**3.2程式架構 :**

**3.2.1 搜尋及查詢資料庫(演算法)**

a.系統邏輯

1.前端協作

2.後端協作

main() 中執行的順序 :

1.呼叫爬蟲

2.確認是否有新商品

3.確認是否有新需求

4.搜尋商品是否有符合需求的，有則寄信

5.刪除到期需求(目前設定期限為兩周)

由於商品價格不會每天都改變，有的甚至一兩個月才會有改變，所以爬蟲這部分是採取 windows 工作排程器，定期呼叫程式去執行(目前設定一天一次)

b.搜尋方法

1.字串分析(英文數字)

目前可搜尋商品分為三類 : 手機、筆電、平板，假設今天要搜尋的商品名稱為 iphone7 plus，在 iphone 跟 7 中間訂閱者輸入時並沒有空格，會造成後端在資料庫內搜尋商品時，會因為空格的關係，明明要比對到的產品，但因為字串比對不到，造成比對不精確，所以遇到數字的部份我們會進行切割，像是上面的例子會被切成 iphone、7 以及 plus 這三個字，在資料庫中商品字串要有符合這三個才會被比對到，英文的部分就是利用空格去做切割。

2.字串分析(中文)

中文的部分，因為要讓系統知道什麼樣單字、句字才會符合語意，所以我們利用「結巴分詞」(jieba)。

結巴分詞是目前大家認為做最好的 python 中文分詞套件，支援的模式有三種 : 「精確模式」、「全模式」以及「搜尋引擎模式」，精確模式顧名思義就是做最精確的切開，而全模式則是速度較快但不能解決歧義，搜尋引擎模式則是在精確的基礎上，對較長的詞語再次切分，提高召回率。

結巴有自己的字典(dict)，裡面有上萬的詞語，包含詞彙出現的次數和詞性，使用者也可以依照自己的需求，將裡面沒有的詞彙加入其中，或是改變出現次數等等，因為結巴分詞屬於概率語言模型，這個模型的任務就是在全切分所得的結果，求某個切分方案 S，使 P(S) 最大。

**3.2.2 寄信與資料庫比對**

1.資料刪除

購物網站的商品如果下架，必須要在我們的資料庫商除這筆資料，避免訂閱者的需求搜尋到一筆不存在的資料。

2.資料更新

商品價格如果有變動，我們並不會建立新的資料，而是更新資料庫舊有的資料。

3.需求到期刪除

需求的搜尋期限為兩周，原本採用 (現在時間 - 發起需求的時間)是否等於14，這樣的方式去做判斷，不過這樣要多做一步運算的步驟，雖然影響不是很大但多少還是會影響程式效率，所以最後就是在需求建立同時設定到期時間，只要比對到期時間是否等於現在時間。

4.需求符合寄信

import python 原有套件 smtplib 來做寄信功能，符合的商品可能不只一件，所以會先把搜尋到的商品放入陣列，待全部比對完成後再做寄信，寄信內容為商品名稱、價錢以及網址，使用 gmail 來寄信。

**3.2.3 網路爬蟲演算法(露天)**

a.總流程及比對資料庫

1. 程式最開始會先初始化資料庫，如果是第一次執行則會建立資料庫以及所需要的 table(手機、筆電、平板)，裡面的欄位分別為 : title, price website, time, flag，前三個都是商品的資訊，time 則是爬到商品的時間，flag 是用來分辨是哪一個購物網站抓取下來的資料。
2. 接著執行 parse() ，parse function 一開始會先執行 initFunc，這個 function 會把舊的資料抓出來，之後在爬蟲時會需要他們去比對是否已經存在，之後因為這邊使用 selenium，所以就會打開瀏覽器開始爬蟲。
3. 爬蟲找每一筆資料時，若是第一次執行則會無條件加入資料庫，否則 會先和網址去比對是否有一樣的，如果沒有表示是新上架商品，直接存入資料庫的 new table，如果有再去比對價錢是否一樣，若一樣代表商品維持原樣且存在，若不一樣則代表價錢更改過，直接更新資料庫資料。
4. 判斷商品還在不在的方式是用一個變數 count ，只要資料庫 old table 的資料有在這次爬蟲中被比對到，count 就設為1，否則維持0，在爬蟲結束之後會把 count = 0 的資料刪除，代表此商品已不再網頁上，這樣可以避免訂閱者需求找到已被刪除的頁面。

b.爬蟲過程

1. parse() 會從 urls 陣列中取出欲爬取的網址，並用 for 迴圈去一一輸入瀏覽器。
2. 因為利用 selenium 模擬登入，所以會打開瀏覽器到接收到的網址，再使用 BeautifulSoup 套件去解析爬到的 html，方便我們去擷取需要的資訊。
3. 每一個網頁的配置不一定一樣，而這次爬的 pchome、露天、樂天都有些許不一樣，所以用不同的方式個別寫了一個爬蟲。
4. 露天的瀏覽方式為換頁式，意即網址最後的數字為頁碼，所以只要將那個數字設定好範圍，利用迴圈下去跑，即可自己設定要爬的頁數。