University of St.Gallen

My Medibox Report:

A case study by Valantic

Group 4:

Valeria Baldini (20- 602-829)

Benedetta Bona (20-601-845)

Irena Marina (18-614-826)

Francesco Salerno (18-617-704)

Hyeon-Ji Yun (17-334-525)

10th September 2021

Background

When visiting the doctor and getting a new prescription, most of the time we need to go back home to check which medicines we already have and if they are expired, before being able to go to the pharmacy. As you may have experienced, having a clear glance of which medicines we have in our cabinet can then be very useful. By implementing an app that scans the QR code of the medicines we have at home and keeps track of the expiration date, we would be able to always control what we need to buy and what not. Therefore, we would save both time and money.

Our solution

As a rough implementation of what could then become an app, we developed an interactive python script that takes a medicine code as an input and creates a csv file where all the medicines present in our cabinet are stored. The file provides the following information: "Product ID", "Drug Name", "Purchase Date", and "Expiration Date". When visiting the doctor and getting a new prescription, we can immediately check what we need and what we have in terms of medicines. If a new drug needs to be bought, we are redirected to an online pharmacy. In case we decide to buy the product, that is added to our cabinet. If we already had a medicine but it is now expired, we get an indication on how to dispose of it safely, and we are kindly invited to purchase a new one. Even here, once the purchase is successful, the purchase date and the expiration date are updated in the csv file. When a medicine in our cabinet expires, we receive both a notification in the app and an email, where we are invited to buy it again while properly disposing of the expired one.

Technical Description

To better show how we technically developed the project, we will divide the explanation into three blocks:

1) For building the cabinet, we adopted the Object Orienting Programming (OOP) concept. Indeed, we created a class called Medicine, showing different methods on how to get the medicine code from the user, the associated name from an international drug database, and a randomized purchase and expiration date. The purchase date was randomized by assuming the medicine was bought not more than

one year and a half ago. The expiration date is assumed to be exactly one year after the purchase date, as the average life span of a medicine is approximal twelve months. After creating the class, we implemented a function (i.e., add_medicine()) that allows users to add all the medicines in the cabinet and return a list of objects. To proceed on the next step, we also defined another function (i.e., update_table()) which converts the list of objects into a csv file.

2) When visiting the doctor, as explained above, we programmed the script to handle many different situations. To do so, we used mainly two of the concepts presented in class: while & for loops and the Pandas software library. The loops helped us designing all the possible potential situations a user may encounter. Instead, the Pandas library gave us the possibility to better access information in the csv file offering higher flexibility and coherence than a list.
3) As a final step, we built a mechanism which sends a notification to the user as soon as the medicine in the cabinet expires. To do so, we used for and while loops to simulate the passing of time.

Future Developments

As we mainly focused on implementing a robust and comprehensive script, an interesting future development would consist in creating a real app out of this code. Indeed, starting from our code, we could add the scan function to recognize the medicine from a QR code and simplify the script by avoiding if conditions created just for misspelling errors.

Group Reflection

Confronting ourselves in working on such a complex project with only a few days of Python tutoring was challenging. However, the concepts we studied in class enabled us to successfully deliver a script, which despite not being an app, still comprehends and solves all the possible situations that a user may face. Indeed, what we found difficult while coding was to foresee misspelling and mistyping errors and provide a solution that could still preserve the logic of the script.