

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: TENGE – 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 ✓
- Code python theo 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 lí 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. Yêu cầu người dùng click vào nút ? Để hiện bảng tooltip
- 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:

Cải thiện sự rõ ràng của tính hướng đối tượng, bỏ qua các class của UI, thư viện và thiết kế lại sơ đồ UML cho mục đích chính của vấn đề.

Xây dựng lại vấn đề: Tạo một **cơ sở quản lí** dữ liệu. **Dữ liệu** được chia làm 2 loại là **người dùng** và **khoá học** và đều được đánh **mã số ID**. Mỗi **người dùng** sẽ có **tên tài khoản, mật khẩu, tên người dùng, vai trò**. Mỗi **khoá học** sẽ có **tên khoá học, tác giả, mô tả, giá, thumbnail**. Biết **người dùng** chia tiếp thành 2 vai trò là **giáo viên** hoặc là **học sinh**. Giáo viên sẽ bao gồm thông tin về tài khoản ngân hàng (tên ngân hàng/ví điện tử, tên chủ thẻ, số tài khoản, mã QR, mã QR) và danh sách các khoá học đã tạo. Học sinh sẽ bao gồm thông tin các khoá học đã mua.

Bổ sung file data_init.py chứa tất cả các class mô tả vấn đề như trên sử dụng: tính đóng gói (public, private), tính kế thừa và quan hệ 1...*, *...1, 1...1, *...* và docstring
Modify file ui lại để chạy với file data_init.py

c. Tuần 3:

d. Tuần 4:

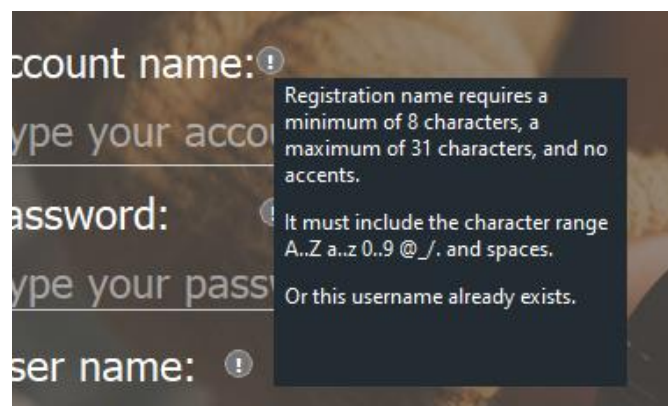
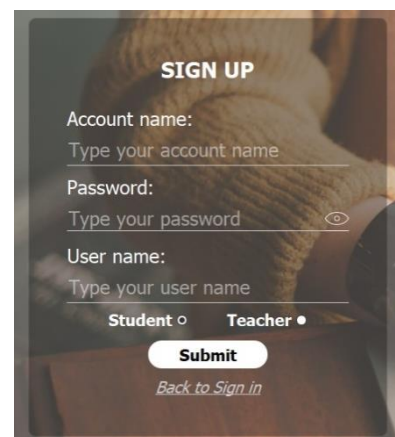
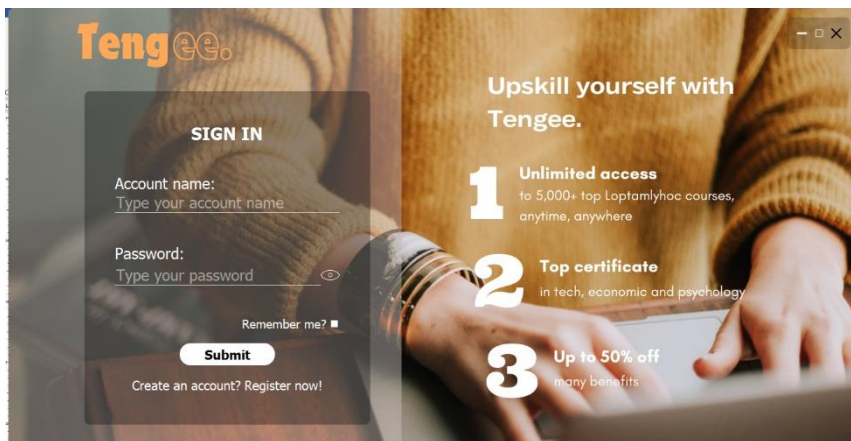
III. Kết quả đạt được

