

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO ĐỒ ÁN**  
**LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG - IT002.O21.TTNT**

**ĐỀ TÀI**  
**TENGEE – APP QUẢN LÝ BÁN KHOÁ HỌC**

Giảng viên hướng dẫn ThS. Nguyễn Thanh Sơn  
Sinh viên thực hiện: Huỳnh Việt Tiến - 23521770

**TP. Hồ Chí Minh, tháng 06 năm 2024**

# MỤC LỤC NỘI DUNG

<b>I. GIỚI THIỆU ĐỒ ÁN</b>	4
<b>1. NỘI DUNG VÀ YÊU CẦU</b>	4
1.1 Nội dung	4
1.2 Yêu cầu	4
<b>2. SẢN PHẨM</b>	5
2.1 Link github	5
<b>II. QUÁ TRÌNH THỰC HIỆN</b>	5
1. Tuần 1:	5
2. Tuần 2:	6
3. Tuần 3:	6
4. Tuần 4:	7
5. Kết quả đạt được	8
6. Sản phẩm cuối cùng	14
7. Tài liệu tham khảo	14
<b>III. PHỤ LỤC DOCSTRING</b>	14
DataManager:	15
__init__():	15
insert_data(type, data):	15
fetch_data(type: bool):	15
get_data(data_type):	15
update_data(data_type, data):	15
find_data(self, typeData, id):	15
delete(self, data, type):	15
getID(self):	15
Data:	16
__init__(id, data_type):	16
Course(Data):	16
__init__(id, data_type, title, author, description, price, oldprice, image)	16

get_this_course().....	16
User(Data).....	16
__init__(id, data_type, accountname, password, username, role) .....	16
get_this_user().....	16
Teacher(User): .....	17
__init__(id, data_type, accountname, password, username, role) .....	17
insert_bank(self, bank_name, card_number, recipient): .....	17
fetch_bank(self): .....	17
Bank: .....	17
__init__(self, bank_name, recipient, cardnumber): .....	17
get_this_bank(self): .....	17
class Student(User): .....	17
__init__(self, id, data_type, accountname, password, username, role): .....	17

# I. GIỚI THIỆU ĐỒ ÁN

## 1. NỘI DUNG VÀ YÊU CẦU

### 1.1 Nội dung

#### a. Mục đích:

- Ứng dụng kiến thức của môn học Lập trình hướng đối tượng;
- Chạy thử nghiệm ý tưởng để giải quyết các vấn đề thực tiễn trong đời sống;

#### b. Ý tưởng:

- Tạo dựng một ứng dụng dùng để quản lý đăng bán các khoá học online. Tạo ra sự tương tác giữa người cung cấp khoá học với vai trò là Teacher (giáo viên) và người mua khoá học là Student (học sinh)

#### c. Ngôn ngữ lập trình, công cụ lập trình:

- Công cụ lập trình: Visual Studio Code, Qt Designer
- Ngôn ngữ lập trình: Python

### 1.2 Yêu cầu

#### a. Yêu cầu chung:

- phụ lục docstring cho từng lớp và phương thức được liệt kê trong sơ đồ lớp (UML) ✓
- có tính hướng đối tượng, kế thừa, đóng gói ✓
- có bản thiết kế UML ✓
- Code python theo bản thiết kế UML ✓

#### b. Yêu cầu riêng:

##### **hệ thống đăng ký/ đăng nhập**

Chia người dùng thành 2 loại là học sinh/ giáo viên (student/ teacher) ✓

##### **Trang cá nhân (Profile Page)**

Tất cả người dùng cung cấp thông tin cơ bản bao gồm: ✓

- accountname (tên tài khoản – const)
- username (tên người dùng)
- password (mật khẩu)

Teacher cung cấp thêm thông tin *tài khoản ngân hàng* vào **hệ thống profile** mới có thể tạo khoá học. ✓

### **Trang chủ (Home Page)**

Trang home yêu cầu hiển thị được các tính năng cơ bản: hiển thị khóa học từ dữ liệu, tìm kiếm khóa học, hiển thị các nút chuyển đổi giao diện, cửa sổ ✓

### **Trang quản lý khóa học (Courses Management Page)**

- Chỉ hiển thị với giao diện giáo viên ✓
- Cung cấp các tính năng quản lý cơ bản bao gồm đăng, sửa, và xóa khóa học ✓

