

# 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 ✓
- 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

(Phần chi tiết của khóa học)

**e. Trang Courses Management**

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

**f. Trang Thanh toán**

Cung cấp màn hình thông tin ngân hàng

Thiết kế chức năng thông báo đơn hàng đến giáo viên

**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:

- Trong quá trình thử nghiệm, tiếp tục thay đổi sơ đồ lớp.
- Thiết kế màn hình quản lý của giáo viên
  - Màn hình xem các khoá học đã tạo
  - Thiết kế màn hình edit khoá học
  - Thiết kế màn hình và chức năng tạo khoá học
  - Nghiên cứu cách đăng ảnh lên app để sử dụng làm banner khoá học: thư viện shutil, hàm shutil.copy(src, dst) thực hiện sao chép thư mục ảnh từ folder local của user vào thư mục src của ứng dụng.

- Màn hình enrollment hiện chi tiết khoá học

d. Tuần 4:

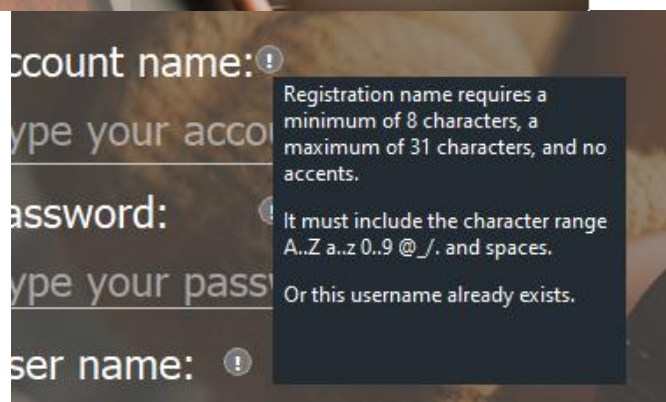
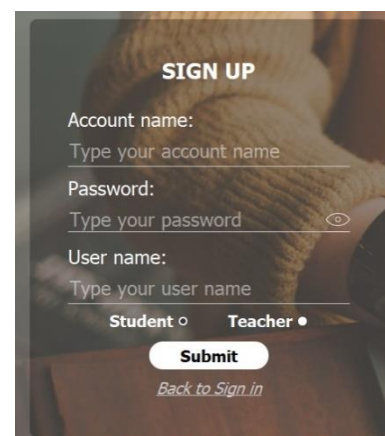
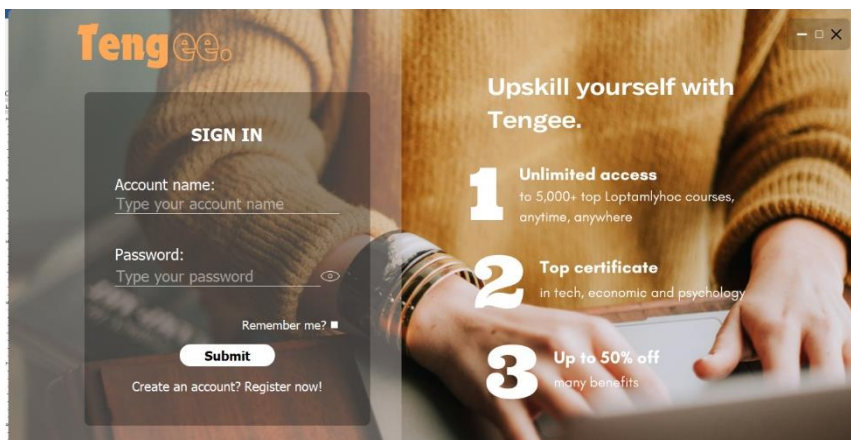
### III. Kết quả đạt được

