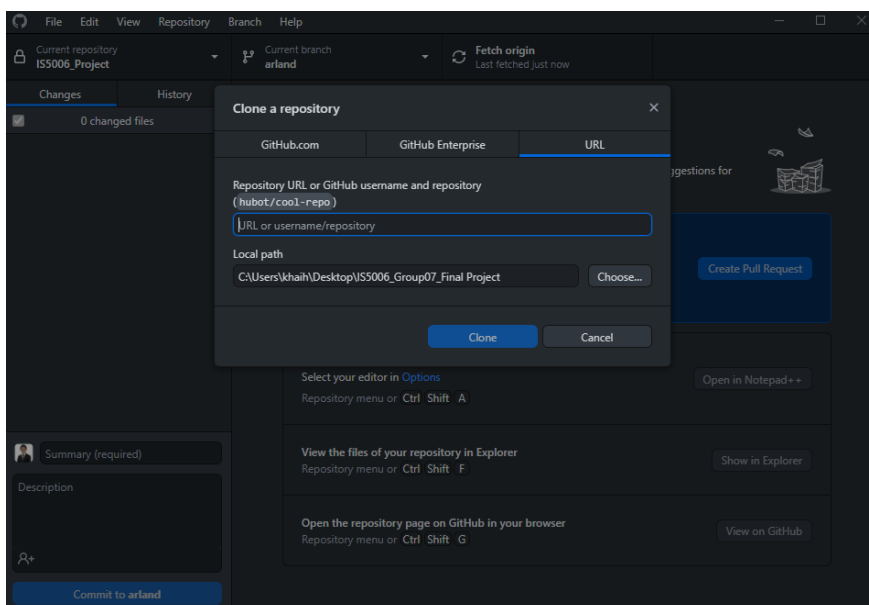
1. Clone the Project Repository to Your Local Machine

To clone the project repository, please go to the [Github repository](https://github.com/jackyxuebin/IS5006_Project) using your browser and run the following command on your command prompt/terminal:

- `git clone https://github.com/jackyxuebin/IS5006_Project.git`



If you prefer Graphical User Interface (GUI), you can download the [GitHub Desktop](#) and use it to clone the repository. Go to File > Clone repository and paste the [URL](#).



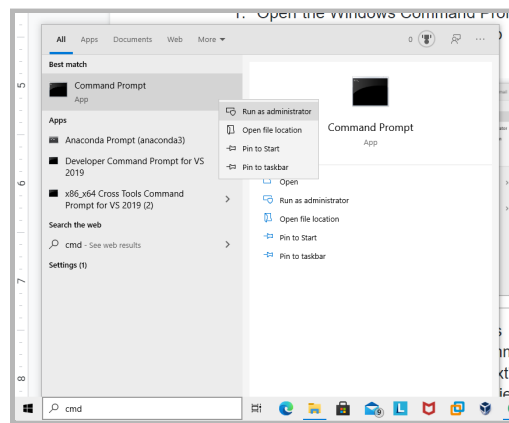
2. Setup the Environment for Group 7 Multi-agent System

You may need to install some Python dependencies on your machine before running our system. We assume that you are running a Windows machine. The system has been tested running under **Python >= 3.7.0** with the following packages installed. The packages/libraries are installed using **pip3** (package manager for Python packages).

2.1 Windows User

To install these dependencies:

1. Open the Windows Command Prompt as an administrator by simply right-clicking on the Command Prompt app in the startup menu and select 'Run As Administrator'.



2. Navigate to the directory of this readme file (which also contains the requirements.txt) in the command prompt.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19041.867]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Users\khaih\Desktop\IS5006_Group07_Final Project\IS5006_Project\installation

C:\Users\khaih\Desktop\IS5006_Group07_Final Project\IS5006_Project\installation>dir
Volume in drive C is Windows-SSD
Volume Serial Number is 76AC-62D6

Directory of C:\Users\khaih\Desktop\IS5006_Group07_Final Project\IS5006_Project\installation