### **Trang Thanh toán (Purchasing Page)**

- Cung cấp màn hình thông tin ngân hàng của tác giả khóa học ✓
- Thiết kế chức năng thông báo đơn hàng đến giáo viên
- Thiết kế chức năng thông báo đơn hàng đã được chấp nhận đến học sinh

### **Trang khóa học (Purchasing Page)**

- Hiển thị các khóa học đã mua và đã được xác nhận từ giáo viên cho học sinh

## **2. SẢN PHẨM**

### **2.1 Link github**

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

## **II. QUÁ TRÌNH THỰC HIỆN**

### **1. Tuần 1:**

- Các thư viện và module có sẵn được sử dụng trong tuần 1:
  - time: dùng cho màn hình loading
  - pyqt5 (QtCore, Qt, QtWidgets, UiC, QMainWindow,...): dùng cho tổng thể giao diện
- Thiết kế màn hình loading
- Tạo hệ thống logic đăng nhập, đăng ký tài khoản
  - Tạo cây thư mục dùng để lưu trữ các tệp dữ liệu
  - Thiết kế cửa sổ đăng ký đăng nhập
  - Thêm các điều kiện nhập khi đăng ký và kiểm tra độ chính xác khi đăng nhập.
- Tính năng remember me, ghi nhớ người dùng khi đóng ứng dụng
- Thiết kế màn hình home

- Thiết kế card khóa học
- Cách hiển thị dữ liệu khoá học:
  - Đọc tệp dữ liệu lưu khoá học đưa vào cấu trúc danh sách (list) dưới dạng các object Course bao gồm các thuộc tính cơ bản như tiêu đề, tác giả, mô tả, giá và hình ảnh.
  - Sử dụng thư viện giao diện tạo một Scrollbox (hộp cuộn), sử dụng vòng for để lần lượt đưa các object Course từ danh sách vào Scrollbox dưới dạng card khoá học.
- Thêm tính năng tìm kiếm thông tin khóa học (bao gồm tìm theo tiêu đề, tác giả và mô tả)
- Thêm các nút phân loại có sẵn để lọc khóa học theo từ khóa
- Thiết kế màn hình profile hiển thị thông tin người dùng và thêm các chức năng tương tác thông tin tài khoản bao gồm sửa tên, sửa mật khẩu, lưu thay đổi và đăng xuất
- Thiết kế cửa sổ xác nhận thoát ứng dụng Yes/ No nhằm tránh trường hợp ấn nhầm nút tắt làm gián đoạn tác vụ đang thực hiện
- Thiết kế sơ đồ lớp UML cơ bản
- Fix Bug:
  - Thêm tính năng vô hiệu hoá tương tác của người dùng khi hộp thoại xác nhận thoát ứng dụng xuất hiện, chỉ khi xác nhận thì mới huỷ vô hiệu hoá.

## 2. Tuần 2:

- Cải thiện sự rõ ràng của tính hướng đối tượng, kế thừa và đóng gói. Theo đánh giá và nhận xét của gv hướng dẫn, lược bỏ, bỏ qua các lớp UML không cần thiết của giao diện UI, thư viện và từ đó thiết kế lại sơ đồ UML đáp ứng được 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 **khóa học** và đều được đánh **mã số ID**. Mỗi **người dùng** sẽ có **tên tài khoản**, **mật khẩu**, **tên người dùng**, **vai trò**. Mỗi **khóa học** sẽ có **tên khóa 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) và danh sách các khóa học đã tạo. Học sinh sẽ bao gồm thông tin các khóa học đã mua.

Bổ sung file data\_init.py chứa tất cả các class mô tả vấn đề như trên ứng dụng tính đóng gói (public, private), tính kế thừa và quan hệ 1...\*, \*...1, 1...1, \*...\* và docstring

