**제목: caffeine tracker**

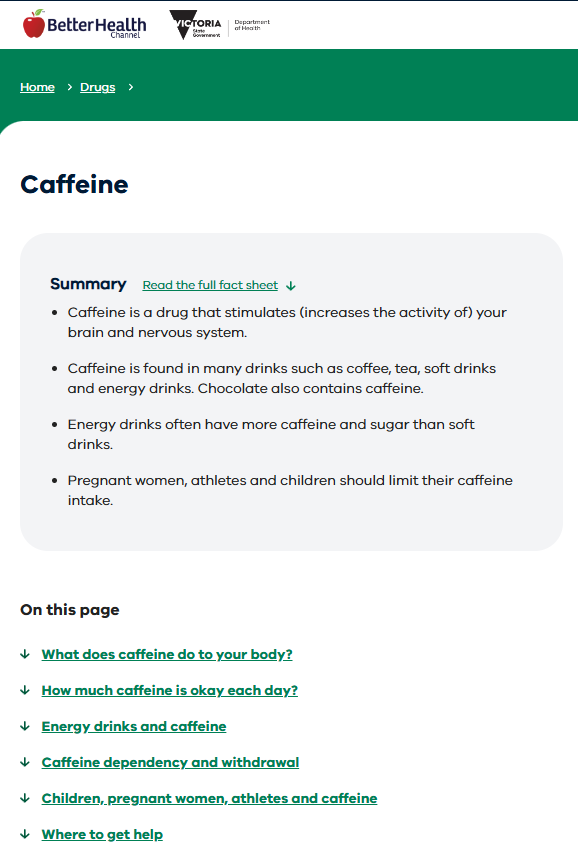
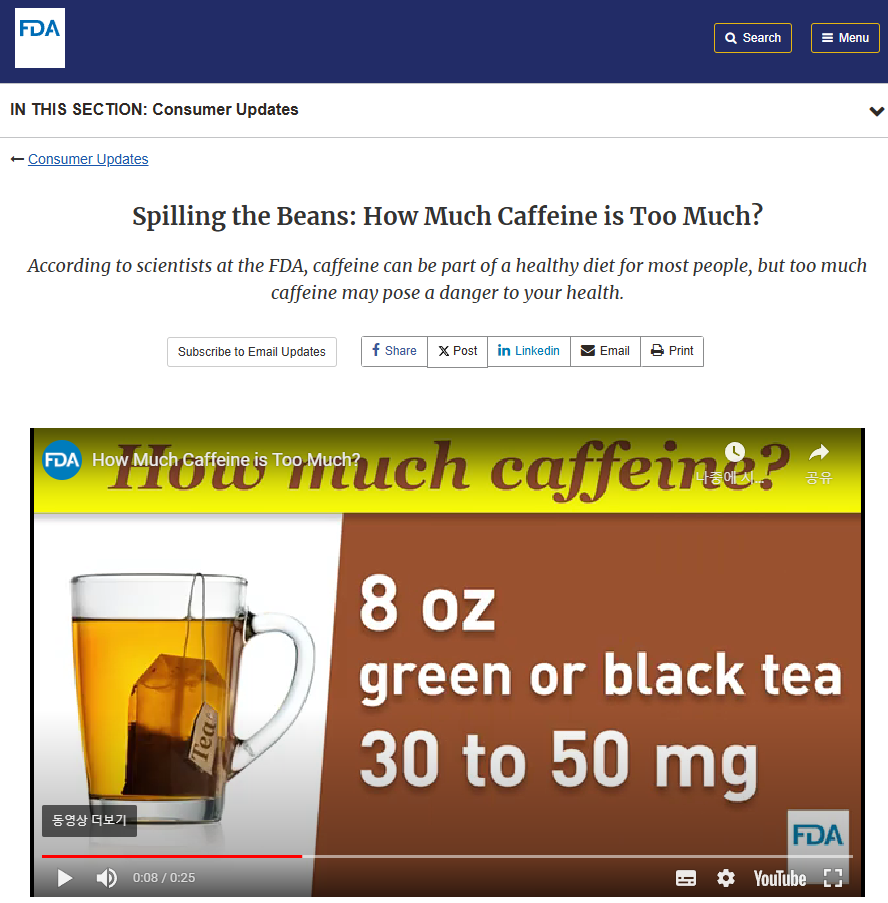
**By 이규빈 peter**

