

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI



---

# REPORT

---

ADVANCED PROGRAMMING WITH PYTHON



TOPIC

Human Resources Management System

Instructor: Dr. Tran Giang Son

JUNE 2, 2021

GROUP ANOTHER

<https://github.com/Lib3Rt9/pp2021/tree/master/ProjectMid2>

## Contents

<b>GLOSSARY</b>	2
<b>I. OVERVIEW</b>	3
1. About	3
2. Class	3
3. Topic	3
4. Roles	3
5. Why Need dreamcatcher	3
6. What dreamcatcher Do	3
<b>II. THEORETICAL BASIS</b>	3
1. Research Topic	3
2. Summary	3
3. Analysis and Evaluation	4
4. Conclusion	4
<b>III. PROJECT CONTENT</b>	5
1. Database Diagram	5
2. Python Modules	5
3. Python Classes	8
4. UI Structure	9
<b>IV. RESULT</b>	11

## GLOSSARY

Jargon	Explanation
(path)	Path way to the folder which contains all files and application
s_list	Staff list
f = open("C:/Users/G40/Desktop/Human Resource/Database/Staff_s.pickle", "wb") pickle.dump(s_list, f) f.close()	Source code
f = open("C:/Users/G40/Desktop/Human Resource/Database/Staff_s.pickle", "wb") pickle.dump(s_list, f) f.close() os.remove("C:/Users/G40/Desktop/Human Resource/Database/Staff{ }.pickle".format(del_id.get()[0]))	Source code
total = wage * working_hour total = math.floor(total)	Source code
birthday = DoB.get() birthday = birthday.split("/", 2) x = datetime.datetime(int(birthday[2]), int(birthday[1]), int(birthday[0])) birthday = (datetime.date.strftime(x, "%d/%m/%Y"))	Source code
def login(): office = check_user_account(username, password) if office == "Alpha": manager.main() else: staff.main(office[0])	Source code

## I. OVERVIEW

### 1. About

- Group name: Another
- Group member:
  - Pham Gia Phuc – BI10-138
  - Tran Dac Minh – BI10-113
  - Nguyen Tan Dung – BI10-041
  - Dang Gia Linh – BI10-101
  - Nguyen Viet Duc – BI10-032

### 2. Class

- ICT

### 3. Topic

- Human Resource Management System
- Language used: Python

### 4. Roles

- Leader: Pham Gia Phuc (Coding and fixing)
- UI Designer and coding: Tran Dac Minh
- OOP Updater, Support Database & Slide: Nguyen Tan Dung, Dang Gia Linh
- Database Diagram & Database Schema: Nguyen Viet Duc

### 5. Why Need dreamcatcher

- Number of employees: growing up
- Need better way to storage and manage resources
- Much more efficient than paper bases.

### 6. What dreamcatcher Do

- dreamcatcher provide:
  - Basic solution for consulting, staffing, recruiting
  - Improve aspects of business

## II. THEORETICAL BASIS

### 1. Research Topic

- Human Resource Management System
- Python programming language

### 2. Summary

- Python is a high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991
- Python is the global most popular programming language. (Mar 2021)

### 3. Analysis and Evaluation

Python is designed with the strong advantage of being easy to read, learn, and remember. Python is a language with a very bright form, clear structure, convenient for beginners to learn programming.

Moreover, there are some other characteristics that made python flexible, which can be mentioned as:

- Both procedural-oriented and object-oriented
- Have standard libraries and external modules which covering all programming needs
- Have ability to interact with modules written in other programming language
- Etc...

Because of the advantages compare to some other popular programming language, such as Java, Java Script, C# and C/C++, etc..., Python has a wide range of usage. Some of them may include as the following:

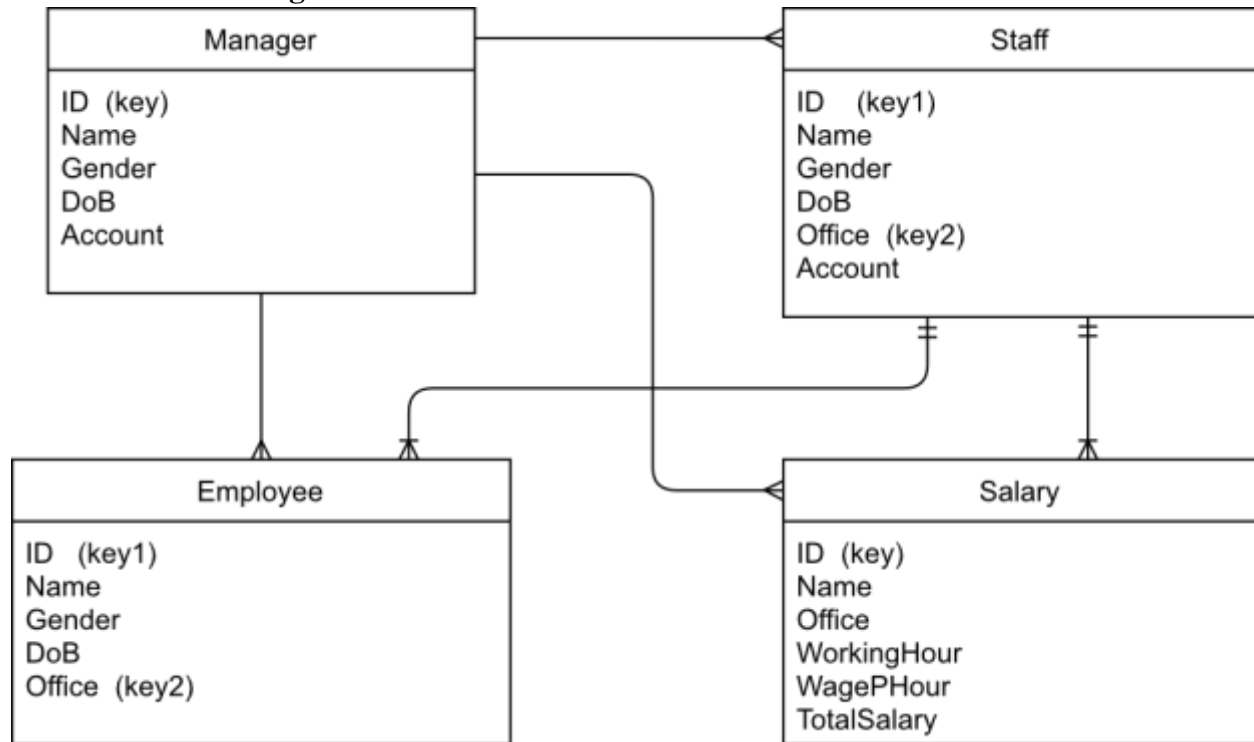
- Script
- Backend
- Machine Learning, Deep Learning
- Applications:
  - o Mobile
  - o Desktop
  - o Web
- ...

### 4. Conclusion

- Python has easy syntax and clear structure
- It is quite easy for beginners to learn about programming
- Python can be applied to a lot of term in the field of programming

### III. PROJECT CONTENT

#### 1. Database Diagram



#### 2. Python Modules

##### a. Tkinter:

For creating GUI. Tkinter is used because it is easy to learn and easy to work with

##### b. Pickle:

For creating and accessing database files, Pickle is used due to its suitability for object data type, in dreamcatcher's case, list of objects

Example:

```
f = open("(path)/Human Resource/Database/Staff_s.pickle", "wb")
pickle.dump(s_list, f)
f.close()
```

Create database file for **s\_list** (staff list) using **pickle.dump()**

```
f = open("(path)/Human Resource/Database/Staff_s.pickle", "rb")s_list = pickle.load(f)
f.close()
```

Read database file for **s\_list** (staff list) using **pickle.load()**

#### c. Os:

For deleting database file when necessary (when deleting info)

Example:

```
f = open("(path)/Human Resource/Database/Staff_s.pickle", "wb")
pickle.dump(s_list, f)
f.close()
os.remove("(path)/Human Resource/Database/Staff{ }.pickle".format(del_id.get()[0]))
```

This is in **del\_staff()** function in manager.py : after deleting staff info from the list, program will update *database* for **s\_list** (staff list) and delete old one using **os.remove()**, note that **del\_id.get()[0]** is the first letter of the staff id which is being deleted, which is also the office of that staff.