Chỉnh sửa nâng cấp tệp python ui để chạy với tệp data\_init.py

## 3. Tuần 3:

- Cập nhật thay đổi sơ đồ lớp:
  - Giới hạn quan hệ giữa lớp Teacher với lớp Bank (ngân hàng) từ 1..\* thành 1..1
  - Thêm và chỉnh các phương thức chính của cấu trúc dữ liệu bao gồm Fetch (nhập dữ liệu), Insert (thêm), Get (nhận dữ liệu), update (thay đổi thông tin tệp dữ liệu), find (tìm kiếm), delete (xóa) và hàm getID tăng giá trị của ID
  - Gộp khoá học đã tạo của giáo viên và khoá học của học sinh làm một đưa vào lớp cha, nếu dữ liệu vai trò là 1 thì là danh sách tạo, dữ liệu vai trò là 0 thì là danh sách mua.
  - Thêm chức năng cập nhật, thêm thông tin ngân hàng cho giáo viên insert\_bank vào lớp Teacher. Sau khi thêm thì mở khoá nút tạo khoá học (ban đầu vô hiệu hoá)
- Thiết kế màn hình quản lý khoá học của giáo viên  
 Cách hiển thị các khoá học đã tạo: Danh sách sở hữu khoá học của giáo viên được lưu trữ dưới dạng danh sách ID với key là id của giáo viên. Sử dụng phương thức tìm kiếm ở cấu trúc dữ liệu DataManager để tìm kiếm ID giáo viên, rồi hiển thị giá trị của thuộc tính data\_courses. Sau đó tương tự như cách hiển thị dữ liệu khoá học.
- Thiết kế màn hình edit khoá học.
- Thiết kế chức năng tạo, chỉnh sửa và xóa khoá học.
- Nghiên cứu cách đăng ảnh lên ứng dụng khi người dùng sử dụng ảnh trong đường dẫn local để làm ảnh của 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 thông tin chi tiết khoá học.
- Màn hình xác nhận thanh toán khoá học.

#### 4. Tuần 4:

## 5. Kết quả đạt được

### 5.1 Các bản thiết kế

#### Màn hình loading



#### Cửa sổ đăng nhập

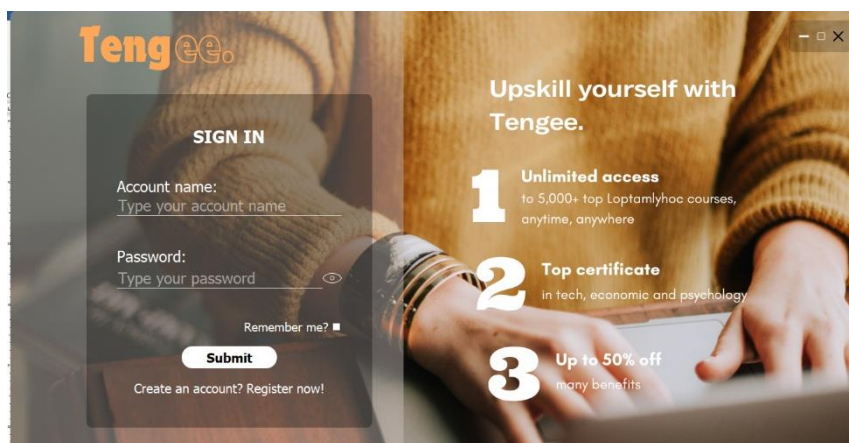


Figure 1Màn hình đăng nhập

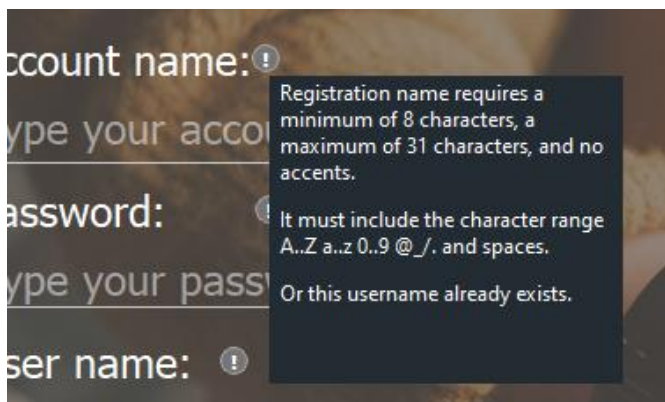


Figure 3 Tooltip điều kiện nhập

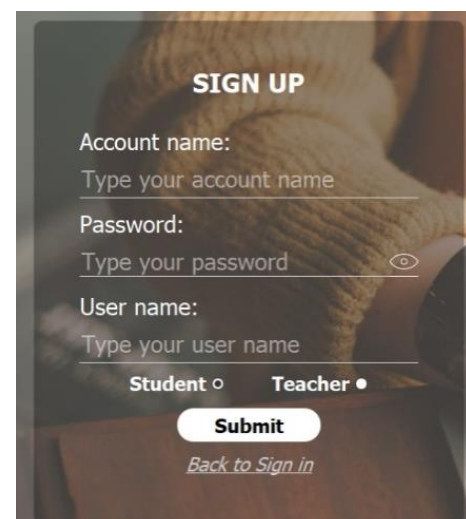



Figure 2 Tab đăng ký



## Thiết kế thẻ khoá học



**Courses Title here**

Author name here

Description here

**₫ Tuition fee here** ~~₫ Old tuition fee here~~

[Enroll now](#)[Edit](#)[More information](#)

