

# Homework 1 Report

R12942099 涂子峻

## 1. 系統架構及分層

- I. UI 層 (interface 資料夾)：處理 CLI 上的用戶輸入與輸出。
- II. 應用層 (application 資料夾)：實現市集流程，包括用戶註冊、物品管理等功能。
- III. 領域層 (models, managers 資料夾)：定義及控制 User、Listing 和 Category。
- IV. 儲存層 (database 資料夾)：使用 JSON 作為儲存格式。

## 2. 實作

使用 python 撰寫。

## 指令

- REGISTER <username>

```
class UserManager:
    def __init__(self, db: Database):
        self.db = db
        self.data = self.db.load()

    def register_user(self, username):
        self.data = self.db.load()
        if username.lower() not in self.data["users"]:
            user = User(username)
            self.data["users"][username] = user.__dict__
            self.db.save(self.data)
            return "Success"
        else:
            return "Error - user already existing"
```

建立一個新的 user 資料結構，將其視為一個 dictionary 存入 json 結構中。

- CREATE\_LISTING <username> <title> <description> <price> <category>

和 REGISTER 邏輯相同，新建一個 listing 資料結構並將其視為一個 dictionary 存入 json 結構中。同時檢查 category 是否存在，若無則新建一個 category，有則維護其 listing\_ids 和 listing\_num 兩變數。

listing\_ids = [] 屬於該 category 下的 listing id

listing\_num 該 category 下的 listing id 數量

- DELETE\_LISTING <username> <listing\_id>

利用 del 刪除對應 listing\_id 的資料結構。

維護該 listing\_id 所屬的 category。

- GET\_LISTING <username> <listing\_id>

獲取該 listing\_id 於資料庫中所有的資訊

- GET\_CATEGORY <username> <category>

```
def get_category(self, username, category):
    self.data = self.db.load()
    u = username.lower()
    if u not in self.data["users"]:
        return "Error - unknown user"
    listings = [l for l in self.data["listings"].values() if l["category"] == category]
    if not listings:
        return "Error - category not found"
    sorted_listings = sorted(listings, key=lambda l: l["created_at"], reverse=True)
    return "\n".join(
        f"{l['title']}|{l['description']}|{l['price']}|{l['created_at']}|{l['category']}|{l['username']}" for l in sorted_listings
    )
```

獲取該 category 下所有 listing 並利用 listing 的建立時間排序。

建立時間採用 utc 時區。

- GET\_TOP\_CATEGORY <username>

獲取最多 listing\_num 的 category，並且依照首字母順序輸出。

## 數據儲存結構

```
from datetime import datetime, timezone

class User:
    def __init__(self, username):
        self.username = username.lower()

class Listing:
    def __init__(self, username, title, description, price, category, listing_id):
        self.listing_id = listing_id
        self.title = title
        self.description = description
        self.price = price
        self.username = username.lower()
        self.category = category
        self.created_at = datetime.now(timezone.utc).strftime('%Y-%m-%d %H:%M:%S')

class Category:
    def __init__(self, name, listing_id):
        self.name = name
        self.listing_num = 1
        self.listing_ids = [listing_id]
```

- 附圖為 model.py 下的各個資料結構
- users：用戶清單
- listings：物品資料
- categories：物品類別中的列表
- parameters: 資料庫中的係數，例如: 下一個 listing 的編號 (next\_listing\_id)

### 3. 執行環境

使用 Python 3.11.4

import: datetime, shlex, json, pathlib (standard library 中都有)

執行方式: 於 git bash 中輸入 bash run.sh

資料庫存於 data 資料夾中，名為 database.json

如果要重置資料庫，可以直接刪除該檔案