

# BÁO CÁO ĐỒ ÁN MÔN HỌC

Lớp: IT002.O21.TTNT

SINH VIÊN THỰC HIỆN

Mã sinh viên: 23521570

Họ và tên: HUỖNH VIỆT TIẾN

TÊN ĐỀ TÀI: TENGEE – APP QUẢN LÝ BÁN KHÓA HỌC

CÁC NỘI DUNG CẦN BÁO CÁO:

## I. Giới thiệu đồ án

### 1. Mô tả chi tiết về nội dung và các yêu cầu

**1.1 Languages:** python (logic), html (ui)

**1.2 Mục đích:** demo cho đồ án môn học OOP.

Đồ án này tạo một ứng dụng tương tác giữa người cung cấp khóa học (teacher) với người mua khóa học (student).

### 1.3 Yêu cầu project:

- docstring cho từng class, method
- có tính hướng đối tượng
- có sơ đồ lớp UML

### 1.4 Yêu cầu các chức năng:

#### a. hệ thống login users

Chia người dùng thành 2 class (student or teacher)

#### b. hệ thống profile

Teacher cung cấp *thông tin cá nhân*, *số tài khoản* vào hệ thống profile

Student cung cấp *thông tin cá nhân* vào hệ thống profile

#### c. Trang home

Hiện thị được các tính năng: hiện khóa học từ dữ liệu, hiện các nút cơ bản

**d. Trang courses**

Cung cấp dữ liệu các khóa học đã mua dành cho student

**e. Trang Courses Management**

Cung cấp các tính năng quản lý bao gồm đăng, phân tích các khóa học mới dành cho teacher.

**2. Link github**

Link: <https://github.com/SharkTien/Tengee>

**II. Quá trình thực hiện**

**a. Tuần 1:**

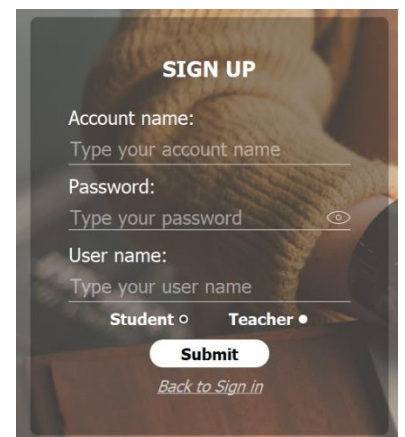
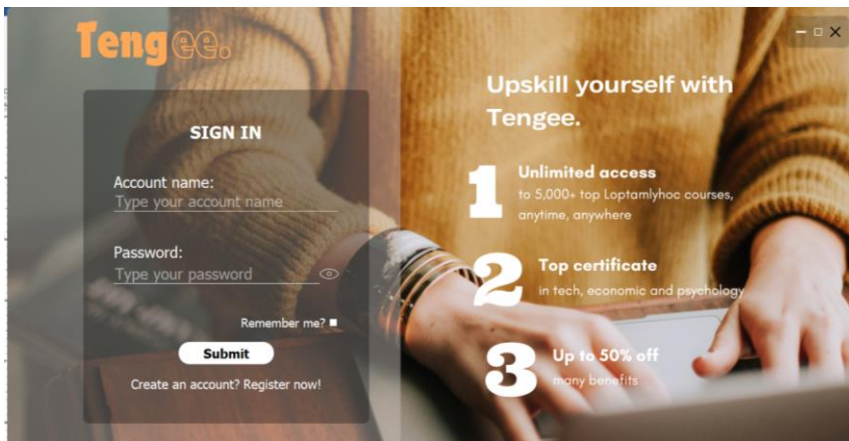
Các thư viện và class module có sẵn xuất hiện ở tuần 1: time, pyqt5 (class QtCore, Qt, QtWidgets, UIC, QMainWindow, và các class giao diện khác)

