THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Opening Minds • Shaping the Future • 啟迪思維 • 成就未來

DEPARTMENT OF COMPUTING

COMP1002 Group Project

Vaccination Tracking System

Name:	*Student ID:*
CHEN Junru	21105092d
JIANG Yiyang	21095707d
Xu Le	21096101d
ZHOU Taiqi	21106717d

Dec 2021

Contents

1 Problem Description

The system was designed for the Faculty of Engineering of the Hong Kong Polytechnic University to manage and process COVID-19 vaccination information for faculty of Engineering students and staff. Through efficient system management, administrators can efficiently view the known information, search the vaccination information of designated personnel, and check the vaccination situation and statistics of students and staff.
This report describes the implementation and functionality of a vaccine tracking system developed in Python. This project is part of the Polyu COMP 1002 Computational Thinking and Problem Solving.

2 Design

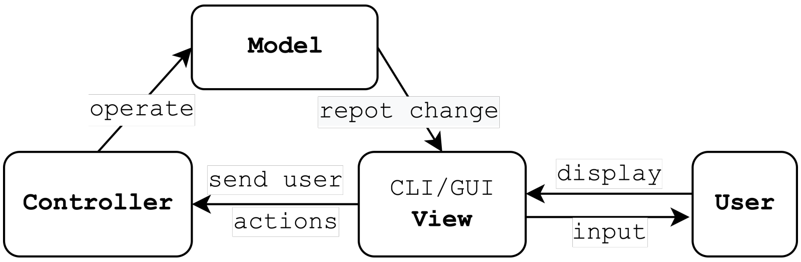

Figure 1: Overall MVC Construction Design

This project mainly adopts Model-View-Controller (MVC) approach. Figure 1 introduces the basic structure. The user interacts with the View. The user sends commands to the View, and the View also sends feedback to users. The View delivers the user input to Controller. Controller then utilises user actions to update the state of Model. Once Model has been changed, the object of View can observe the changes directly through instance variable `self.model`.

Our design adopts the most classic composite pattern, the MVC pattern, where the Model provides the data and logic (if the information is stored and queried), the View is responsible for the presentation of the data (how it is presented), and the Controller is the glue between the two, coordinating the Model and View.

