Group 6 -- Vaccination Tracking System

21107569d, ZHANG Haolin
21099191d, WANG Yihe
21102833d, ZENG Yuejia
21095134d, DUONG VanNam

Problem Description

This project uses student id to track vaccine status, supports multiple administrators, supports multiple client queries, and single client updates data. For multiple clients, this project involves a php server to distribute the data. For ensure security, this project uses encryption algorithms for administrators' passwords and server requests. At the same time, in order to increase the efficiency of use, OCR is introduced to identify student id

Data Abstraction

In this project, the built-in dictionary will be used as the foundation. There are two types of dictionary. First stores vaccine data whose key is id (String), and value is status (True, False). Second stores administrator account and password whose key is id (string) and encrypted password (String).

Install Menu

Warning: Must use Python3.8 and below

1. Install Tesserocr Engine (please follow https://tesseract-ocr.github.io/tessdoc/Installation.html)
2. Install Python library through pip

```
pip install -r requirements.txt
```

3. Start by looking at the help message

```
python main.py -h
```

(Any questions are welcomed to ask me through email: me@hlz.ink)

Design & Comment

OCR & Data Transfer (ZHANG Haolin)

I would like to share about codes in OCR and DataTransfer. In the OCR part, I did some research and found out that there is a handy OCR engine called Tesserocr, which could load local pictures and detect the student id. So, the point is getting images from the camera. I find a project on Github (https://github.com/nathanaday/RealTime-OCR) which contains several handy API for me to use, which basically a wrapper of CV2 library. Combine them together and the OCR part is done. In the DataTransfer part, the first thing comes to my mind is concurrency. And I realized separate the file and do the enquiry locally is the best to way. Thus, simply get and post request would fulfill our target. In addition, a time-based token is needed to verify the client identity. With these ideas, the server which is written by PHP and clients codes are finished.

Console (WANG Yihe)

We chose the second project, which is a monitoring program for vaccination information. The part I was responsible for is use console of interactive mode to form interaction with users, which allow users to choose the query mode to check or change their vaccination status and form a complete python project. In command line mode, we can either execute python to enter the python interactive environment, or we can execute a python file. We chose interactive mode here because the python interactive environment automatically prints out the result of each line of python code but running python code directly does not.

1) I start by writing out the rough template and as I finish my part, put the code my group mates have written into a folder and open it in VSCODE. And then we imported the OCR, DataHandler and GUI code at the beginning.

2) Then, we defined a function named typeQuery to allow the user to enter their student studentID and then define a function call the getStastus function to look up their vaccination status in the database.There is also an OCR pattern which does the same thing as the previous function and returns the result. As vaccinated, unvaccinated or not yet recorded. The user will be in password entry mode followed by a pop-up screen where they can enter their password directly into the console (not visual) and if they choose camera mode, they can read your student ID number by capturing the ID card in front of the camera.

3) Next, the user can accept instruction that how and what to enter numbers. Besides, we designed three modes to get the student card, a direct input mode, one to get it via the OCR mode and a display analysis mode. I designed a function to allow the user to select their preferred mode by entering a number. Once the studentID was obtained, the code written by the rest of my group could be used to obtain the user's vaccination status.

4) From there, we could ask the user to enter their password and student id to log into the system and if entered correctly, the user could further view and update their vaccination status by entering numbers.

Finally, we defined a CMD function that was designed to wrap these fragmented functions into a unified function that would call the code written by the rest of our group by entering 1, 2 and 3 and selecting the mode they needed.

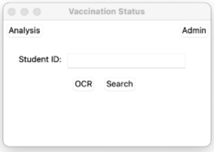
If the code does not execute properly due to network or other uncontrollable reasons, we have provided solutions to exit the application in almost every code block. The user can exit the application by typing "-1".

My part was to form interactions with users using the console interaction mode and combine all of everyone's code blocks together and use the functionality they had written combined with my own code to form a complete, working program. We were assigned different tasks once we were in a group and thanks to the cooperation and help of Haolin. Besides, It was very exciting to see that the program we wrote worked successfully when we worked together to see the final result. I also learned a lot of new things in the process and it was a very happy experience to work with everyone in this group.