**목차**

* **서론**
  + **동기** : 왜 만들게 되었는가? / 개인적 근거와 객관적 근거
  + **목적** : 이루고 싶은것!
* **본론**
  + **선행조사(자료 조사)**
  + 카페인이 무엇인가?
  + 권장량이 있는가?
    - 개인별 권장량은 어떻게 다른가?
    - 많이 마시면 왜 안좋은가?
  + 음식별 카페인 함량?
* 다른 작품
  + **설계방법(어떻게 프로그램을 만들것인가?)**
    - PyQt5란?
    - Qt Designer란?
    - UI 디자인
    - 동작 Flow Chart
    - 코드 설명
* **결론**
  + 결과물 : 최종 코드와 작품 캡쳐
  + 느낀점 : 부족한점, 개선했으면 좋겠는점. 깨달은점
* **참고문헌**
  + 링크
    - <https://blog.naver.com/niar_/220664635104>
    - <https://play.google.com/store/apps/details?id=com.dci.dev.caffeinme&hl=ko>
    - <https://apps.apple.com/us/app/hicoffee-caffeine-tracker/id1507361706>
    - <https://recaf.app/>
    - <https://caffeine-app-track-caffeine.softonic.kr/iphone>
    - [Caffeine - Better Health Channel](https://www.betterhealth.vic.gov.au/health/healthyliving/caffeine)
    - [How Much Caffeine in a Cup of Coffee? A Detailed Guide (healthline.com)](https://www.healthline.com/nutrition/how-much-caffeine-in-coffee)
    - [Is decaf coffee bad for you? Caffeine content and health benefits (medicalnewstoday.com)](https://www.medicalnewstoday.com/articles/325502#bad-for-health)
    - [Spilling the Beans: How Much Caffeine is Too Much? | FDA](https://www.fda.gov/consumers/consumer-updates/spilling-beans-how-much-caffeine-too-much)
    - <https://www.hopkinsmedicine.org/health/wellness-and-prevention/is-coffee-bad-for-kids>
  + 논문 등

This program I have developed is a tracker to keep track of daily intake of caffeine, and make sure the user does not consume caffeine excessively over their recommended amount. The reason why I decided to develop this specific program is because I have several people around me who enjoy drinking coffee or sometimes excessively. in this modern world, I realised that it is not only coffee which contain caffeine, and I thought that several people could be consuming more caffeine than they think they are, through various foods and drinks such as chocolate, energy drinks, tea, and even some water which are sold in grocery markets. After finding out about the seriousness of the risks of caffeine through reading several scientific articles about the risks and disadvantages of caffeine, I decided to develop this program in which the user can record their caffeine consumption and notifies the user if they have consumed excessive amounts of caffeine. my goal which i desire to achieve by developing this program is to make people stay healthy, and stay away from the risks of excessive caffeine

Caffeine is a chemical which acts as a stimulant when consumed in our body. It can speed up the activities in the brain, increase the circulation of chemicals such as adrenaline in the body. Due to this, consuming caffeine can make people who consume it feel awake and refreshed and help them to work more efficiently. However, caffeine also has some serious health issues. if excessively consumed, it could cause several problems in the person’s body, such as dehydration, dizziness, headaches, anxiety, trembling hands, difficulty sleeping, and several more issues. According to the US food and drug administration, it is recommended that a healthy adult should consume no more caffeine than 5 milligrams per kilogram of body weight daily, which is around 400mg of caffeine per day for an 80 kg adult. Caffeine is not only found in coffee but in foods and drinks which we can easily eat or drink in daily life, such as energy drinks, tea, chocolate, and coke.