06/04/2021  11:40 pm    <DIR>          .
06/04/2021  11:40 pm    <DIR>          ..
06/04/2021  11:42 pm               1,681 README.md
06/04/2021  11:41 pm               2,150 requirements.txt
                2 File(s)              3,831 bytes
                2 Dir(s) 158,488,965,120 bytes free
```

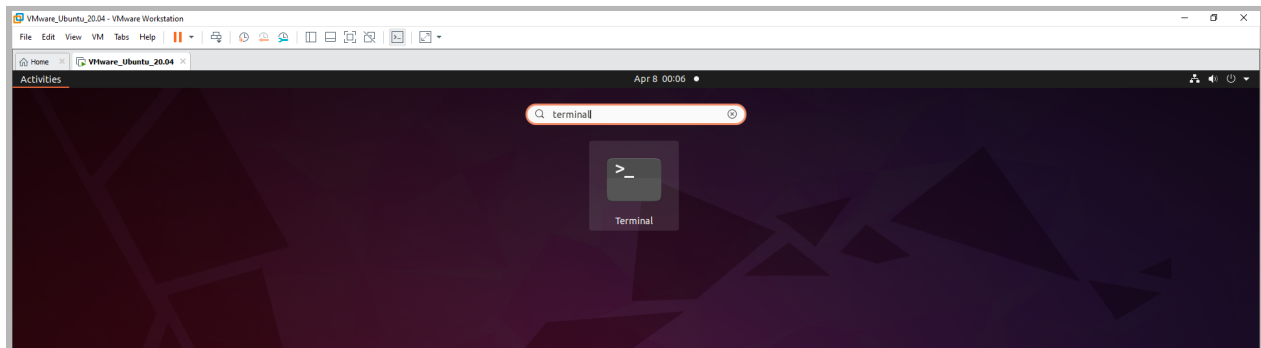
3. Run 'pip3 install -r requirements.txt'

```
C:\Users\khaih\Desktop\IS5006_Group07_Final Project\IS5006_Project\installation>pip3 install -r requirements.txt
```

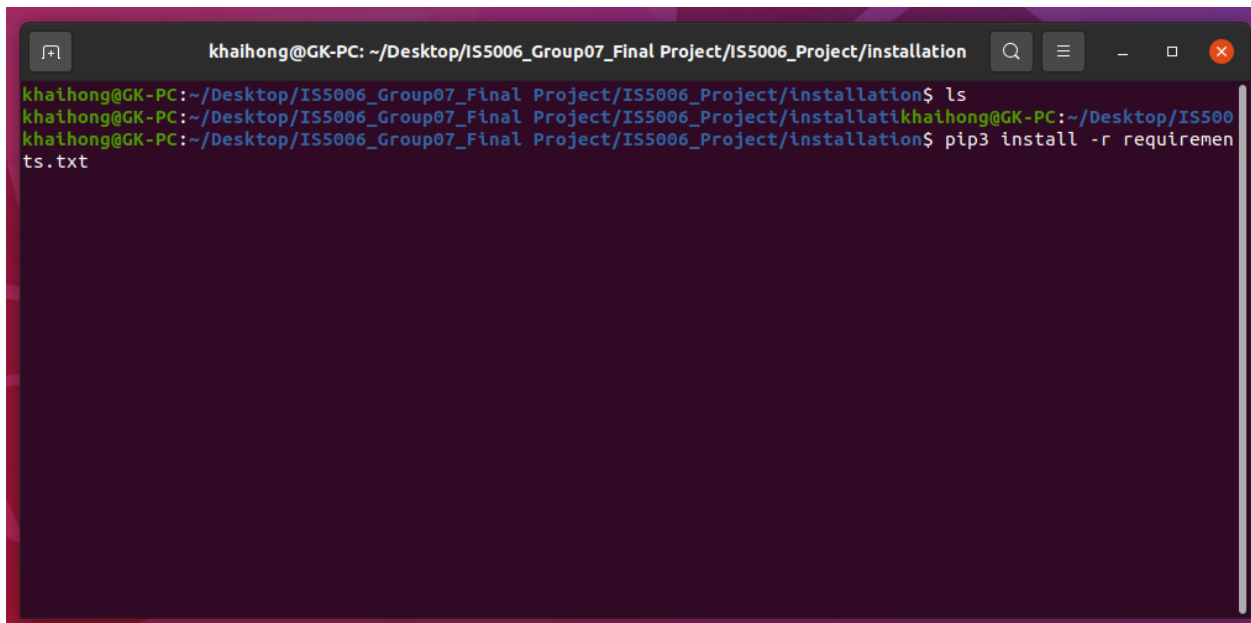
2.2 Ubuntu User

To install these dependencies:

1. Open the Terminal.



2. Navigate to the directory of this readme file (which also contains the requirements.txt) in the Terminal.
3. Run 'pip3 install -r requirements.txt'



2.3 Mac User

To setup python env:

1. Open the Terminal and update homebrew with `brew update --verbose`

