

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



KỸ THUẬT LẬP TRÌNH (CO1011)

Đề bài tập lớn 02 (**Phiên bản 1.0**)

CBGD: TS. LÊ Thành Sách
TS. NGUYỄN Đức Dũng

Tp. Hồ Chí Minh, Tháng 04/2017



Mục lục

1	Lịch sử các phiên bản	2
2	Yêu cầu của Bài tập lớn số 2	2
2.1	Về tính năng của phần mềm	3
2.2	Về lưu trữ và truy xuất dữ liệu	4
2.3	Về giao diện	5
2.4	Về quy trình thực hiện và báo cáo kết quả	5
3	Hướng dẫn thực hiện	5
3.1	Tổ chức nhóm	5
3.2	Quy trình thực hiện	5
3.2.1	Khảo sát và Phân tích yêu cầu của phần mềm	6
3.2.2	Thiết kế phần mềm	6
3.2.3	Tổ chức và quản lý mã nguồn trong quá trình phát triển	11
3.2.4	Thu thập số liệu	11
3.2.5	Kiểm tra phần mềm	11
3.2.6	Lập tài liệu hướng dẫn phát triển, triển khai và sử dụng	11
4	Báo cáo kết quả	12
5	Ngày báo cáo	12
6	Tiêu chí chấm bài	12

1 Lịch sử các phiên bản

Ngày	Phiên bản	Nội dung
28-04-2017	1.0	Công bố đề bài BTL Số 02

2 Yêu cầu của Bài tập lớn số 2

Giả sử có một thư viện cần một phần mềm (chương trình) để hỗ trợ các hoạt động diễn ra hằng ngày ở đó. Về quy mô, các sinh viên có thể chọn một thư viện thuộc Khoa, Trường, hay Quốc gia, tùy theo năng lực lập trình của các thành viên trong từng nhóm. Trong tài liệu này, chương trình đó được đặt tên là **LIBPRO**¹ để dễ tham chiếu.

Ở Bài tập lớn số 2 này, các sinh viên làm việc theo nhóm để cho ra sản phẩm của nhóm mình. Mỗi nhóm phải có số thành viên từ 3-4 người (Xem Phần 3.1).

Đề bài không đưa ra *yêu cầu* và *giải thuật* cụ thể đến mức theo đó các nhóm có thể lập trình được. Nó chỉ đưa ra các yêu cầu tối thiểu mà chương trình các nhóm **PHẢI** hỗ trợ cho người sử dụng. Như vậy, sản phẩm của các nhóm là khác nhau. Số lượng tính năng và chất lượng của sản phẩm phụ thuộc nhiều vào sự tích cực, kỹ năng lập trình và sự sáng tạo của các thành viên.

Yêu cầu đầu tiên trong bài tập lớn này là: các nhóm cần tách biệt 3 khái niệm sau: (a) người sử dụng, (b) tài khoản, và (c) vai trò.

1. Người sử dụng (user):

- Người sử dụng, còn được gọi ngắn gọn là người dùng, là những cá nhân được quản lý bởi hệ thống **LIBPRO**. Cũng có nghĩa là, thông tin về họ được **LIBPRO** lưu trữ và quản lý. Họ cũng có tài khoản (gồm tên tài khoản và mật khẩu) để đăng nhập và sử dụng các tính năng mà phần mềm cung cấp cho họ.
- Thông tin tối thiểu mà phần mềm phải lưu trữ cho người dùng là: **mã số** (để định danh họ trong hệ thống, việc này tương tự như mỗi sinh viên đều được cấp một mã số để nhà trường định danh); **số chứng minh nhân dân** hoặc **mã số sinh viên** (dùng để định danh họ trong xã hội hoặc tổ chức); **họ và tên**; **ngày, tháng, và năm sinh**; **nghề nghiệp** (có thể là sinh viên, giảng viên, v.v); **địa chỉ email**.

2. Tài khoản (account):

- Trong thực tế, nhiều phần mềm thư viện có cung cấp những tính năng mà bất kỳ ai cũng có thể sử dụng mà không cần đến việc đăng nhập vào hệ thống. Tuy nhiên, cũng có nhiều tính năng người sử dụng phải đăng ký trước để có tài khoản và dùng tài khoản đăng nhập vào hệ thống, họ mới có thể dùng được. Một tài khoản bao gồm ít nhất các thông tin như: **tên tài khoản** (có thể khác mã số nhân viên), **mật khẩu**, và **trạng thái** (được mở (active) hay bị khóa (disable)).