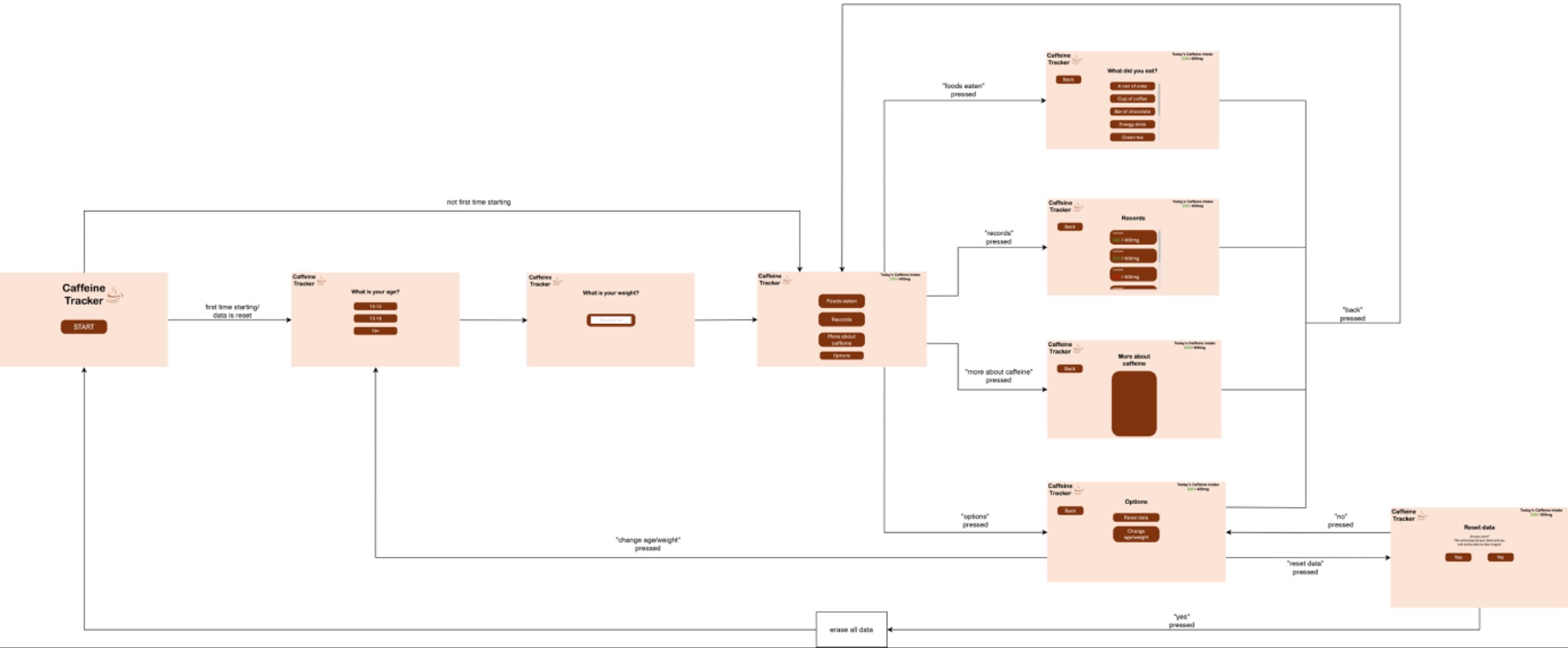


[The recommended amount of caffeine, FDA](https://www.fda.gov/consumers/consumer-updates/spilling-beans-how-much-caffeine-too-much) [The risks and the effects of caffeine](https://www.betterhealth.vic.gov.au/health/healthyliving/caffeine)

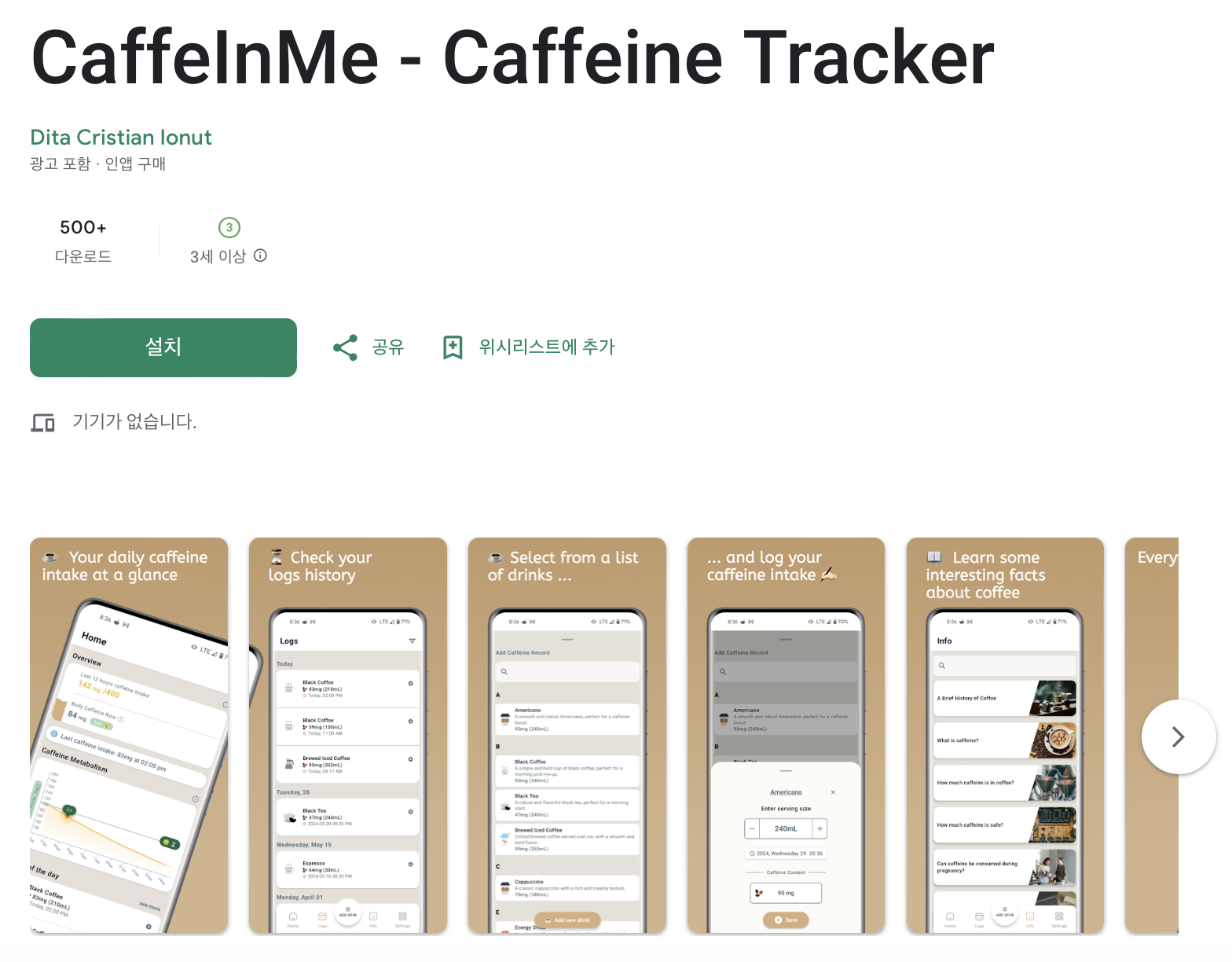
The program will allow the user to input the foods or drinks they have consumed and calculate the total amount of caffeine consumed throughout the day. For example, when a person inputs that they have drunk a cup of coffee, the program will put in the data for the average amount of caffeine in a cup of coffee. as there are different recommended maximum amounts of caffeine, the program will get information about the user’s age and weight and calculate the maximum recommended caffeine consumption and notify when the amount of the caffeine consumed exceeds the recommended amount. The program will also record the amount of caffeine consumed during the day every day, and the user will be able to check the amount of caffeine consumed on other days. Also, the user will be able to know additional information about caffeine to motivate the user’s caffeine consumption reduction. In case the personal information such as age and weight changes or the user of the program changes, the program will as well offer an option to change personal data and reset data. The preview design of the program can be seen through the link below.