```
IS5006_Project — -bash — 128x31
[~/Developments/xm/IS5006_Project (main)]$ brew update
Updated 2 taps (homebrew/core and homebrew/cask).
==> New Formulae
net-tools
==> Updated Formulae
clusterctl
cleanmymac
goland
==> Deleted Casks
battery-guardian
systemd
exploitdb
httpx
mediahuman-audio-converter
purei-play
taskwarrior-tui
tencent-docs
xlink-kai
xcodegen
yandex
flash-ppapi
beatport-pro

You have 36 outdated formulae installed.
You can upgrade them with brew upgrade
or list them with brew outdated.
~/Developments/xm/IS5006_Project (main)$
```

2. Install pyenv with homebrew with `brew install pyenv`

```
[~/Developments/xm/IS5006_Project (main)]$ brew install pyenv
Warning: pyenv 1.2.26 is already installed and up-to-date.
To reinstall 1.2.26, run:
  brew reinstall pyenv
```

3. Install Python 3.7 with pyenv with `pyenv install 3.7.3`

```
[~/Developments/xm/IS5006_Project (main)]$ pyenv install 3.7.3
pyenv: /Users/huangbh/.pyenv/versions/3.7.3 already exists
continue with installation? (y/N)
```

4. Set global default Python version with `pyenv global 3.7.3`

```
[~/Developments/xm/IS5006_Project (main)]$ pyenv global 3.7.3
~/Developments/xm/IS5006_Project (main)$
```

5. Update .bash_profile with

```
if command -v pyenv 1>/dev/null 2>&1; then
```

```
eval "$(pyenv init -)"
```

```
fi
```

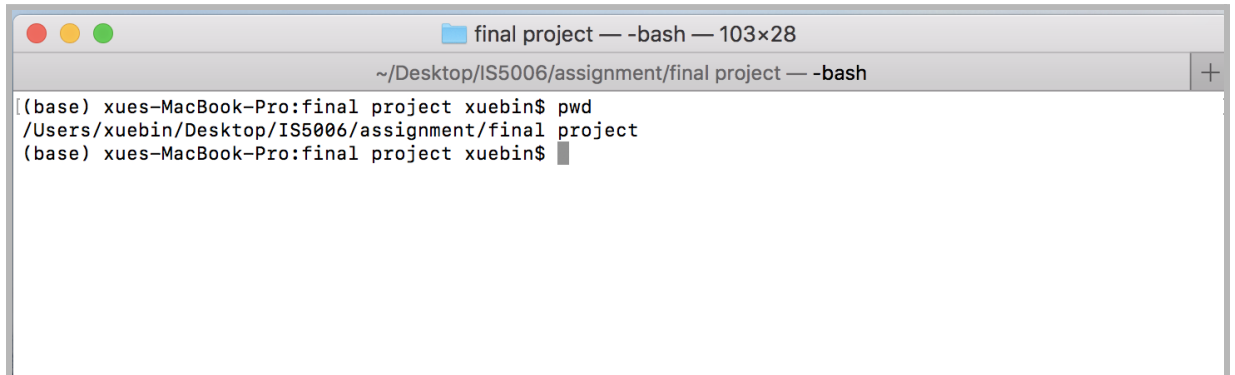
```
if command -v pyenv 1>/dev/null 2>&1; then
eval "$(pyenv init -)"
fi
```

6. Source .bash_profile with `source ~/.bash_profile`