a. Các bản thiết kế

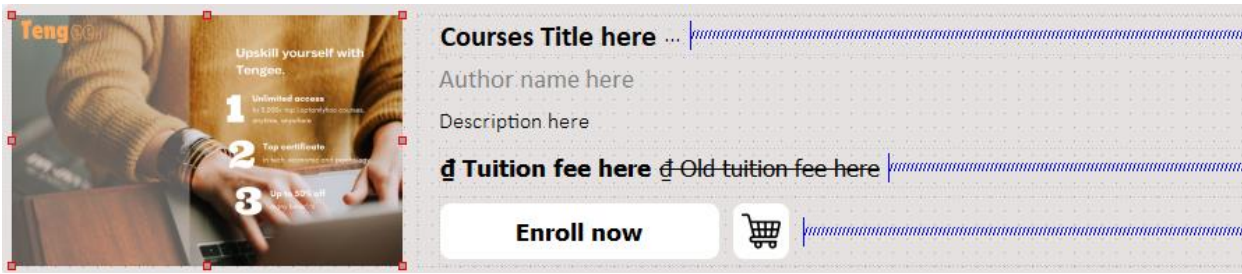
Loading Screen



Login window



## Card Layout



## Home window

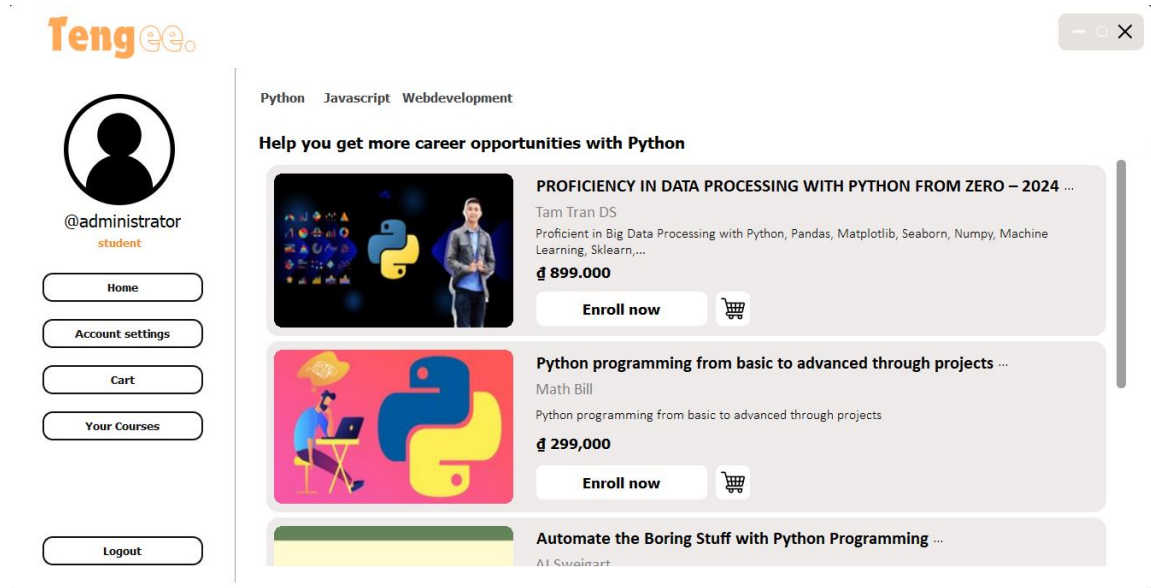


Figure 1 Home page

