

## 資料庫管理（二）期末專題

### 一、專題簡介

在這次期末專題中，我開發了一個可以簡單的記帳應用程式，功能包括開始記帳、更改紀錄、刪除紀錄、搜尋紀錄、分析帳務。

### 二、開發方法

開發語言	Python 3.9
開發環境	Ubuntu20.04
前端介面	Tkinter
資料庫	MongoDB

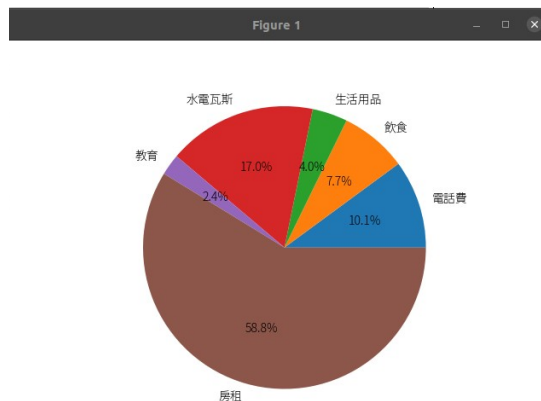
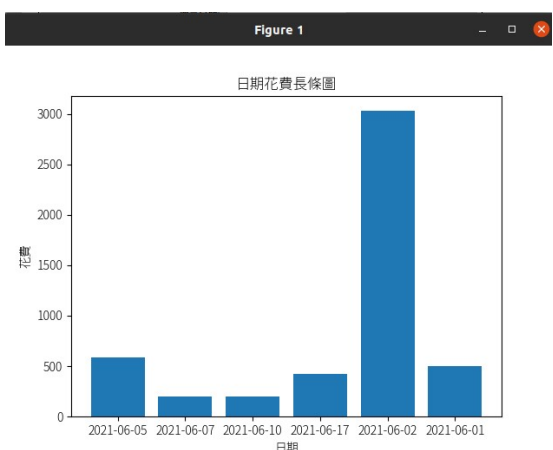
### 三、程式功能介紹

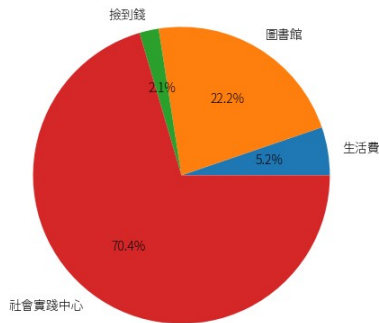
說明	主畫面	記帳畫面
執行畫面		
功能	功能選單	利用 comboBox、entry 進行輸入

說明	記帳成功畫面	更新畫面
執行畫面		
功能	利用 messagebox 提示記帳成功	先進行 query，篩選要更新的内容

說明	搜尋結果	更改頁面
執行畫面		
功能	兩擊左鍵列，可進入更改頁面	會將預設原本的容顯示在輸入欄裡

說明	刪除頁面	分析頁面
執行畫面		
功能	鎖定輸入欄	可以分析如上按鈕的圖表

說明	花費圓餅圖	花費長條圖
執行畫面		
功能	可以依日期分析花費圓餅圖	可以依日期分析花費長條圖

說明	收入圓餅圖	
執行畫面		
功能	可以依日期分析收入圓餅圖	

#### 四、程式碼說明（僅說明比較重要的程式碼）

##### • 使用套件

```

1  import tkinter as tk
2  import datetime
3  import tkinter.ttk as ttk
4  from bson.objectid import ObjectId
5  import pymongo
6  from tkinter import messagebox
7  import pandas as pd
8  import matplotlib.pyplot as plt
9  import numpy as np

```

##### • 連結 mongoDB、建立 collection 等

```

12 #建立連線
13 client = pymongo.MongoClient(
14     "mongodb+srv://B10723059:lisji8380@cluster0.r36jl.mongodb.net/TRY?retryWrites=true&w=majority"
15 )
16
17 #創建名為bookkeep的DB，及record的collection
18 db = client.bookkeep
19 mydb = db.record

```

##### • 建立 GUI 的底板

```

class basedesk():
    def __init__(self, master):
        self.root = master
        self.root.config()
        self.root.title('資料庫II 期末')
        self.root.geometry('600x300')

        initlface(self.root)

```

- **Insert 相關程式碼**

```
def change2(self,):
    data = {
        "date" : self.date_entry.get(),
        "class" : self.class_entry.get(),
        "item" : self.item_entry.get(),
        "cost" : int(self.cost_entry.get()),
        "income" : int(self.income_entry.get())
    }
    mydb.insert_one(data)
    messagebox.showinfo("提示", "紀錄成功")
    self.face1.destroy()
    initlface(self.master)
```

- **update 相關程式碼**

```
def change3(self,):
    id = self.id_entry.get()
    date = self.date_entry.get()
    class_1 = self.class_entry.get()
    item = self.item_entry.get()
    cost = self.cost_entry.get()
    income = self.income_entry.get()
    mydb.update_one({'_id':ObjectId(id)},{'$set':{'date':date,'class':class_1,'item':item,'cost':int(cost),'income':int(income)}})
    messagebox.showinfo("提示", "更新成功")
    self.face2.destroy()
    initlface(self.master)
```

- **delete 相關程式碼**

```
def change3(self,):
    id = self.id_entry.get()

    mydb.delete_one({'_id':ObjectId(id)})
    messagebox.showinfo("提示", "刪除成功")
    self.face2.destroy()
    initlface(self.master)
```

- **query 相關程式碼**

```
for x in mydb.find({"date":{"$regex":"info"}}):
    data = list(x.values())
    data.pop(0)
    tree.insert('', 'end', values=data)
tree.grid()
tree.grid(row=1, column=0)
```

- 利用 aggregate 進行資料整理並繪製圖表

```
def change2(self,):
    data = self.sea_entry.get()
    agg_result= mydb.aggregate(
    [
        {"$match" : {"date":{"$regex":data}}},
        {"$group" :
            {"_id" : "$class",
             "total" : {"$sum" : "$cost"}}
        }
    ])
    class_1 = []
    size = []
    for i in agg_result:
        if i['total'] > 0:
            class_1.append(i['_id'])
            size.append(i['total'])
    plt.rcParams['font.sans-serif'] = ['Taipei Sans TC Beta']
    plt.pie(size, labels = class_1, autopct='%1.1f%%')
    plt.axis('equal')
    plt.show()
```

=====詳細的程式碼附件將會附上=====

謝謝老師，老師辛苦了！！