```
[~/Developments/xm/IS5006_Project (main)]$ source ~/.bash_profile
~/Developments/xm/IS5006_Project (main)$
```

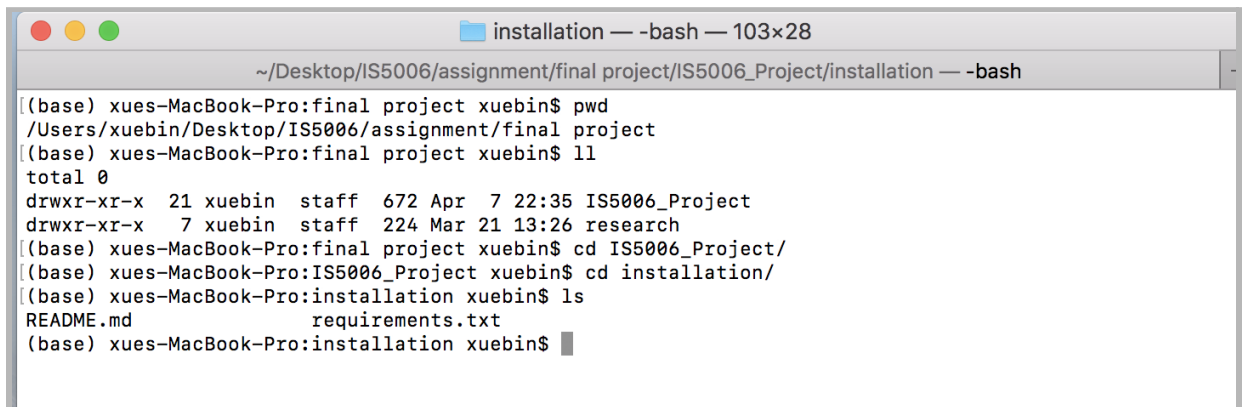
To install these dependencies:

1. Open the Terminal.



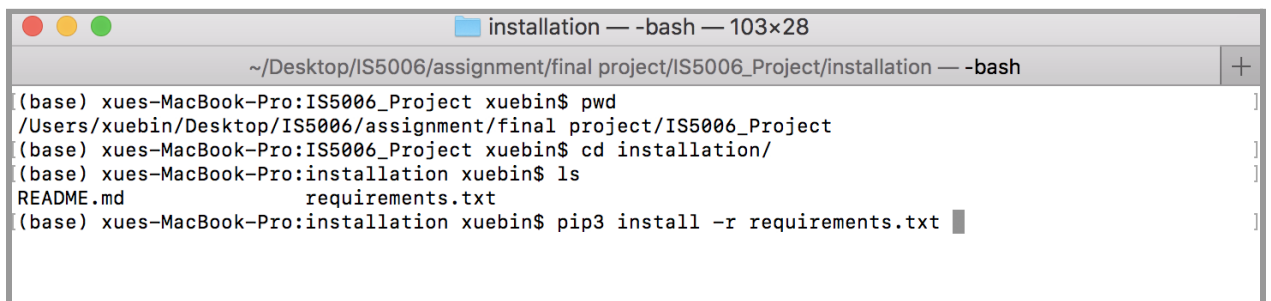
```
final project — -bash — 103x28
~/Desktop/IS5006/assignment/final project — -bash
((base) xues-MacBook-Pro:final project xuebin$ pwd
/Users/xuebin/Desktop/IS5006/assignment/final project
((base) xues-MacBook-Pro:final project xuebin$
```

2. Navigate to the directory of this readme file (which also contains the requirements.txt) in the Terminal.



```
installation — -bash — 103x28
~/Desktop/IS5006/assignment/final project/IS5006_Project/installation — -bash
((base) xues-MacBook-Pro:final project xuebin$ pwd
/Users/xuebin/Desktop/IS5006/assignment/final project
((base) xues-MacBook-Pro:final project xuebin$ ll
total 0
drwxr-xr-x  21 xuebin  staff   672 Apr  7 22:35 IS5006_Project
drwxr-xr-x   7 xuebin  staff   224 Mar 21 13:26 research
((base) xues-MacBook-Pro:final project xuebin$ cd IS5006_Project/
((base) xues-MacBook-Pro:IS5006_Project xuebin$ cd installation/
((base) xues-MacBook-Pro:installation xuebin$ ls
README.md      requirements.txt
((base) xues-MacBook-Pro:installation xuebin$
```

3. Run 'pip3 install -r requirements.txt'