- Sử dụng pyqt5 để xây dựng thiết kế giao diện cửa sổ
- Thiết kế màn hình loading

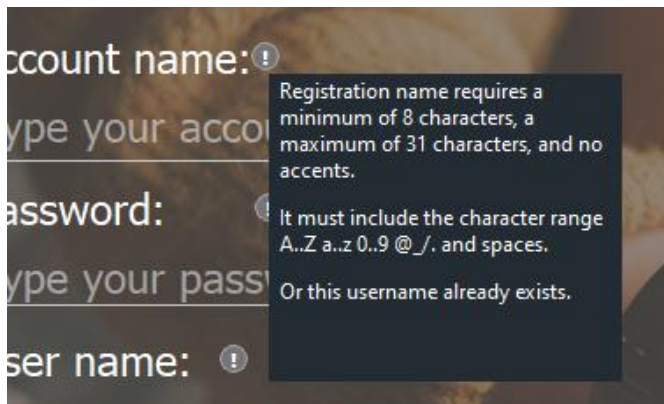


- Tạo hệ thống logic đăng nhập, đăng ký tài khoản

- Thiết kế màn hình login và tab register



- Tạo các điều kiện nhập khi đăng ký và kiểm tra độ chính xác khi đăng nhập



Người dùng hover vào icon ! để hiện bản tooltip này

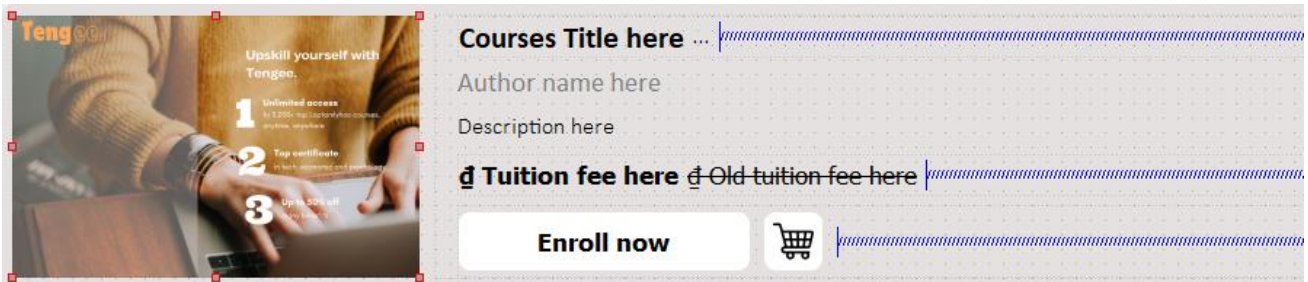
- Tạo truy xuất file dữ liệu người dùng cơ bản (account name, password, user name, role)

Dữ liệu sau khi đọc file là list có format:

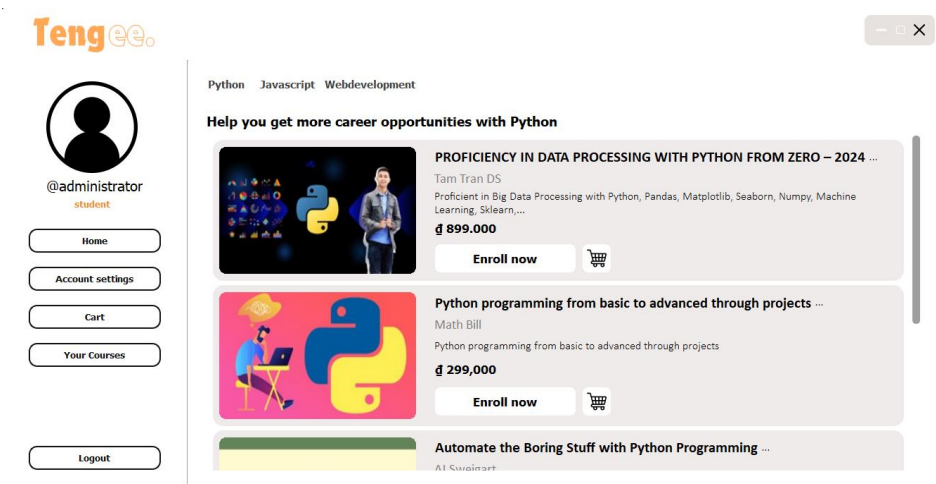
```
[{'accountname': 'password', 'username', '1' nếu là teacher, '0' nếu là student'}]
```