3. Vai trò (role):

- Vai trò là khái niệm dùng để chỉ ra một nhóm tính năng của người sử dụng trong hệ thống.

¹Nhiều chỗ trong tài liệu này, **LIBPRO** còn được gọi là **hệ thống**, **chương trình**, và **phần mềm** để tránh sự lặp lại

- Một số vai trò trong hệ thống này mà sinh viên PHẢI hỗ trợ là: **thủ thư, độc giả, quản lý người dùng**. Xem Phần 2.1 để rõ hơn về các tính năng tương ứng với vai trò.

Hệ thống **LIBPRO** của các nhóm cần cho phép một người dùng có thể có một hoặc nhiều tài khoản. Mỗi tài khoản có thể được cấp một hoặc nhiều vai trò. Mỗi vai trò có thể được cấp cho một hoặc nhiều tài khoản.

Ví dụ, thư viện có 4 người dùng sau: **NHẬT, NGUYỆT, THẦN, và GIÁO**.

- Anh **NHẬT** chỉ có một tài khoản với tên là "**minh.nhat**". Tài khoản "**minh.nhat**" cũng được cấp cho vai trò là "Độc giả". Do đó, anh **NHẬT** là độc giả, và chỉ có thể thực hiện các tính năng được hệ thống **LIBPRO** cung cấp cho độc giả.
- Chị **NGUYỆT** có hai tài khoản. Một tài khoản có tên là "**minh.nguyet.a**", và một còn lại có tên là "**minh.nguyet.b**". Tài khoản "**minh.nguyet.a**" được cấp cho hai vai trò là "Thủ thư" và "Quản lý người dùng". Còn tài khoản "**minh.nguyet.b**" được cấp cho vai trò là quản lý tài chính (vai trò này là không bắt buộc hỗ trợ, không được giải thích chi tiết trong Phần 2.1) của tổ chức. Do vậy, cũng chỉ là một người dùng, nhưng chị **NGUYỆT** có 3 vai trò khác nhau trong tổ chức của mình; chị cũng có thể thực hiện được 3 nhóm tính năng tương ứng. Tuy nhiên, để thực hiện nhóm tính năng tương ứng với vai trò quản lý tài chính thì chị phải thoát khỏi hệ thống, nếu đang đăng nhập với tài khoản "**minh.nguyet.a**" và đăng nhập lại với "**minh.nguyet.b**". Đây cũng là hình thức mà tổ chức giúp chị chú ý hơn với vai trò mới trong hệ thống.
- Còn chị **THẦN** thì được cấp một tài khoản là "**lam.y.than**". Vai trò "Quản lý người dùng" cũng được cấp cho tài khoản "**lam.y.than**". Do đó, chị **THẦN** chỉ có thể thực hiện cụm các tính năng liên quan đến Quản lý người dùng.
- Tương tự anh **GIÁO** chỉ là một độc giả, thông qua một tài khoản được cấp là "**minh.giao**".

2.1 Về tính năng của phần mềm

Phần này trình bày những tính năng **bắt buộc** dành cho các vai trò mà hệ thống **LIBPRO** của các nhóm **PHẢI** hỗ trợ. Những gợi ý để thực hiện cũng được nêu ra, nhưng các nhóm có thể làm khác, miễn sao hệ thống **LIBPRO** vẫn hỗ trợ những tính năng theo yêu cầu (được in đậm)

1. **Độc giả:**

- **Tìm sách:** Để tìm, người dùng có thể nhập vào cụm các từ (chưa cần hỗ trợ Tiếng việt có dấu). Chương trình sẽ tìm các tựa sách trong phần tựa đề hoặc phần mô tả ngắn về sách có chứa cụm từ này. Nếu danh sách trả về quá dài, chương trình phải tự phân trang hoặc tự tìm hoặc cách hiển thị nào khác sao cho thuận tiện cho người dùng. Để so trùng văn bản nâng cao, nên tìm hiểu về khoảng cách chuỗi ² hoặc khoảng cách Levenshtein ³.