<https://docs.google.com/presentation/d/1wUksTPMWEhlcEw-YuX5kQ26Nvei7VHHuDLr5EqdCH_M/edit?usp=sharing>

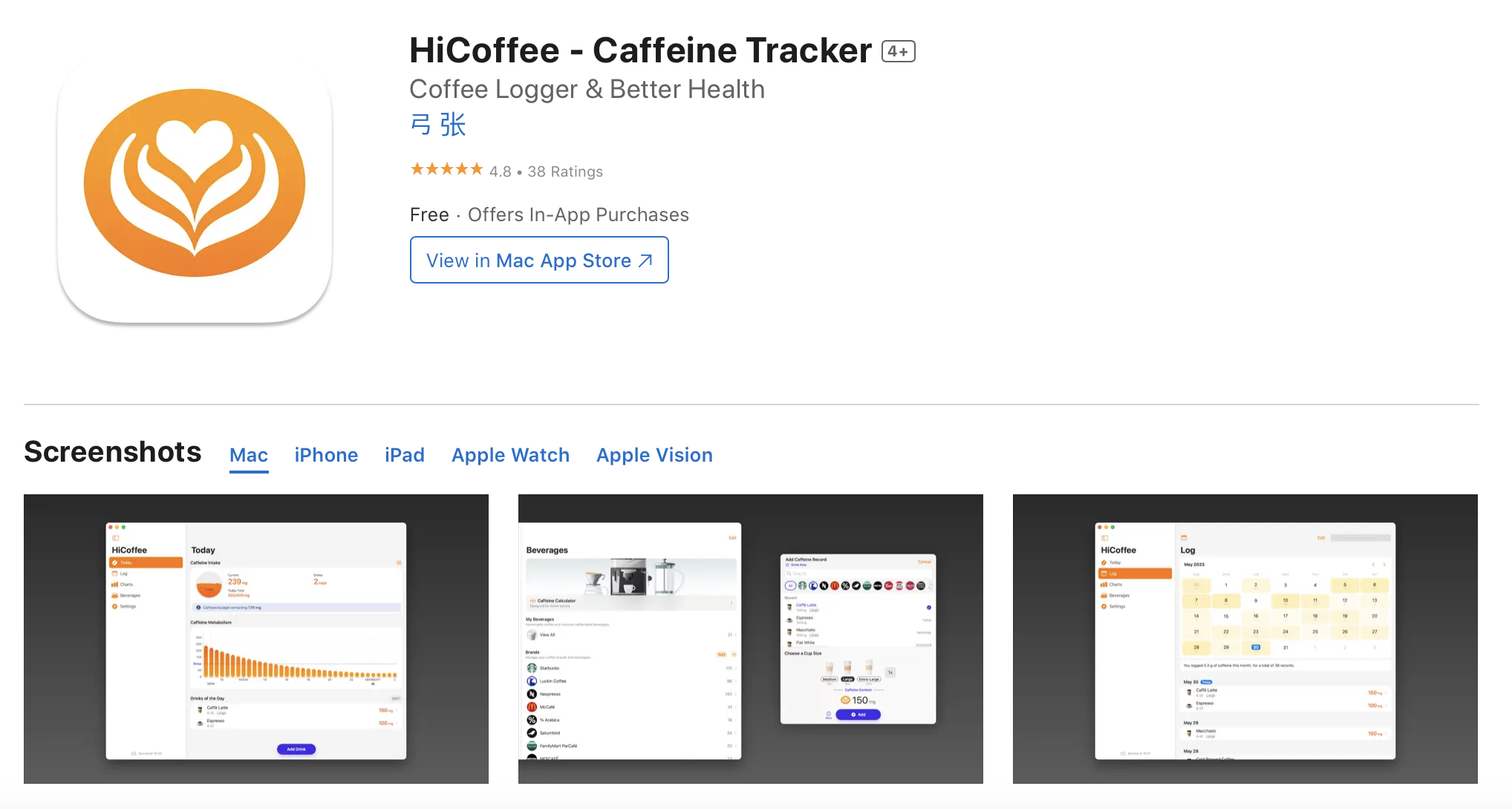
this is a flowchart to show how the program should work