- Tính năng remember me

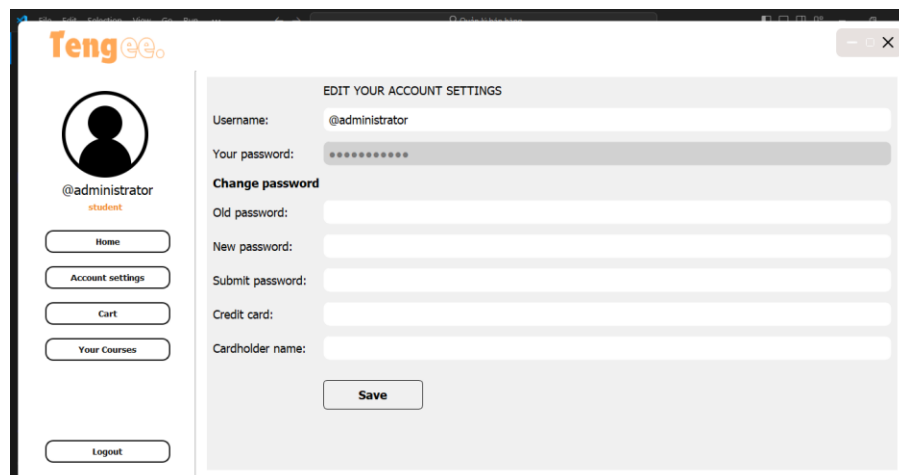
- Thiết kế màn hình home
- Thiết kế card khóa học



- Truy xuất dữ liệu và lặp vòng for cho từng card khóa học trong một scrollable frame

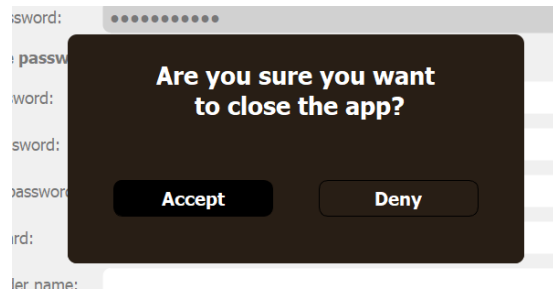


- Thêm tính năng tìm kiếm thông tin khóa học (bao gồm title, author, description...)
- Thêm các button categories để lọc khóa học theo lĩnh vực
- Thiết kế màn hình chức năng sửa tên, sửa mật khẩu, đăng xuất, lưu mật khẩu



- Thêm nút Logout đăng xuất để thuận tiện cho chuyển đổi giữa các tài khoản
- Thiết kế màn hình xác nhận thoát ứng dụng

Nhằm tránh trường hợp ấn nhầm nút tắt làm gián đoạn tác vụ đang thực hiện



Khi nút này xuất hiện, mọi tác vụ sẽ bị disabled và không tương tác được cho đến khi người dùng chọn Thoát ứng dụng (Accept) hoặc tiếp tục dùng ứng dụng (Deny).

b. Tuần 2:

c. Tuần 3:

d. Tuần 4:

### III. Kết quả đạt được

a. Các bản thiết kế

Loading Screen

Login window

Home window

Card Layout

Profile window

UML database

UML class

b. Sản phẩm cuối cùng

#### IV. Tài liệu tham khảo

Đề tài NCKH: <https://github.com/hoangm960/Astraea>

#### V. Phụ lục: docstring

#### Phụ lục docstring

Controller.....	8
__init__(self, version).....	8
show_loading(self) .....	8
show_login(self) .....	9
show_home(self, role, data) .....	9
show_quit(self) .....	9
disable_windows(self, state, all_main=False).....	9
close_login(self) .....	9
close_home(self).....	9
close_pg(self).....	9
LoadingScreen .....	9
__init__(self, version).....	9
initUI(self) .....	9
UIFunction>Loading Screen).....	10
__init__(self, ui) .....	10
update_version(self, ui) .....	10
delay(self, point, wait) .....	10
progress(self, ui) .....	10
LoginScreen(QMainWindow).....	10
__init__(self) .....	10

initUI(self) .....	10
moveWindow(self, event) .....	11
mousePressEvent(self, event) .....	11
LoginFunctions(LoginScreen) .....	11
__init__(self, ui) .....	11
connect_btn(self, ui) .....	11
default(self, ui) .....	11
openQuitFrame(self, ui) .....	12
check_autosave(self, ui) .....	12
maximize_restore(self, ui) .....	12
get_data(self) .....	12
check_SI(self, ui) .....	12
open_home(self, ui, role, data) .....	12
check_SU(self, ui) .....	12
CardFrame(QWidget) .....	12
__init__(ui, *args, **kwargs) .....	12
HomeScreen(QMainWindow) .....	12
__init__(self, role, data) .....	13
init_UI(self) .....	13
define_role(self) .....	13
UIFunctions(HomeScreen) .....	13
__init__(self, ui) .....	13
connect_btn(self, ui) .....	14
quit(self, ui) .....	14

maximize_restore(self, ui) .....	14
change(self, ui) .....	14
backlogin(self, ui) .....	14
load_data(self, ui, keyword) .....	14
StudentUIFunctions(UIFunctions) .....	14
__init__(self, ui) .....	14
TeacherUIFunctions(UIFunctions).....	14
__init__(self, ui) .....	15

## Controller

Controller: Class controll opened windows and related functions.

method:

    \_\_init\_\_(version)

    show\_loading

    show\_home(self, role, data)

    show\_quit

    disabled\_windows

**\_\_init\_\_(self, version)**

\_\_init\_\_ initiate the attributes for Controller

**show\_loading(self)**

show\_loading: open loading window



### **show\_login(self)**

show\_login: open login window

### **show\_home(self, role, data)**

show\_home(self, role, data): show home screen after sign in succession

### **show\_quit(self)**

show\_quit: open quit window

### **disable\_windows(self, state, all\_main=False)**

disable\_windows(state): Disable windows, Users cannot interact to windows when this function is running

### **close\_login(self)**

close\_login(): close login window after sign in succession

### **close\_home(self)**

close\_home(self): close home window after logout

### **close\_pg(self)**

close\_pg: close window

## **LoadingScreen**

class LoadingScreen:

method:

\_\_init\_\_(version)

initUI(version)

### **\_\_init\_\_(self, version)**

\_\_init\_\_(version): initiate attributes for LoadingScreen

### **initUI(self)**

initUI(): initiate and move window to center of screen, then initiate attributes

## UIFunction>Loading Screen)

class UIFunction for LoadingScreen including methods processing progress bar  
include version

**\_\_init\_\_(self, ui)**

\_\_init\_\_ initiate methods and attributes for UIFunction

**update\_version(self, ui)**

update\_version(ui): update the version of the entire Uis

**delay(self, point, wait)**

delay(point, wait): This affects to progress function, if count == /point then  
time.sleep(/wait)

**progress(self, ui)**

progress(ui): the preprocessing data, ui,... before running the Home Window and  
functions... This include increasing value of progressBar in range 0 to 100

## LoginScreen(QMainWindow)

class LoginScreen:

method:

\_\_init\_\_()

initUI()

moveWindow(event)

mousePressEvent(event)

**\_\_init\_\_(self)**

\_\_init\_\_(): initiate attributes for Login Screen.