## Màn hình trang chủ

Tengge



@administrator  
student

Home

Account settings

Cart

Your Courses

Logout

Python Javascript Webdevelopment

Help you get more career opportunities with Python



**PROFICIENCY IN DATA PROCESSING WITH PYTHON FROM ZERO – 2024 ...**

Tam Tran DS

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

**₫ 899.000**

[Enroll now](#)



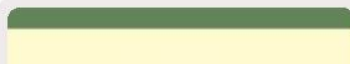
**Python programming from basic to advanced through projects ...**

Math Bill

Python programming from basic to advanced through projects

**₫ 299,000**

[Enroll now](#)



**Automate the Boring Stuff with Python Programming ...**

Al Sweigart

Figure 4 Home page

## ✚ Màn hình thông tin cá nhân

The screenshot shows a web application window titled 'Teng' with a logo. On the left is a sidebar with a user profile icon, the username '@administrator', the role 'student', and buttons for 'Home', 'Account settings', 'Cart', 'Your Courses', and 'Logout'. The main content area is titled 'EDIT YOUR ACCOUNT SETTINGS'. It contains fields for 'Username' (set to '@administrator'), 'Your password' (masked with dots), and a 'Change password' section with fields for 'Old password', 'New password', and 'Submit password'. There are also fields for 'Credit card' and 'Cardholder name', followed by a 'Save' button.

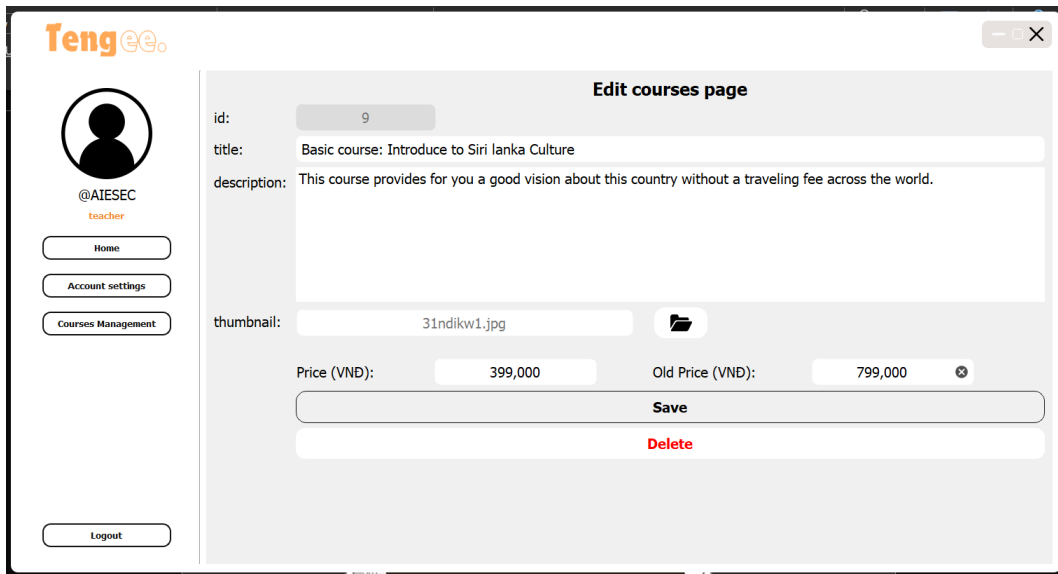
Figure 5 Profile settings page

## ✚ Màn hình quản lý khoá học đã tạo

The screenshot shows a web application window titled 'Teng' with a logo. On the left is a sidebar with a user profile icon, the username '@AIESEC', the role 'teacher', and buttons for 'Home', 'Account settings', 'Courses Management', and 'Logout'. The main content area is titled 'MANAGE YOUR COURSES HERE' and includes an 'Add more' button. It displays two course cards. The first card is for 'Basic course: Introduce to Siri lanka Culture' by '@AIESEC', with a description 'This course provides for you a good vision about this country without a traveling fee across the world.', a price of '399,000 799,000', and an 'Edit' button with a shopping cart icon. The second card is for 'AIESEC Journey with us: What is the Global Volunteer?' by '@AIESEC', with a description 'This course will help you broaden a larger vision about @iesec and GV activities. Let join with us, don't make you be regretful', a price of '199,000 299,000', and an 'Edit' button with a shopping cart icon.