²https://en.wikipedia.org/wiki/String_metric

³https://en.wikipedia.org/wiki/Levenshtein_distance

- **Gửi yêu cầu mượn sách:** Dựa trên kết quả tìm kiếm, các độc giả có thể yêu cầu mượn những cuốn sách mà họ quan tâm. Lúc này, hệ thống phải yêu cầu người dùng đăng nhập với tài khoản họ đã đăng ký. Nhưng nếu họ đã đăng nhập rồi và chưa thoát ra thì không cần yêu cầu đăng nhập lại. Hệ thống có thể nhận thông tin thêm về yêu cầu mượn như ngày bắt đầu, ngày trả, vân vân. Mỗi khi người dùng đã đăng nhập vào hệ thống, họ có thể chuyển qua lại giữa các màn hình (dựa trên các menu của hệ thống). Trong suốt thời gian đó, họ có thể chọn tùy ý các đối tượng (sách) mà họ quan tâm và bỏ vào giỏ (như xe đẩy mua hàng trong siêu thị). Các nhóm có thể xem xét tính năng "giỏ hàng" như vậy. Xem thêm về những hệ thống mua sắm trực tuyến, như Sendo ⁴, để nắm thêm về khái niệm này.
- **Xem thông báo:** Sau khi thực hiện đăng nhập, các độc giả có thể xem được thông báo mà thư viện gửi đến họ. Các thông báo này có thể là: thông báo về sách mới về, thông báo yêu cầu trả sách sắp/đã quá hạn mượn.
- **Xem thông tin:** Sau khi thực hiện đăng nhập, các độc giả có thể xem được lịch sử các lần đã mượn và trả sách, thậm chí là bị phạt. Những sách mà độc giả chưa trả cho thư viện cũng có thể xem được khi chọn chức năng này. Nhắc lại, hình thức triển khai các menu và thiết kế tương tác, thiết kế giao diện là sáng tạo của sinh viên.
- **Đổi mật khẩu:** Hệ thống phải cho phép các độc giả thay đổi mật khẩu của họ. Tính năng này cũng được cung cấp cho tất cả các vai trò khác.

2. Quản lý người sử dụng:

- Cụm các tính năng liên quan đến quản lý thông tin người dùng là: **Thêm vào, Chính sửa, Xoá khỏi hệ thống, và Tìm kiếm**.
- Cụm tính năng liên quan đến quản lý tài khoản là **Thêm vào tài khoản** và liên kết tài khoản với người dùng tương ứng; **Mở và Khoá tài khoản**; **Lập lại mật khẩu** (reset) cho tài khoản; **Tìm kiếm tài khoản** theo tên tài khoản, hay tên người dùng được liên kết đến. Về mật khẩu, ứng dụng của các nhóm có thể lưu lại dưới tập tin dạng văn bản [đọc được](#). Tuy nhiên, trong thực tế thì không nên làm như vậy, phải mã hoá trước, và chỉ lưu lại chuỗi dữ liệu sau khi mã hoá ([đọc không được ra nghĩa](#)).

3. Thủ thư:

- Cụm các tính năng liên quan đến quản lý sách và các thông tin liên quan đến sách (ví dụ, tác giả và nhà xuất bản) là: **Thêm vào, Chính sửa, Xoá khỏi hệ thống, và Tìm kiếm**.
- Cụm tính năng liên quan đến giải quyết yêu cầu của độc giả là **Ghi nhận mượn/trả sách, phạt vì quá hạn, giải quyết việc làm thất lạc sách**.

2.2 Về lưu trữ và truy xuất dữ liệu

Hệ thống **LIBPRO** cần lưu trữ dữ liệu xuống đĩa cứng cũng như **đọc** dữ liệu từ đó.

Ví dụ, hệ thống **LIBPRO** cần lưu trữ danh sách các đối tượng (sách, người dùng, v.v) xuống đĩa. Do đó, hệ thống có thể bị tắt đi mà dữ liệu để hoạt động thì vẫn còn. Mỗi khi được mở lại, nó đọc dữ liệu dưới đĩa và lưu vào bộ nhớ trong (RAM). Với dữ liệu trong RAM nó có thể hỗ trợ việc tìm kiếm, ví dụ, tìm kiếm sách.

⁴<https://www.sendo.vn>

Chương trình của các nhóm **PHẢI** có khả năng lưu lại tất cả các loại dữ liệu cần trong quá trình làm việc. Các sinh viên có thể chọn lưu dưới dạng tập tin văn bản hoặc tập tin nhị phân ⁵.