## Profile window

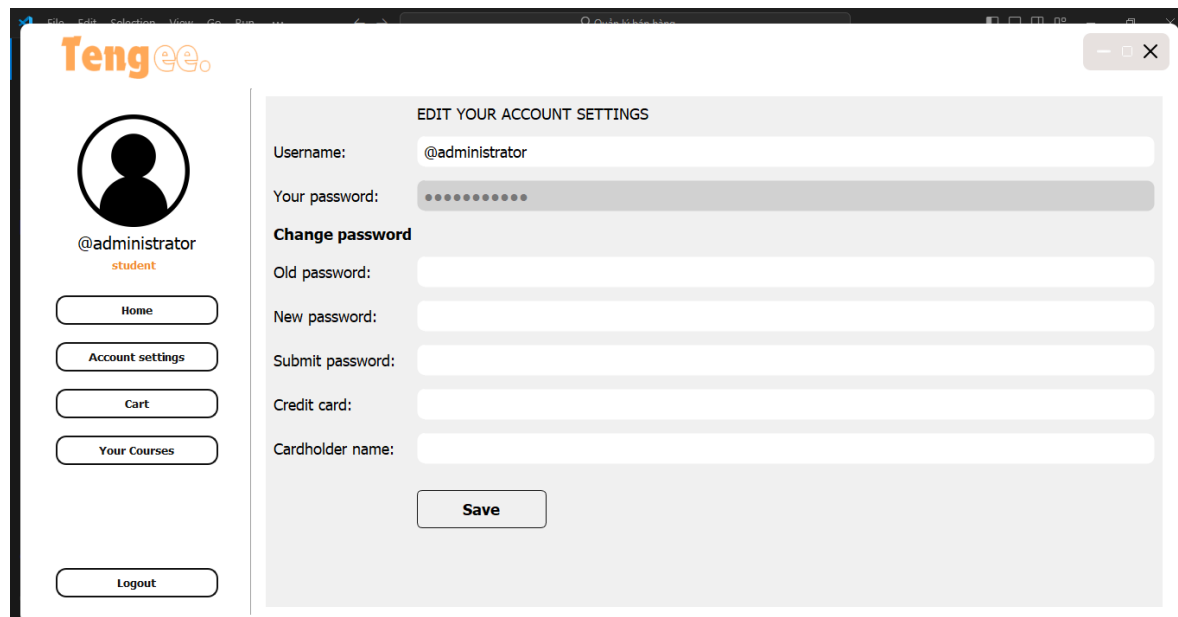


Figure 2 Profile settings page

## Courses management page:

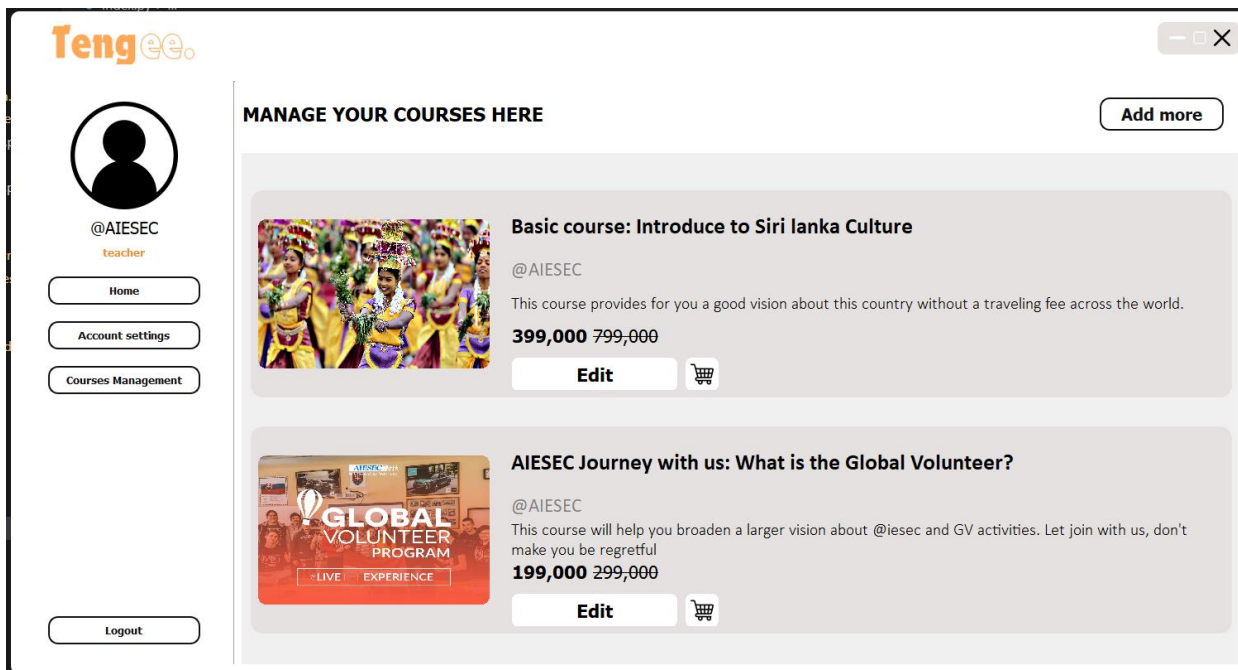


Figure 3 Courses management page

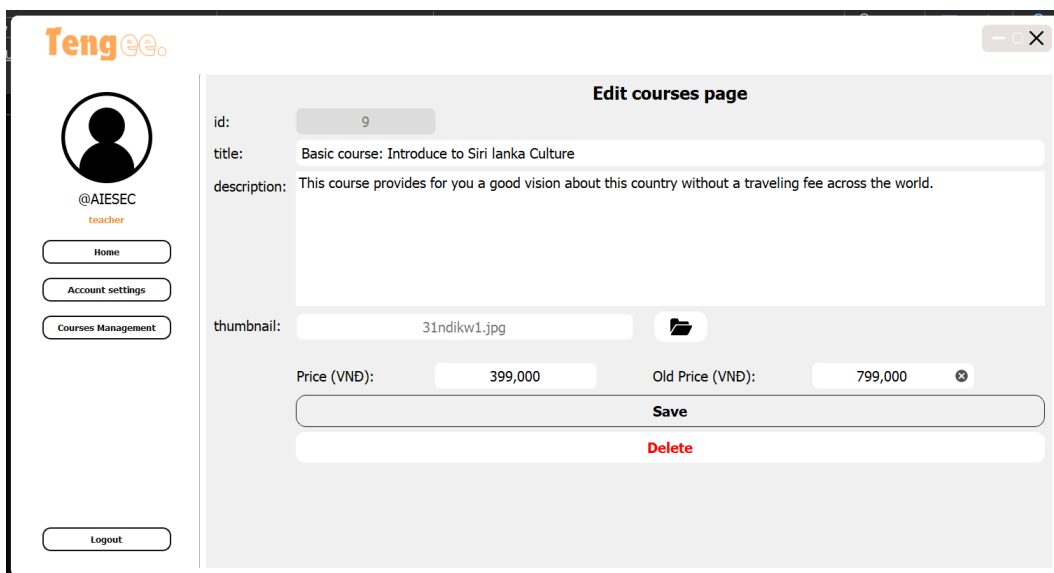