Figure 6 Màn hình quản lý khoá học đã tạo

## Màn hình Sửa thông tin khoá học



**Teng**

**Edit courses page**

id: 9

title: Basic course: Introduce to Siri lanka Culture

description: This course provides for you a good vision about this country without a traveling fee across the world.

thumbnail: 31ndikw1.jpg

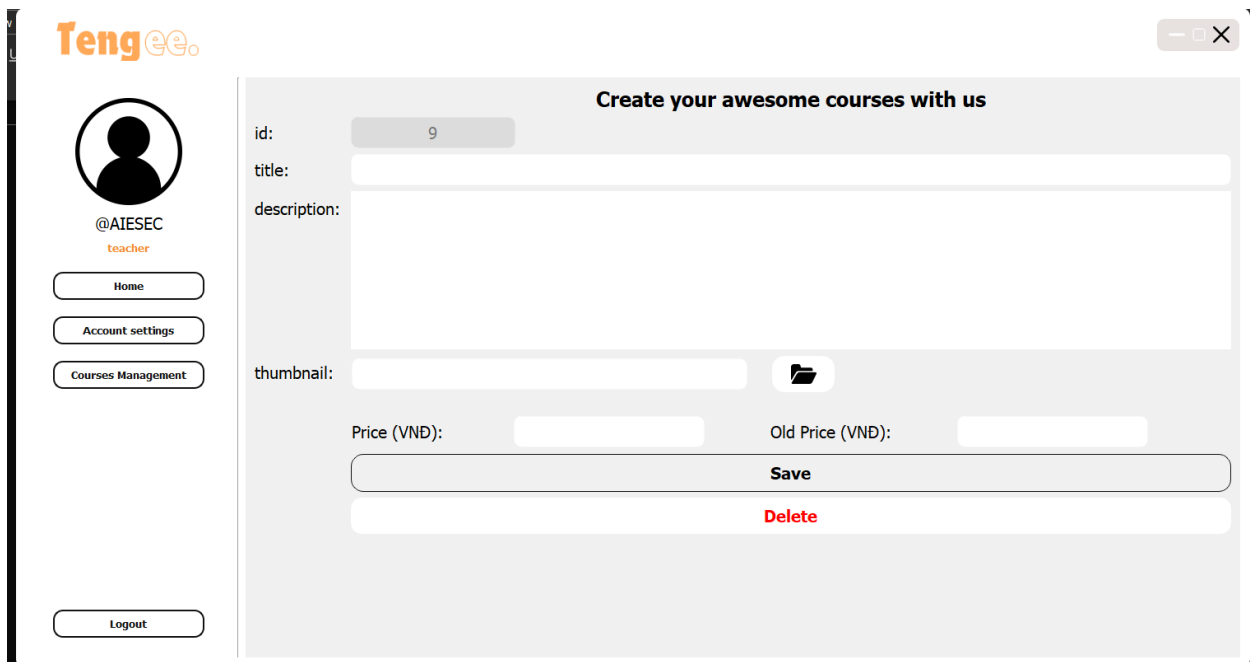
Price (VND): 399,000 Old Price (VND): 799,000

**Save**

**Delete**

Figure 7 Courses Edit page < Edit button

## Màn hình tạo khoá học



**Teng**

**Create your awesome courses with us**

id: 9

title:

description:

thumbnail:

Price (VND): Old Price (VND):

**Save**

**Delete**

Figure 8 Create Course Page > Add course button

🚦 Màn hình hiển thị chi tiết thông tin khoá học

The screenshot shows the course details page for 'PROFICIENCY IN DATA PROCESSING WITH PYTHON FROM ZERO – 2024'. The course is created by Tam Tran DS and was last updated on 6/5/2024. It is in English. The price is 899,000 VND. The course includes 20 lessons and a 30-day money-back guarantee. The user is logged in as 'taikhoanteacher' (teacher). The sidebar contains links for Home, Account settings, Courses Management, and Logout.