#### d. Math:

For calculating and rounding up the 'Salary'

Example:

```
total = wage * working_hour
total = math.floor(total)
```

Using **math.floor()** to round up total salary

#### e. Datetime:

For checking if the 'DoB' is input in DD/MM/YYYY form

Example, in sources code, **datetime** is used as

```
birthday = DoB.get()
birthday = birthday.split("/", 2)
x = datetime.datetime(int(birthday[2]), int(birthday[1]), int(birthday[0]))
birthday = (datetime.date.strftime(x, "%d/%m/%Y"))
```

Display 'DoB' in DD/MM/YYYY form: three steps

Split the input 'DoB', got birthday which is a list consists of 3 numbers

Create new datetime object x and assign its value correspond to each element of 'birthday' in the reversed order, use **datetime.datetime()**

Display x using **datetime.date.strftime()**, this will display attributes of x in reversed order (the default order is YYYY/MM/DD)

If there is exception then there will be a message box popping up

#### f. Manager.py and Staff.py:

Two modules for separating Manager and Staff account, which lead to the **main()** function for Manager or Staff.

Based on Username and Password

Example:

```
def login():
    office = check_user_account(username, password)
    if office == "Alpha":
        manager.main()
    else:
        staff.main(office[0])
```

Input "username" and "password", in case of correct "password" :

if "username" is "Alpha", then run **manager.main()**

if "username" different to "Alpha", then run **staff.main()**

if "username" is not in *database*, then get error message box

"office" is assigned with value of the input "username"

The **office[0]** is the first letter of the input "username" which is also the ID given to the Staff, it will decide which office for the Staff

#### g. Check\_Func.py:

The module for keeping gender, id and office of Manager, Staff and Employee from inaccuracy and duplication.



Contains 3 functions:

def check_gender(gender):	
def prevent_duplicate(id, list):	
def prevent_duplicate2(office, list):	# For Staff only

### 3. Python Classes

For more organization when inputting new info or displaying info from database files, this program has 4 classes:

- Manager (ID, Name, DOB, Gender, Password):
  - o The Manager ID is permanently set as 'Alpha'
- Staff (ID, Name, DOB, Gender, Password, Office)
  - o Staff ID cannot be 'Alpha', Staff Office is based on the first letter of ID
  - o Staff ID is unique
- Employee (ID, Name, DOB, Gender, Office)
  - o Employee ID is unique
- Salary (ID, Name, Work hour, Wage/hour, Office, Total Receive)

The Manager ID is 'Alpha'. The Staff ID will be in charge by the Manager and the Employee ID will be in charge by the Staff.

In this program, the number of staff is limited to maximum 7, while number of employees is 10 in maximum per office.

Technically, there can be as many staff or employee as possible, however, it is limited for better demonstration.

'DoB' must be in form DD/MM/YYYY.

'Gender', only take 'Male' or 'Female' value.

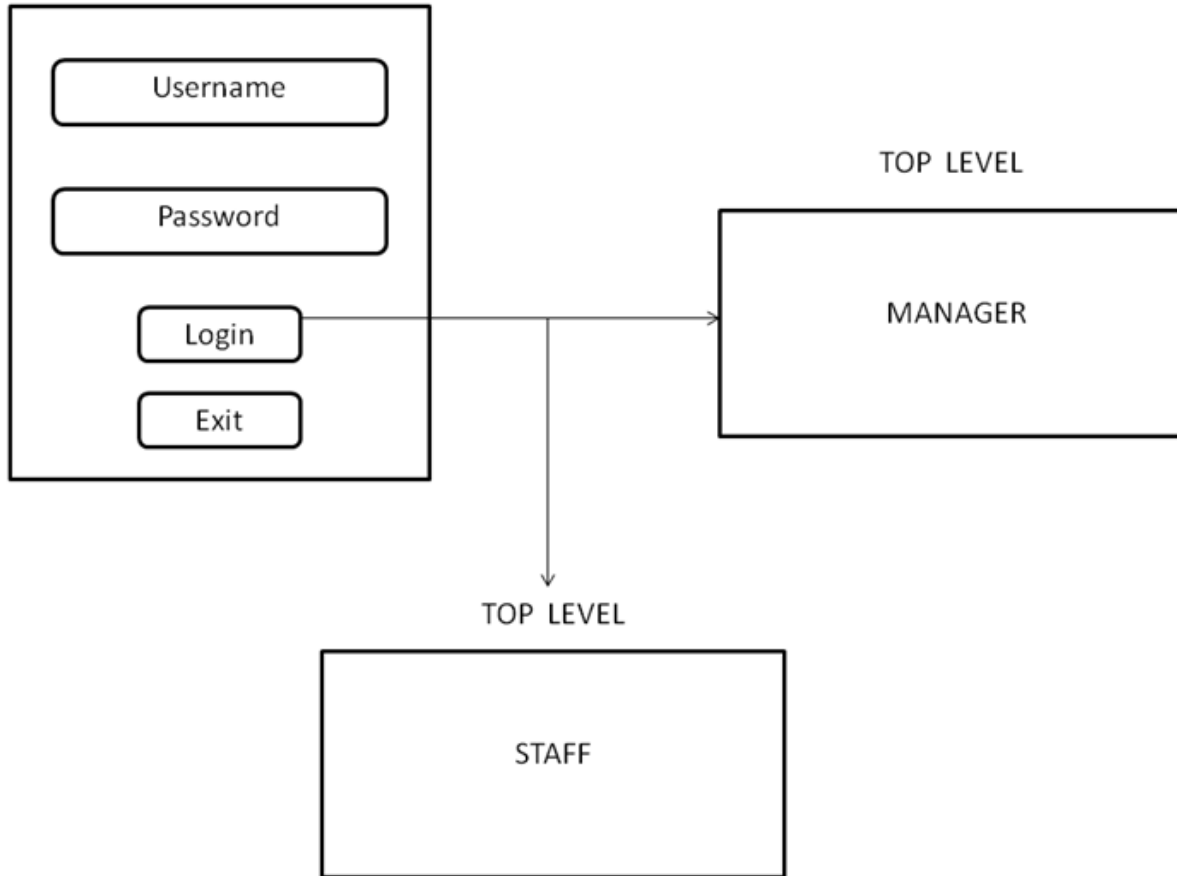
The 'Office' is based on the first letter of Staff ID, so only Manager can decide the office.