2.1 Model Class, View Class, Controller Class

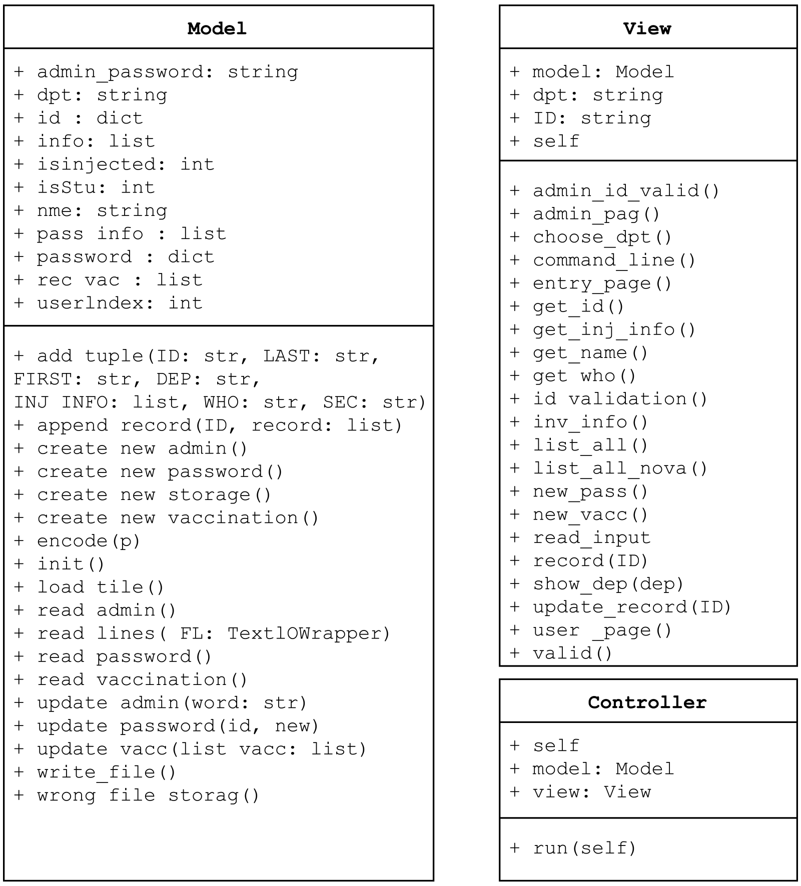

Figure 2: Model, View and Controller Class

2.2 Encrypted password

In order to prevent users from accessing the file and viewing other users' password, we irreversibly encrypt user passwords: hash a password using three different prime hash keys, and concatenate the hash values to obtain the encrypted text. The encrypted text cannot be inverted from the original text, so the user password would not be leaked.

After the password is encrypted, it is verified against the encrypted text stored in the file to determine the correctness of password. Since three hash keys are used, the probability of a collision is negligible. At the beginning of Controller Run () we call init() in Model to initialize the Instance fields of Model. Real-time updates of user information can be seen in the Model. When the whole system runs, the user may update their information, and those changes reflect in the changes of the instance variables of `model` object. At the end of the program we record the latest system state in our TXT file (i.e. write these instance fields to our TXT file).