**TengEE**

**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

**₫ 899.000**

**Buy now**

30-Day Money-Back Guarantee

**This course includes:**

20 Lessons

**Course content**

**taikhoanteacher**  
teacher

Home

Account settings

Courses Management

Logout

🚦 Màn hình thanh toán khoá học <- Buy now button

The screenshot shows the payment page for the 'IELTS 9: Speaking Mastery' course. The course is created by Misa Zhang, Shani Raja and has a rating of 4.4 out of 5.44 (2,955 ratings). The price is 849,000 VND. The user is logged in as 'Tien Huynh' (student). The sidebar contains links for Home, Account settings, Your Courses, and Notifications. The payment page shows the card number, bank name, and recipient. The total amount is 849,000 VND. The user can confirm the payment.

**TengEE**

**Purchasing**

**IELTS 9: Speaking Mastery**

Superpower Your IELTS Speaking Score by Mastering Advanced Conversation Skills  
Rating: 4.4 out of 5.44(2,955 ratings)

Created by **Misa Zhang, Shani Raja**

**Banking information**

Card number: MISA ZHANG  
Bank name: momo  
Recipient: 0335903327

Bill Code: GkNm9SN9ntwdWzu  
Total: **₫849,000**

**Confirm**

**Tien Huynh**  
student

Home

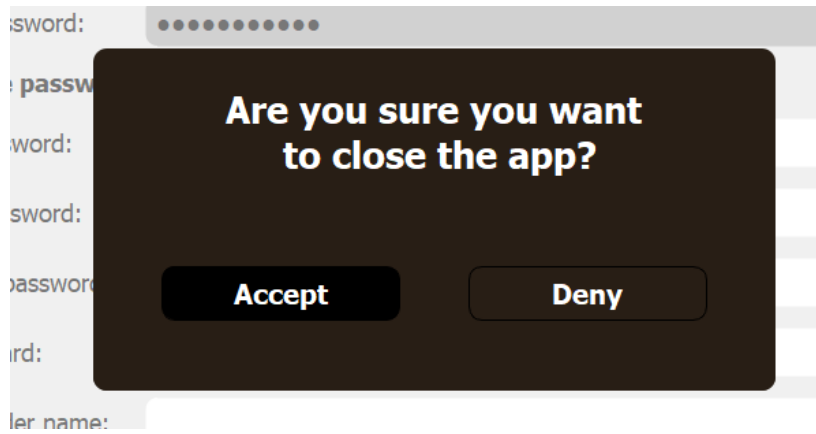
Account settings

Your Courses

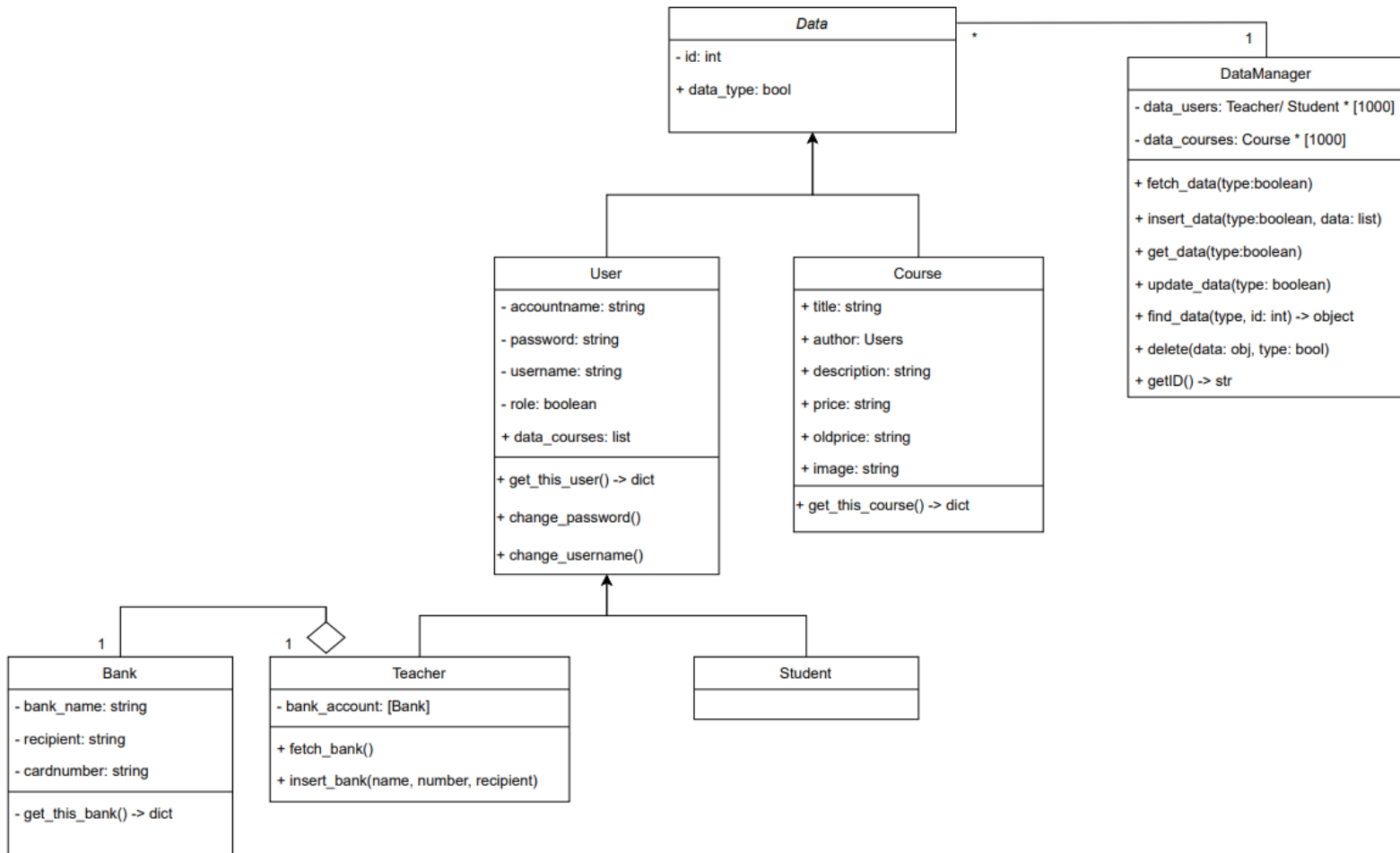
Notifications

Logout

🚦 Cửa sổ xác nhận thoát khỏi ứng dụng



## UML class diagram



### 6. Sản phẩm cuối cùng

### 7. Tài liệu tham khảo

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

## III. PHỤ LỤC DOCSTRING

## 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]

insert\_data(type, data):

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: [id, type, 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]

update\_data(data\_type, data):

update\_data(data\_type): update list of users or courses based on [data\_type]