The Salary is only for Staff and Employee. The Manager can input Salary for Staff and Staff can input Salary for Employee.

Note that the Manager, even though he cannot modify Employee Info or Salary, he can see all the Employee Information and Salary of all Office. And that is the only difference between the Manager and Staff account.

#### 4. UI Structure

##### LOGIN WINDOW:

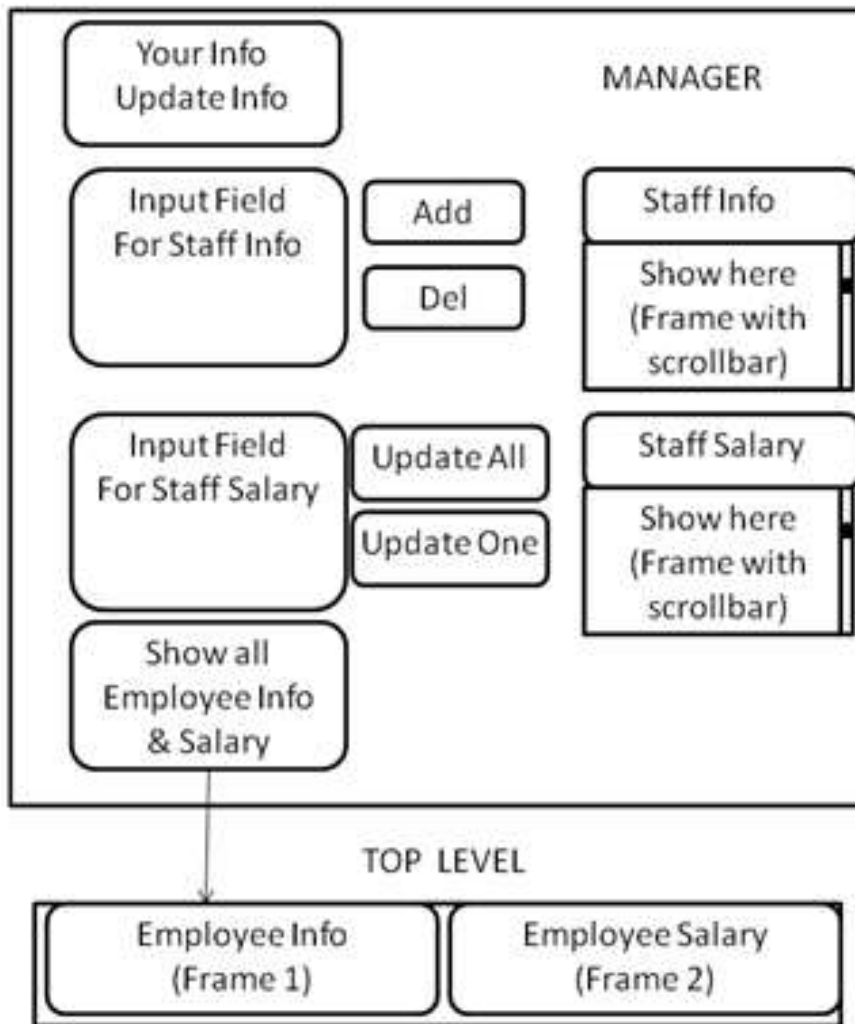


When running the program, the first thing appear is the login window. Inputting the correct.