Figure 4 Courses Edit page < Edit button

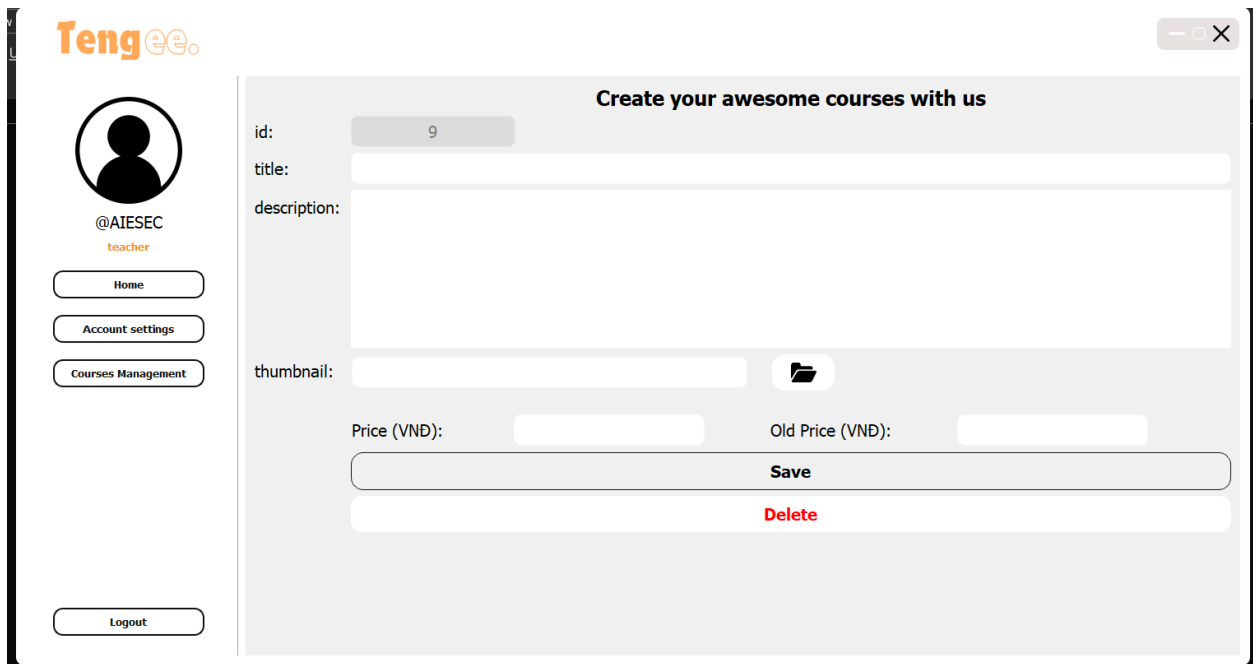
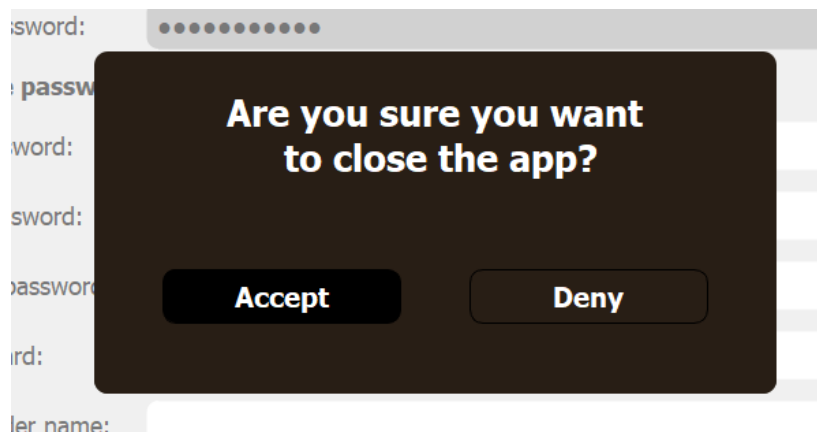



Figure 5 Create Course Page > Add course button

Exit Query window



TengGo



taikhoanteacher

teacher

Home

Account settings

Courses Management

Logout

PROFICIENCY IN DATA PROCESSING WITH PYTHON FROM ZERO – 2024

Proficient in Big Data Processing with Python, Pandas, Matplotlib, Seaborn, Numpy, Machine Learning, Sklearn,...