find\_data(self, typeData, id):

find\_data(typeData, id): find data from data users [1] and data courses [0] based on [typeData] and [id].

If type = -1, find accountname of users

delete(self, data, type):

delete(data, type): function delete account or course based on [type] (bool)

getID(self):

getID(self): create new object's ID

## 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()

change\_password(data)

change\_username(data)

\_\_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 a dict {id, accountname, password, username, role}



### **Teacher(User):**

class Teacher: generate a data storing bank\_account (-> Bank) then inherits id, data type, accountname, password, username and role

method:

\_\_init\_\_(id, data\_type, accountname, password, username, role)

insert\_bank(bankname, card\_number, recipient)

fetch\_bank()

\_\_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

insert\_bank(self, bank\_name, card\_number, recipient):

insert\_bank(bank\_name, card\_number, recipient): insert new bank value to this Teacher account

fetch\_bank(self):

fetch\_bank(self): get bank account data

### **Bank:**

class Bank: generate a series data for bank account

method:

\_\_init\_\_(bank\_name, cardholder, cardnumber)

get\_this\_bank()

\_\_init\_\_(self, bank\_name, recipient, cardnumber):

\_\_init\_\_(bank\_name, cardholder, cardnumber): initiate attributes

get\_this\_bank(self):

get\_this\_bank(): return a dictionary of Bank information

### **class Student(User):**

class Student(User): inherits id, data type, accountname, password, username and role

method:

\_\_init\_\_(id, data\_type, accountname, password, username, role)

\_\_init\_\_(self, id, data\_type, accountname, password, username, role):

\_\_init\_\_(id, data\_type, accountname, password, username, role): inherits and initiates attributes