There have been some similar programs to mine, and some of them are CaffeInMe, HiCoffee and RECaf, which are softwares/applications which the program calculates the user’s recommended maximum caffeine amount and records the amount of caffeine consumed by the user during the day.



[CaffeInMe](https://play.google.com/store/apps/details?id=com.dci.dev.caffeinme&hl=ko)

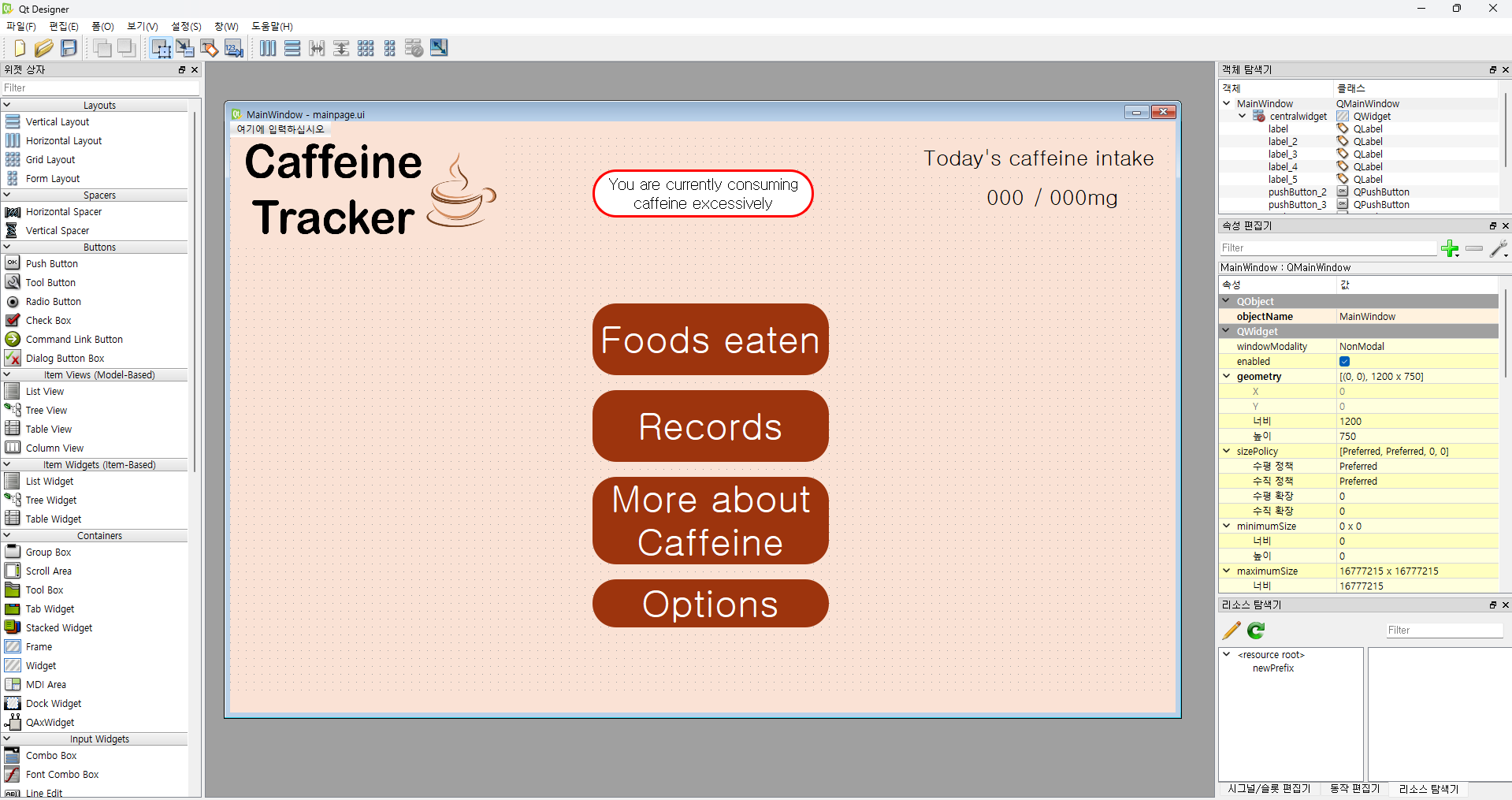


[HiCoffee](https://apps.apple.com/us/app/hicoffee-caffeine-tracker/id1507361706)