```
installation — -bash — 103x28
~/Desktop/IS5006/assignment/final project/IS5006_Project/installation — -bash
((base) xues-MacBook-Pro:IS5006_Project xuebin$ pwd
/Users/xuebin/Desktop/IS5006/assignment/final project/IS5006_Project
((base) xues-MacBook-Pro:IS5006_Project xuebin$ cd installation/
((base) xues-MacBook-Pro:installation xuebin$ ls
README.md      requirements.txt
((base) xues-MacBook-Pro:installation xuebin$ pip3 install -r requirements.txt
```

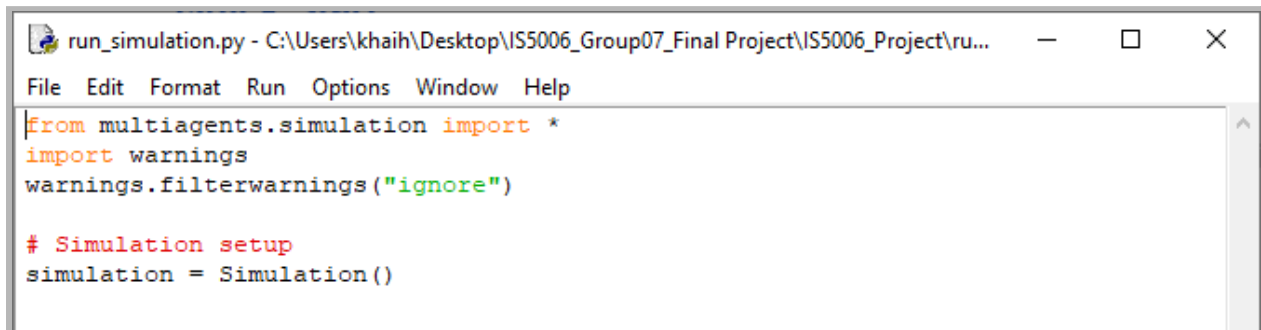
3. Quick Start [Without Running Flask Server]

For Ubuntu, Mac OS X and Windows 10:

To quick start the system, please navigate to the project folder and use the following command in your command prompt/Terminal:

- `python3 run_simulation.py`

or open the 'run_simulation.py' in Python IDE or any other IDE and run it.

A screenshot of a Python IDE window. The title bar shows the file name 'run_simulation.py' and the path 'C:\Users\khaih\Desktop\IS5006_Group07_Final Project\IS5006_Project\ru...'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor displays the following Python code:

```
from multiagents.simulation import *
import warnings
warnings.filterwarnings("ignore")

# Simulation setup
simulation = Simulation()
```


4. Start the Multi-agent System in Flask Server

For Ubuntu, Mac OS X and Windows 10:

To start the Flask Server, please use the following command in your command prompt/Terminal, the system will be served on port 7000:

- **python3 run_server.py**

or open the 'run_simulation.py' in Python IDE or any other IDE and run it



```
run_server.py - C:\Users\khaih\Desktop\IS5006_Group07_Final Project\IS5006_Project\run_ser...
File Edit Format Run Options Window Help
from app.utils.logger import *
from config.mas_config import *
from gevent.pywsgi import WSGIServer

from app import create_app
import sys

def run_server():
    try:
        system_logger_agent = Logger('server_logger', info_flag = True)
        system_logger = system_logger_agent.setup_logger('server_logger', os.pat

        app = create_app()
        app_server = WSGIServer(('0.0.0.0', 7000), app)

        logging.info("The flask app has been started!")
        system_logger_agent.start_logging(system_logger, 'The server has started
        app_server.serve_forever()

    except Exception as e:
        system_logger_agent.start_logging(system_logger, 'Exception - ' + str(e)

    # Raise KeyboardInterrupt to stop our MAS system
    except KeyboardInterrupt as e:
        system_logger_agent.start_logging(system_logger, 'The server has been sh