Username and Password then pressing Login will open a new window on top of the login window (TopLevel in Tkinter).

It can be either Manager window or Staff window depending on user's input.

Warning that if login window is closed, the whole program will be shut down.

**MANAGER WINDOW:**

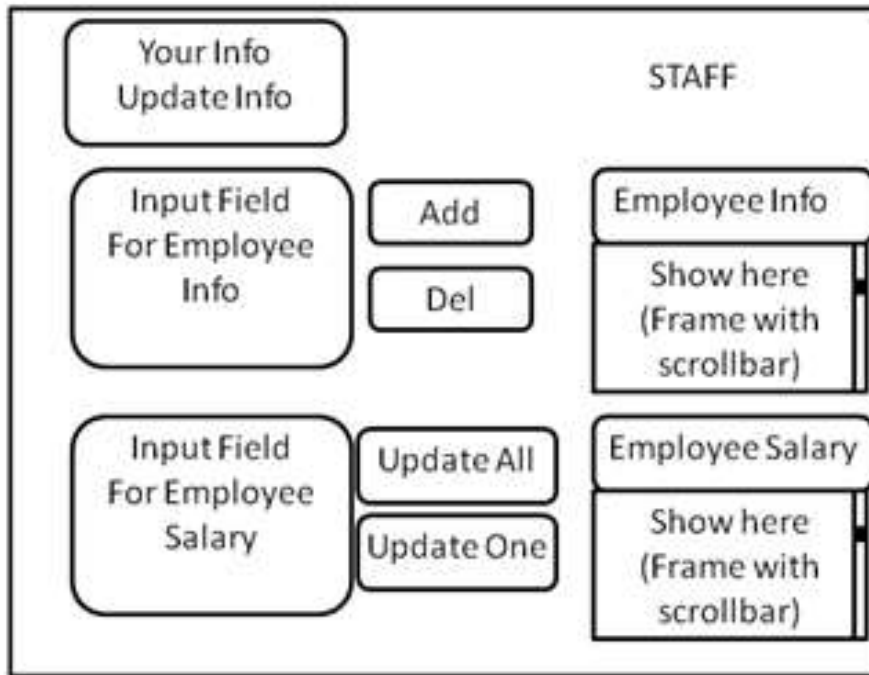
When login to the Manager account, a window with some buttons and input fields as the image in the left is popped up.

It will also show the existing Staff Info and Staff Salary from the database; if there is no database file existing, then there will be a message box.

When input something in the input field and click, for example, 'Add' button, it will check if the input is valid or not and then add the staff info to the list. After that, the program will

automatically refresh the frame and show the new list on the screen.

When click the 'Show all...' button there will be a new window (TopLevel on TopLevel) pop up which contains two frames for showing the list 'Employee Info' and 'Employee Salary' of all offices.

**STAFF WINDOW:**

The Staff window is pretty much the same as the Manager window except for the 'Show all...' button as the image in the left.

Note:

The scrollbar feature hasn't been updated in the demo version because it has not been necessary yet. As mentioned before, all the lists in this program are smaller or less than 10 elements, while

the scrollbar is only necessary when there are more than 12 elements. As a result, it is being removed.

## IV. RESULT

In General, dreamcatcher has the basic functions of a management system.

About dreamcatcher:

The program provides a better solution of storing and managing human resources.

It is suitable for companies, schools or individual groups.

Some advantages when using dreamcatcher:

- Easy for controlling and managing details of the human????
- Manager can base on it to promote or demote staffs and employees
- Friendly user interface
- Take short time to get used

Some disadvantages when using dreamcatcher:

- Cannot reset password in case of forgot.
- There is no edit history

- Cannot search one by one or sort one column

In conclusion, even though dreamcatcher still lack some advanced features, it performs pretty well in different occasions.

Also, it is light, so that machine resources will be saved.

Furthermore, dreamcatcher is free to use and suitable for wide range of user.