[RECaf](https://recaf.app)

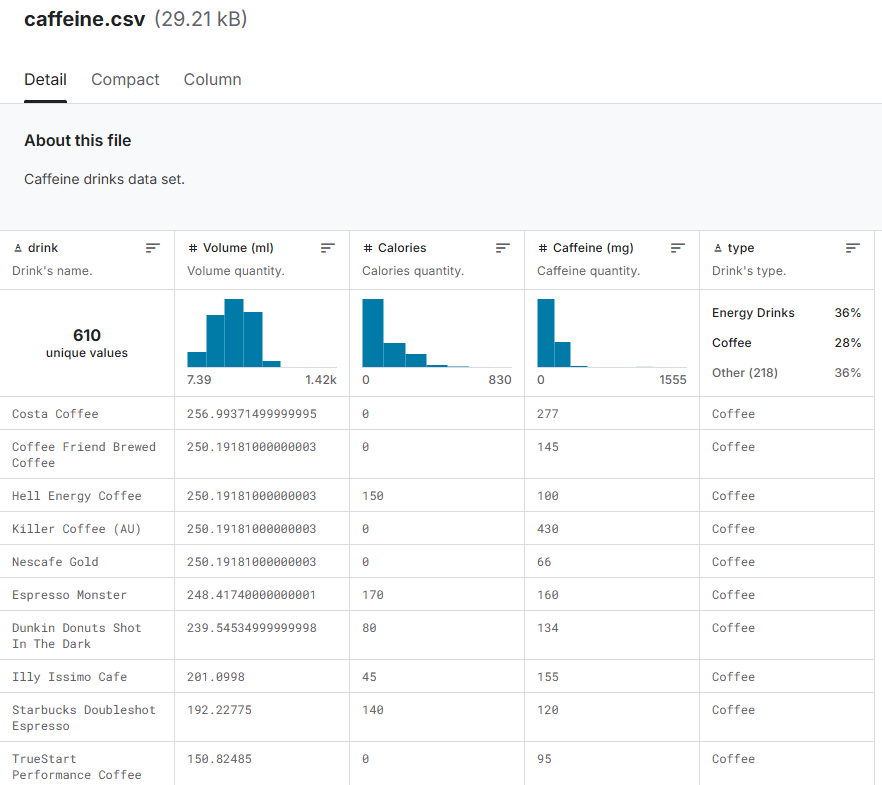
When I started making the program, I firstly designed the GUI for the program, which was designed using a GUI toolkit called “Qt designer” which allows people to freely make the user interface for their program. I tried to make the program like the preview design, with the same colour, same logo, and same buttons. However, there were also some additions to the program, which i added several more options which the user is able to choose what type of drink they have consumed and allow them to input the specific product they have consumed by adding a function to search for the specific drink, making the program more accurate at calculating the user’s caffeine consumption.



[Qt designer](https://doc.qt.io/qt-6/qtdesigner-manual.html)

After I was done with designing the user interface, I needed specific information about various drinks and the amount of caffeine the drink contains. For this, I found a dataset from kaggle.com, which is a website which has a collection of various datasets. The dataset I found was a long list of numerous drinks with the information specific amount of caffeine contained in the drink.