2.3 GUI Interface

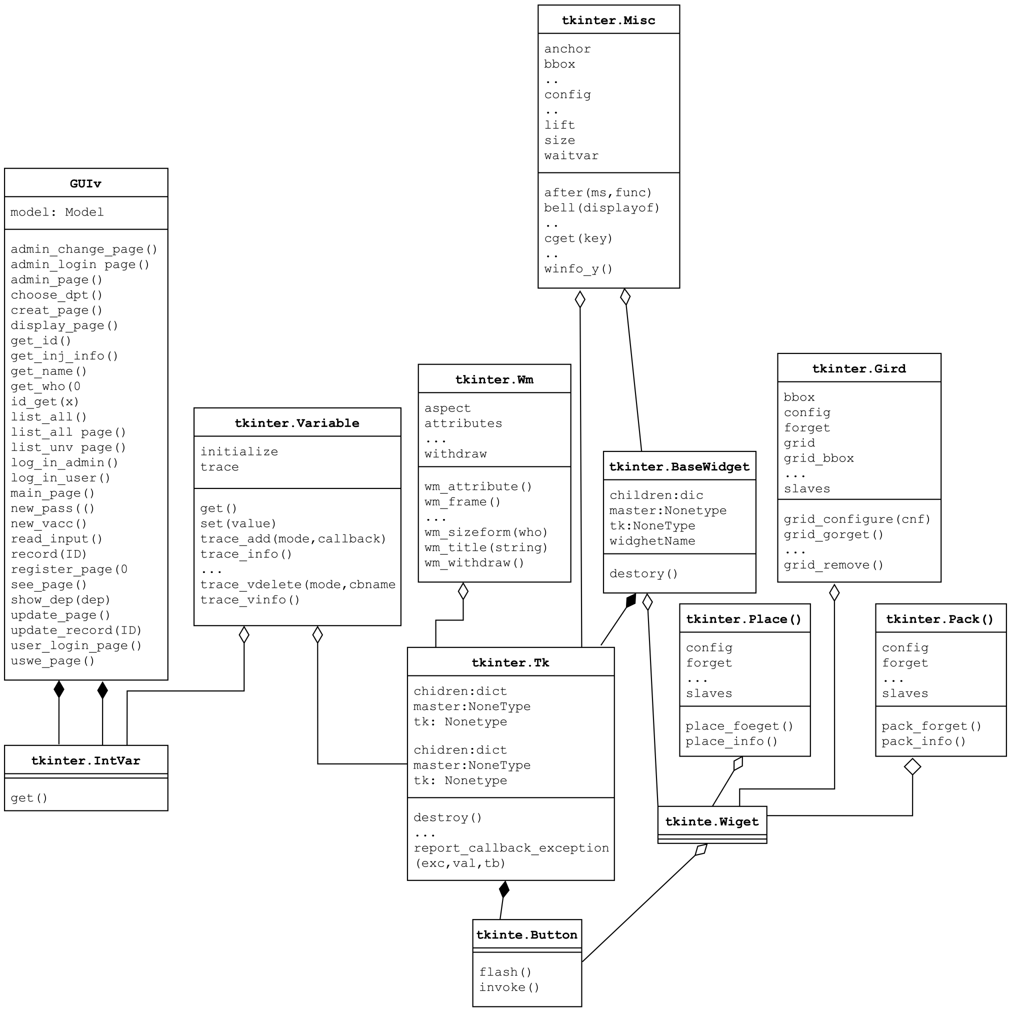

Figure 3: GUI Class UML

3 Data Abstraction

```
def __init__(self):
        self.info = []
        '''[user id, last name, first name, department, number of injection
            (-1 means fully vaccinated),[injection info],'studentornot']'''
        self.pass_info = []
        '''[user id, password]'''

        '''[the name of recognized vaccines, ... ] the admin will update the
            list of recognized vaccines according to the latest information
            from Hong Kong government. Only the recognized vaccines can be
            recorded in the file system. '''
        self.rec_vac = []
        self.admin_password = '' # password of admin

        self.id = dict()
        self.dpt = defaultdict(list)
        '''a dictionary which's key is department, admin can search the
            department to get the information of its students and stuffs'''
        self.isInjected = defaultdict(list)
        ''' a dictionary which's key is number of injection, admin can get
            the information that who have not been inejected'''
        self.userIndex = defaultdict(list)
        '''a dictionary help to get the one user's index of info[] and
            pass_info[], which can help update the information quickly'''
        self.password = dict()
        '''a dictionary help to confirm the user's password'''
```

Individual user(student or staff) information is stored as a list in the format of `[ID, first name, last name, department, vaccination number, [vaccination record, ...], student or not]`

Since the standards for complete vaccination of different vaccines may be inconsistent, we set the "-1" as the flag for complete vaccination and '0', '1', '2', '3' as the number of vaccinations. Meanwhile, the vaccination record is a list, which stores students' specific vaccination information, and we have conduct format checking for those entered vaccination record.

List is used to store all the information about a single user(student or staff) except password. This has the advantage of being able to print out the list directly into a file, or read it from a file directly through the function `eval()` of python3. String is uniformly lowercase to facilitate `dict()` queries. The last name, department and name of vaccine are capitalized as usual.

The administrator has the right to view the information of all students and staff. In order to facilitate the administrator to view specific information, we provide the administrator with two filters. The administrator could view the information of the staff or students in a specific department or the information of the students or staff who have not been vaccinated at all. In order to achieve the above functions, we use the `dict()` from python3. We use the collections `defaultdict(list)` statement in python3 to define a partial dict() whose value has an initial type of list. We use `defaultdict(list)` named **dpt** to look up user information for a particular department, such as 'COMP', where `dpt['comp']` is a list of all COMP students and staff. `isInjected['0']` returns the information of the completely non-vaccinated persons when the administrator queries their information. Password and information of student and staff are stored separately for the sake of habit and security.

4 Implemented Functions

The section introduces what function has the whole system realized.

4.1 Client and Administrator

We provide two ends to the user, different identity of the user into different interface. The user and administrator permissions are different

4.2 Log in, Register, Change Password

Users can log in with their own ID and password, and can change their password after entering the system. The initial password is 123456. New users can use the registration function to register new accounts. For administrators, the password can be changed and the initial password is admin but administrators cannot register. There is only one administrator for the entire system.

4.3 Fallback

In the selection of features, if the selection is wrong can be returned to the previous layer, do not need to reopen the program selection, improve efficiency.

4.4 Statistics and analysis

This is the function of the administrator side, the administrator can view all data, according to the partial classification. Second, administrators can see what students and teachers in the faculty of Engineering are not fully vaccinated and the percentage of them. Finally, administrators can choose to view vaccination status and vaccination rates in different departments.

4.5 Add approved vaccine

Vaccines are time-sensitive, new vaccines will be introduced and only some vaccines are approved by the Hong Kong government, and the government can change the approved types of vaccines at any time. The administrator will adjust the list according to the latest information.