GUI (ZENG Yuejia)

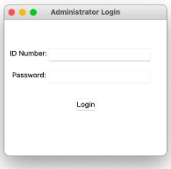
GUI is the graphical user interface, which can be used for the users to interact with electronic devices through graphical icons and audio indicator. Its goal is to enhance the efficiency for the underlying logical design of a stored program. It has four traits, which are human-computer interaction, better visualization for users, practical, and technical. As one of the most helpful toolkit, Tkinter is used in this assignment for creating GUI. The overall code can be divided into three parts. Firstly, import required library; secondly, define functions; thirdly, create interfaces. Due to the requirement of assignment 3, we have to have boxes for user input and buttons for save information or go on to the next page. In this code, GUI were used mainly for four windows, which are administrator login, vaccination status (main page), data analysis, and vaccination status (used for modifying vaccination status). Following are four windows we design

1. Vaccination Status (main window):

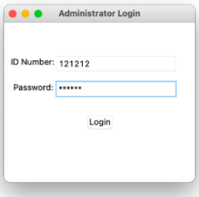


As the main page of our system, It contains 4 buttons with various functions and an entry for input. "Analysis" directs to "Data Analysis" window; "Admin" directs to "Administrator Login"; "OCR" starts the function of "OCR"; "Search" directs to " Vaccination Status (used for modifying vaccination status)".

2. Administrator Login:



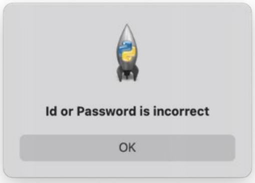
After clicking "Admin" in main window, this page will be shown without closing the main window. We have set an original password of "121212" for the ID Number of "121212".



ID Number: 121212
Password: ••••••

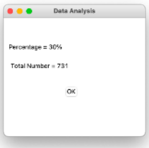
After clicking "Login" button, the third page "Vaccination status (used for modifying vaccination status)" will be disclosed, and "Administrator Login" page will disappear.

Unsuccessful user login triggers a new layout indicating error (shown on the right).



Id or Password is incorrect

3. Data Analysis

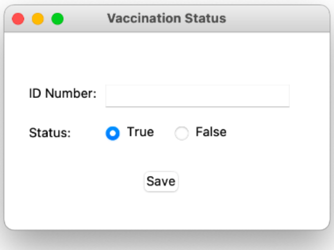


Percentage = 30%

Total Number = 731

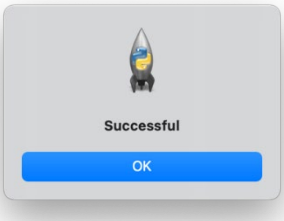
After launch "Analysis" button on the main page, "Data Analysis" is opened showing the statistics. After clicking "OK", this page will be automatically closed.

4. Vaccination Status (used for modifying vaccination status) After the successful administration login, "Vaccination Status" is activated. Notification of successful save of changes will be shown after clicking "Save" button.



ID Number:

Status: ● True ○ False



Successful

Data (DONG VanNam)

The Data Handler includes 2 parts:

1. Vaccination Information
2. Admin Information

Vaccination information:

The Vaccination Information of student/staff will be stored into data.data and its form should be:

```
Student ID;Status
21107569;1
```

The process will allow user to check the status of student/staff based on the available data. It also help admin to update and make analysis on it. For more detail, please read the DataHandler.py

Admin Information:

The Admin Information will be stored into auth.csv and its from should be:

```
Account;Random Key;Hash Code
abc;oqcambbnem;2378321
```

The process will allow admin to register, log in account. The Admin Information will include the Random Key, which is randomly generated through the log in process. The HashCode will be encrypted based on the Account and the Key. This process is based on Message Digest Algorithm, which is formally known as MD5.

For the detail of the process, please read the Data.Handler.py.