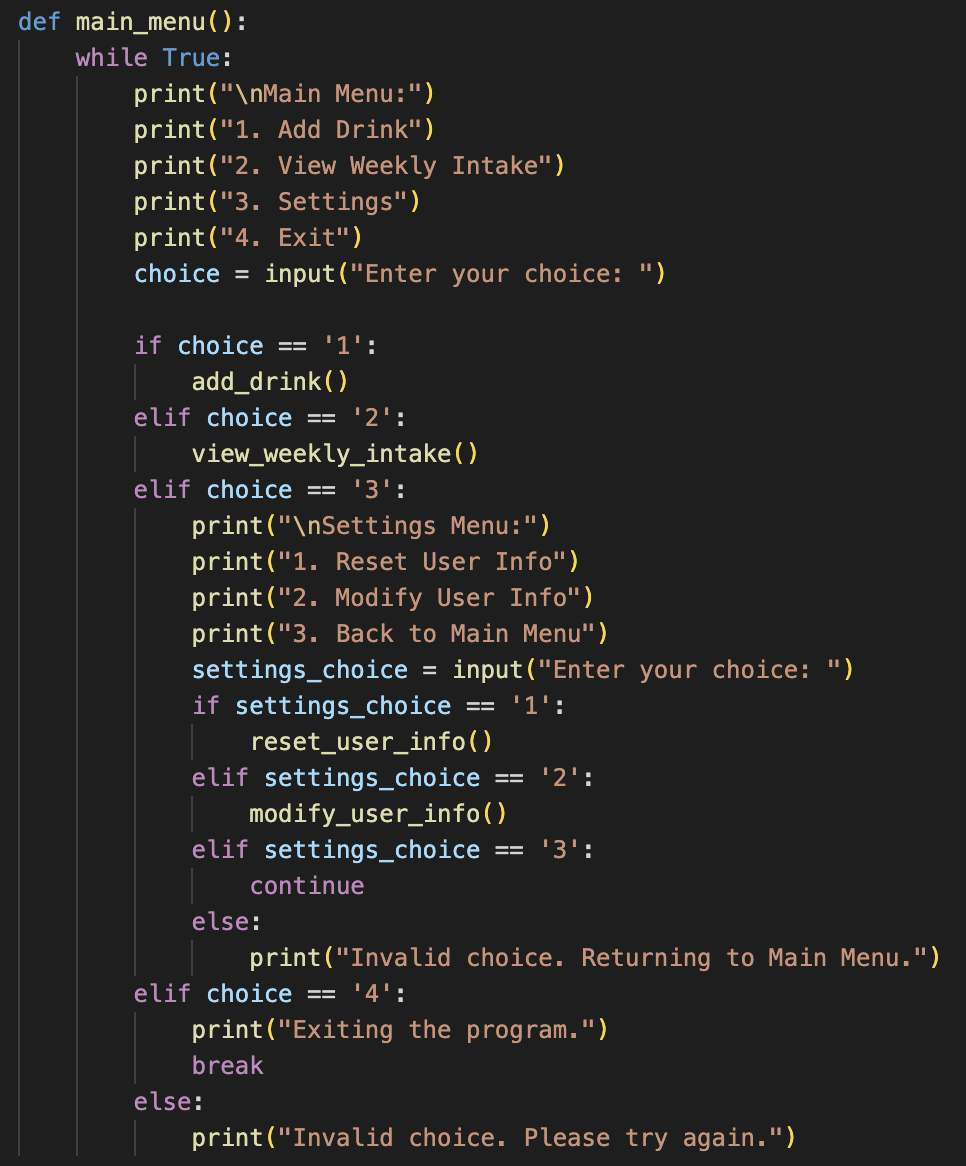
this is the dataset which i used for the program:



<https://www.kaggle.com/datasets/heitornunes/caffeine-content-of-drinks>

Before I started on developing the actual program with functional user interfaces, I decided to make the program only in python script which runs on the terminal. The first part I started making was the main menu for the program in which the user can access the four choices, which are [foods eaten], [records], [more about caffeine], and [options]. In the program it was written as [add drink], [view weekly intake], [settings] and [exit]. There is not an option to see the information about caffeine, but it still is added in the final program. The reason why [foods eaten] was changed to [add drink] was because I decided to change the program to only drinks, not food, as I was unable to find the dataset with both food and drinks. In this program which runs on terminal, the user can access the four options by pressing the numbers on their keyboard, which pressing 1 would lead the user to [add drinks], pressing 2 would lead the user to [view weekly intake], pressing 3 would lead the user to [settings] and pressing 4 would quit the program.

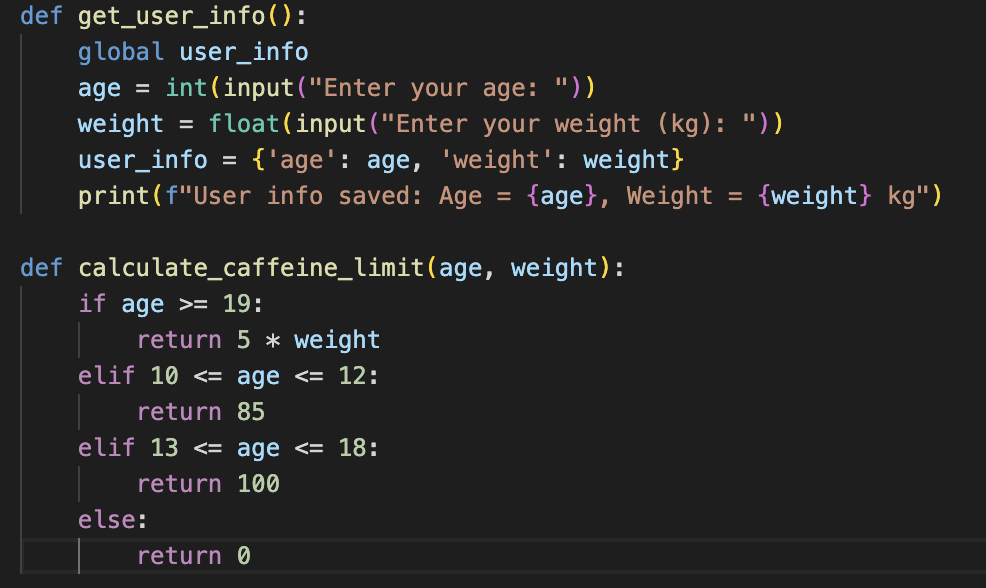
this is the code for the main menu



There are several functions written for each task. For example, when the user chooses to add a drink, there is a function to take in the input about the drink, another function to search the drink from the database table, and another function to calculate the caffeine amount and add it to the total caffeine consumption. the process is as follows:

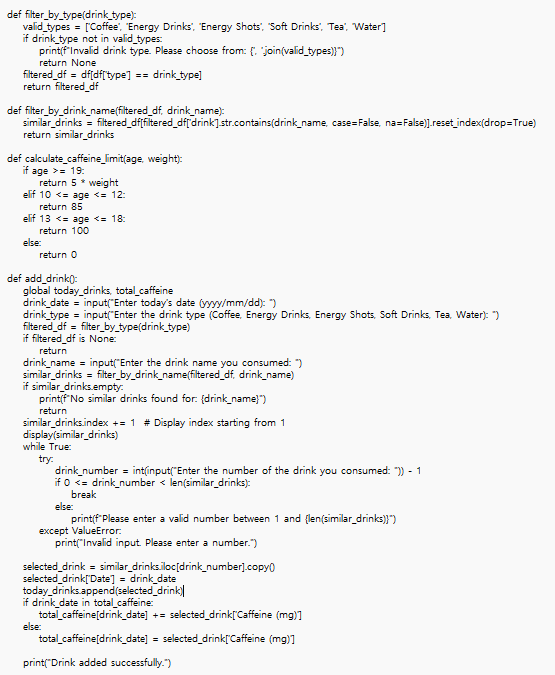
Upon startup of the program, the program will first ask the user about the user’s age and weight through the function named get\_user\_info(). With the information, the function calculate\_caffeine\_limit() would calculate the maximum amount of caffeine. If the user’s age is between 10 and 12, the limit would be 85mg, if the user’s age is between 13 and 18, the limit would be 100mg, and if the user’s age is over 19, the maximum limit would be calculated by multiplying the user’s weight by 5. There is no calculation for ages below 11, as I do not expect a person below 11 would drink caffeine excessively

These are the functions used in the process



First, the user would type 1 on their keyboard to access the input drink menu. The program would first ask the user about the date, as the user may have forgotten to input their caffeine consumption and want to add it to the records. Next, the program will ask the user about what type of drink the user has consumed, and the options are: coffee, energy drinks, energy shots, soft drinks, tea and water. Next, the program will ask the user the name of the drink. When the user inputs the name of the drink, the program will find all relative drinks from the database according to the user’s inputs. From this, the program will calculate the amount of caffeine consumed from the drink and add it to today’s caffeine consumption.

these are the functions used in inputting the drink information



For the records, a function named view\_weekly\_intake() would get out the results from previous days and show them to the user. The records are stored as an external file, in excel table format. When the user chooses to view the function, the user will be able to see their caffeine consumption for the previous 7 days, and the data which are older than 7 days will be deleted. in the data, the user will be able to see the date, their caffeine consumption and the comparison with their maximum recommended caffeine amount.

when the user accesses the options menu, they will be offered with two choices, to edit the user information or reset the data. As written above, choosing ‘reset data’ will erase all records and user information(weight and age), while choosing ‘edit user information’ will erase only user information not records, and the user will have to fill in their age and weight information again.

In case the user happens to accidentally choose the wrong option of the program, there is always a ‘back’ option, which the user is able to return to the main menu whenever they want to.

Writing the code was not as easy as I thought it would be. Despite the simple design, several parts such as using new modules and making an option to search for a specific drink from the database and making the program to be able to record the user’s caffeine intake, I had to get some help from my teacher, but I was able to do it myself after.

When I was done with the code, I used a module named “PyQt5”, which is a python module that allows me to add the UI which I have previously designed using Qt designer onto the program which runs on the terminal. This is done by linking the buttons of the ui to the codes. For example, when the ‘start’ button is pressed at the start of the program, the program will run an according function to the button, which will be led to inputting the user’s age and weight.

However, there have been parts which I had to go differently from the plan as I added the ui onto the program. One of them was the function to search for the specific drink, which I was unable to add due to the size of the database, which slowed down the program. Therefore, I decided to make the program to be able to choose the food/ drink type, choose the food/ drink of the type of the user has consumed, and input the amount the user has consumed. I will be adding the function to search for the specific drink from the database in the future.

This experience of making the program felt meaningful to me. Since I only used to write algorithms to solve online questions, writing a program was a new experience. Although writing the code itself did not require any new knowledge about the language except for modules, it was not as easy as I thought. Using functions more frequently and making lines shorter to make the program more efficient. Also, from making the program, I learnt and understood more about the importance of the user experience, as well as the need to make the user interface better and easier to use. Overall, I was able to learn more about software development and take a lot from this experience.