a. Các bản thiết kế

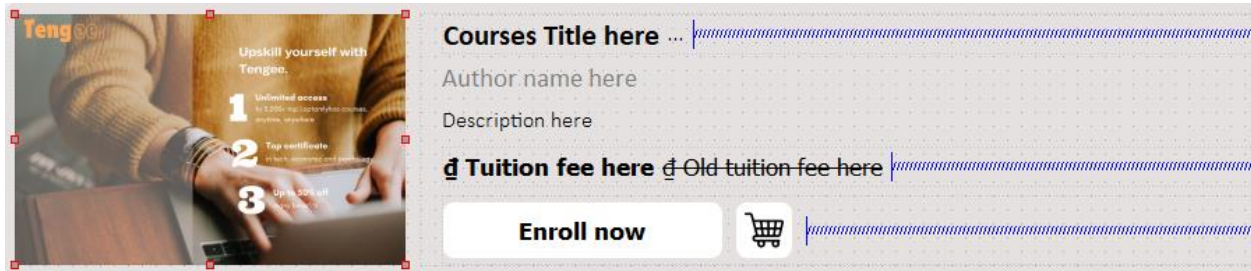
Loading Screen



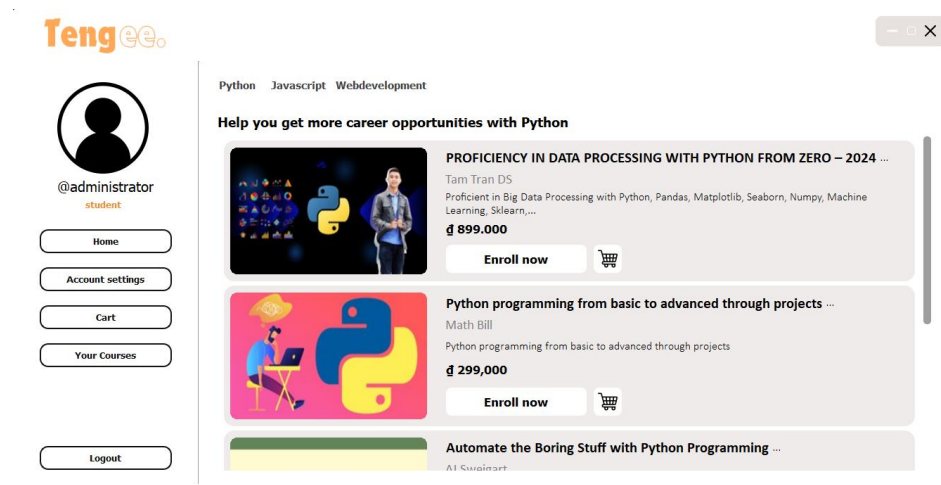
Login window



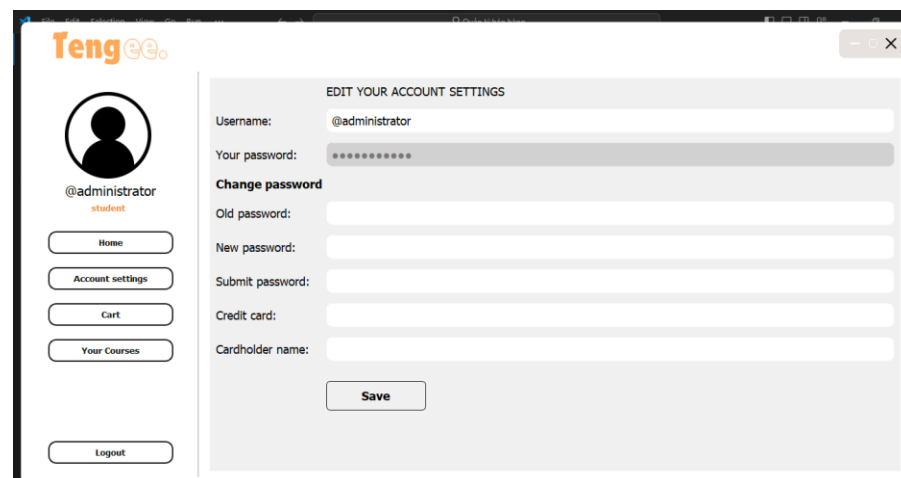
Card Layout



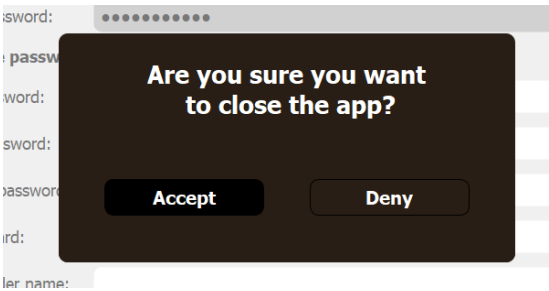
Home window



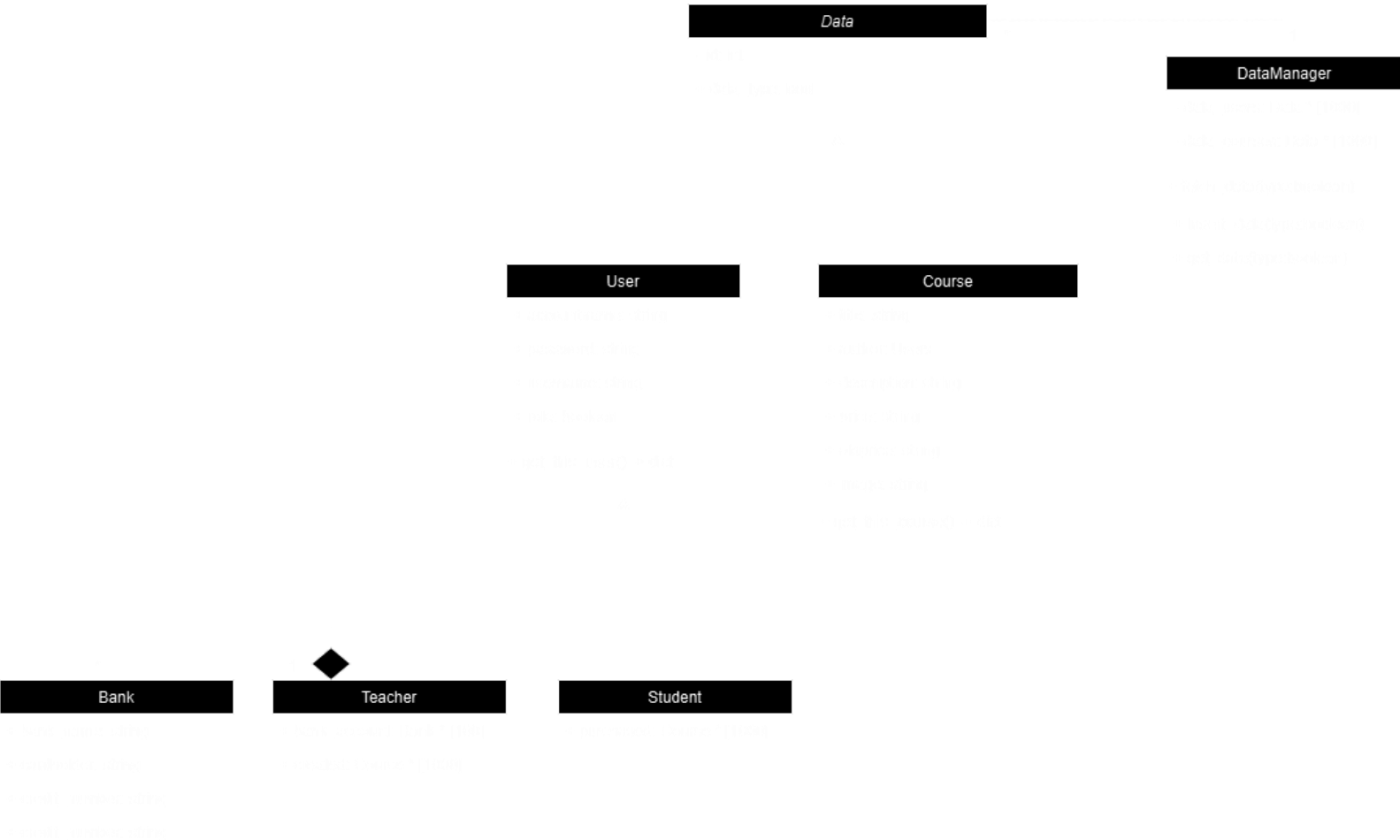
Profile window



Exit Query window



UML class diagram



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

DataManager:	7
__init__():	8
fetch_data(type: bool):	8
get_data(data_type):	8
Data:	8
__init__(id, data_type):	8
Course(Data):	9
__init__(id, data_type, title, author, description, price, oldprice, image)	9
get_this_course()	9
User(Data)	9
__init__(id, data_type, accountname, password, username, role)	9
get_this_user()	10

DataManager:

Class DataManager initiates list users = [], list courses = [] with basic method

method:

+ __init__()

+ fetch_data(type: bool)

+ insert_data(type:bool)

+ get_data(type:bool)

__init__():

initiates original variables and const value

insert_data(type: bool, data: list): insert a data (user/ course based on [type]) into user list/ course list

data is a list with format: [accountname, password, username, role]

fetch_data(type: bool):

get data from data file and add into user list/ course list

user: [id, 1, accountname, password, username, role]

course: [id, 0, title, author, description, price, oldprice, thumbnails url]

get_data(data_type):

return list of users or courses based on [data_type]

Data:

class Data: generate a data storing id and data_type, which is the familiar attributes of Course and User data.

method:

__init__(id, data_type)

__init__(id, data_type):

initiates original variables

Course(Data):

class Course(Data): generate a data storing title, author, description, price, oldprice and image then inherits id and data type

method:

`__init__(id, data_type, title, author, description, price, oldprice, image)`

`get_this_course()`

`__init__(id, data_type, title, author, description, price, oldprice, image)`

`__init__(id, data_type, title, author, description, price, oldprice, image)`: initiates original variables and inherits variables from parent classes

`get_this_course()`

`get_this_course()`: return a dictionary with keys including: title, author, description, price, oldprice and image

User(Data)

class User(Data): generate a data storing accountname, password, username and role then inherits id and data type

method:

`__init__(id, data_type, accountname, password, username, role)`

`get_this_user()`

`__init__(id, data_type, accountname, password, username, role)`

`__init__(id, data_type, accountname, password, username, role)`: initiates original variables and inherits variables from parent classes

get_this_user()

get_this_user(): return two value accountname, [password, username, role]