**initUI(self)**

initUI(): inititate and move window to center of screen, then initiate attributes

**moveWindow(self, event)**

moveWindow(event): enable to drag the window

**mousePressEvent(self, event)**

mousePressEvent: catch event dragging

## **LoginFunctions(LoginScreen)**

class LoginFunctions:

method:

\_\_init\_\_(ui)

connect\_btn(ui)

default(ui)

openQuitFrame(ui)

check\_autosave(ui)

maximize\_restore(ui)

check\_SI(ui)

check\_SU(ui)

open\_main(ui)

get\_data(ui)

**\_\_init\_\_(self, ui)**

\_\_init\_\_(ui): initiate functions for ui

**connect\_btn(self, ui)**

connect\_btn(ui): add function for button close, minimize, maximize

**default(self, ui)**

default(ui): Function reset state

### **openQuitFrame(self, ui)**

openQuitFrame(ui): open Quit window

### **check\_autosave(self, ui)**

check\_autosave(ui): open data file to check saved account

### **maximize\_restore(self, ui)**

maximize\_restore(ui): check if the window is maximize or not to maximize and minimize

### **get\_data(self)**

get\_data() -> list: get data from file data. Format:

[{'accountname': ['password', 'username', bool]]

### **check\_SI(self, ui)**

check\_SI(ui): Function check if the information login is valid

### **open\_home(self, ui, role, data)**

open\_home(ui, role, data): open home window

### **check\_SU(self, ui)**

check\_SU(ui): check if the register information is valid

## **CardFrame(QWidget)**

class CardFrame: Generate UI Card for fetching data

method:

\_\_init\_\_(ui, \*args, \*\*kwargs)

**\_\_init\_\_(ui, \*args, \*\*kwargs)**

initiate class with attributes from class QWidget and load UI file

## **HomeScreen(QMainWindow)**

Class HomeScreen:

method:

`__init__(self, role)`

`init_UI(self)`

`define_role(self)`

**`__init__(self, role, data)`**

`__init__(self, role, data)`: initiate attributes for home screen with users' role and page status to define users' role

**`init_UI(self)`**

`init_UI(self)`: initiate attributes for home screen

**`define_role(self)`**

`define_role(self)`: divides class functions based on users' role

## **UIFunctions(HomeScreen)**

class UIFunctions:

setup all functions needed for both students and teachers role

methods:

`__init__(self, ui)`

`connect_btn(self, ui)`

`quit(self, ui)`

`maximize_restore(self, ui)`

`change(self, ui)`

`backlogin(self, ui)`

**`__init__(self, ui)`**

`__init__(self)`: initiate button functions and data connection.

### **connect\_btn(self, ui)**

connect\_btn(self, ui): connect functions of button

### **quit(self, ui)**

quit(self, ui): show quit screen

### **maximize\_restore(self, ui)**

maximize\_restore(ui): check if the window is maximize or not to maximize and minimize

### **change(self, ui)**

change(self, ui): edit data including password, nameaccount, creditcards and save into data file

### **backlogin(self, ui)**

backlogin(self, ui): sign out account

### **load\_data(self, ui, keyword)**

load\_data(self, ui, keyword): Fetch data based on the keyword from data file

## **StudentUIFunctions(UIFunctions)**

class StudentUIFunctions

This class generates functions for users who are students.

methods:

\_\_init\_\_(self, ui)

### **\_\_init\_\_(self, ui)**

\_\_init\_\_(self, ui): initiate attributes for Student role and related functions

## **TeacherUIFunctions(UIFunctions)**

class TeacherUIFunctions

This class generates functions for users who are teachers.

methods:

`__init__(self, ui)`

`__init__(self, ui)`

`__init__(self,ui):` initiate attributes for Teacher role and related functions