if __name__ == "__main__":
    run_server()
```

To run the Multi-agent system in the server, please open your browser and send the following http request to the server (using the address bar in the browser):

- http://localhost:7000/run_simulation

To check the cumulative profit/loss graph, you may use the /cumulative_profit_plot API:

- http://localhost:7000/cumulative_profit_plot

To check the other plot, you may invoke all/one of the following APIs below:

- http://localhost:7000/signals_action_plot
- http://localhost:7000/action_pnl_plot
- http://localhost:7000/takeprofit_stoploss_plot

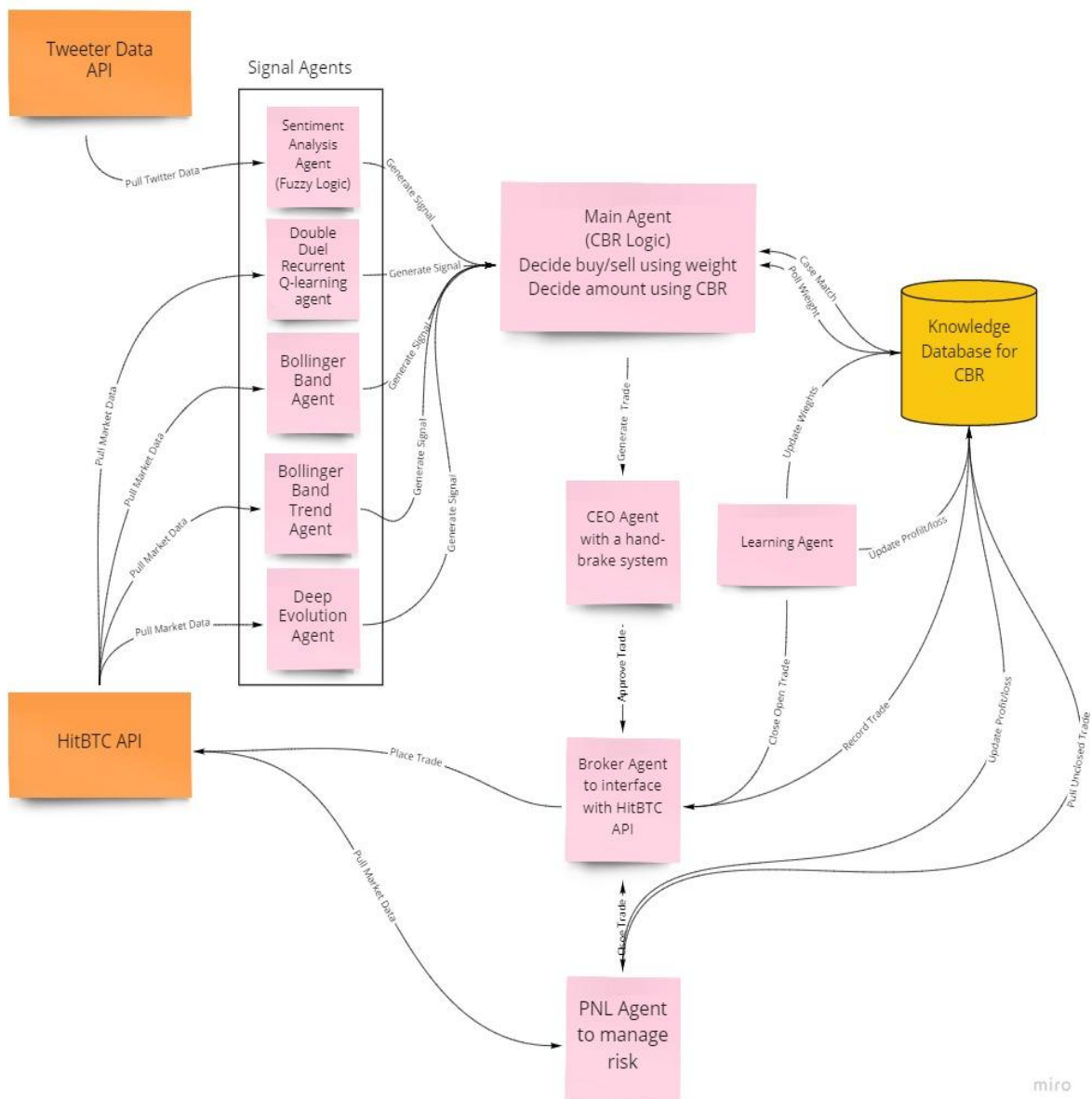
5. Flow Chart for Group 7 Multi-agent System

The flow chart below shows how the agents communicate among themselves. The external data sources used in the multi-agent system are:

1. Twitter data (Orange color box)
2. OHLCV data using the ccxt library and HitBTC APIs (Orange color box).

The dataset generated from the system will be saved in **Knowledge Database**:

1. locally in local database in 'local_db' directory as csv file
2. on cloud (Google Drive) as Google Sheets



6. Contact US

If any problems occur, please email one of the following members and paste the error message.

- Ge Xiaomeng (e0403444@u.nus.edu)
- Goh Khai Hong (e0503476@u.nus.edu),
- Shen Siyuan (e0403443@u.nus.edu),
- Xue Bin (e0573004@u.nus.edu),
- Yong Chee Xian Matthew (e0573096@u.nus.edu),
- Zou Yang (e0403394@u.nus.edu)