Created by


Tam Tran DS

Last updated

6/5/2024

English

Course content



₫ 899.000

Buy now

30-Day Money-Back Guarantee

**This course includes:**

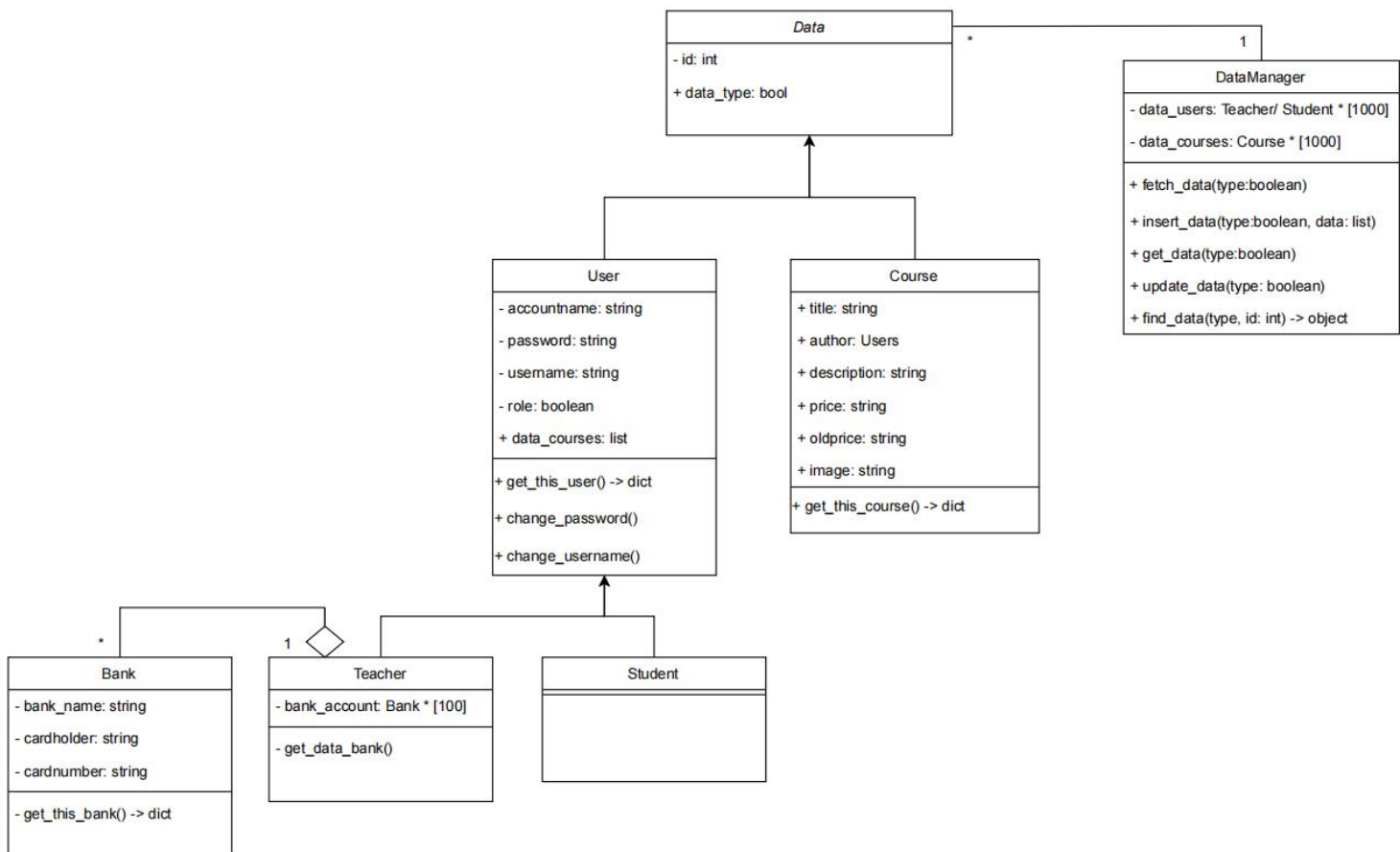
20 Lessons

Figure 6 Page course details

Trang 8



## 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: .....            | 10 |
| __init__(): .....             | 10 |
| fetch_data(type: bool): ..... | 11 |
| get_data(data_type): .....    | 11 |

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
| Data:.....                                                                        | 11 |
| __init__(id, data_type):.....                                                     | 11 |
| Course(Data):.....                                                                | 11 |
| __init__(id, data_type, title, author, description, price, oldprice, image) ..... | 11 |
| get_this_course().....                                                            | 12 |
| User(Data).....                                                                   | 12 |
| __init__(id, data_type, accountname, password, username, role).....               | 12 |
| get_this_user() .....                                                             | 12 |

## **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]