2.3 Về giao diện

Chương trình của các nhóm phải có khả năng giúp người dùng chuyển qua lại giữa các màn hình. Chương trình chỉ kết thúc khi người dùng chọn menu "Dừng/Thoát/Stop" từ màn hình chính (không nên là màn hình con).

Ví dụ, khi độc giả chọn tìm sách từ màn hình chính (gọi là màn hình 1), **LIBPRO** có thể xoá màn hình 1, in ra menu yêu cầu nhập từ khoá để tìm kiếm (gọi là màn hình 2). Nếu người dùng nhập chuỗi và nhấn ENTER thì ứng dụng xoá màn hình 2 và in ra danh sách kết quả (màn hình 3), vân vân. Người dùng có thể chuyển ngược từ màn hình 3 sang 2, rồi sang màn hình chính. Tại đây, họ có thể dừng chương trình, nếu muốn.

2.4 Về quy trình thực hiện và báo cáo kết quả

Tiêu chí chấm bài (xem Phần 6) không chỉ quan tâm chương trình chạy minh hoạ, mà còn quan tâm đến quy trình thực hiện, mã nguồn, nội dung và hình thức của cuốn báo cáo.

Các nhóm phải tuân thủ quy trình như gợi ý trong Phần 3.2 để có kết quả tốt.

3 Hướng dẫn thực hiện

3.1 Tổ chức nhóm

Bài tập lớn này yêu cầu các sinh viên làm việc theo nhóm. Mỗi nhóm có từ 3 đến 4 thành viên. Các thành viên trong cùng nhóm phải cùng lớp thí nghiệm (LAB). Các trợ giảng sẽ triển khai việc thành lập nhóm và ghi nhận danh sách.

Một số gợi ý cho việc thành lập nhóm:

- Một nhóm không thể gồm chỉ toàn các thành viên thụ động hoặc điểm Bài tập lớn số 1 đều thấp. Tương tự, một nhóm không thể chỉ toàn các thành viên có khả năng lập trình tốt.
- Các sinh viên thấy mình có khả năng lập trình nên chọn đi kèm các sinh viên có khả năng thấp hơn.
- Một nhóm phải có trưởng nhóm. Trưởng nhóm chịu trách nhiệm quản lý lịch họp nhóm và phân chia công việc cho các thành viên. Nên sử dụng Doodle ⁶ để thu xếp lịch, sử dụng Google Drive để đồng bộ các tài liệu trong nhóm, và sử dụng Group trên Viber để liên lạc. Các nhóm có thể sử dụng hình chụp, lịch sử Doodle, và thông tin Viber để làm minh chứng cho việc nhóm có làm việc và phối hợp tốt.

3.2 Quy trình thực hiện

Kết quả các bước của quá trình thực hiện sẽ được lập thành tài liệu trong cuốn báo cáo cuối cùng. Do đó, các nhóm cần phải tuân thủ tiến hành các bước tại đây để có kết quả báo cáo tốt.

⁵Các giảng viên sẽ giới thiệu kỹ thuật này trên lớp

⁶<http://doodle.com>

3.2.1 Khảo sát và Phân tích yêu cầu của phần mềm

Phần 2 chỉ nêu ra những yêu cầu rất cơ bản và ở mức tổng quát mà một hệ thống **LIBPRO** PHẢI hỗ trợ, các nhóm cần tự thực hiện các công việc tiếp theo để làm rõ hơn, cũng như bổ sung thêm hay hạn chế lại các yêu cầu để có thể phát triển được ⁷.

Thực tế, để phát triển được phần mềm đúng yêu cầu của tổ chức nào đó, sản phẩm phải tuân thủ những quy định và những yêu cầu rất cụ thể.

Ví dụ, có thể tổ chức yêu cầu phần mềm chạy trên Hệ điều hành Windows hay Linux, giao tiếp được qua di động, vân vân. Cũng có tổ chức yêu cầu cụ thể, một độc giả chỉ có thể mượn (giữ) tối đa ba hay năm cuốn sách mà thôi. Độc giả phải trả lại mới được mượn tiếp. Khi họ trả trễ thì bị phạt, số tiền phạt bao nhiêu cũng tùy quy định. Khi họ làm mất sách thì đền bù, số tiền đền bù được tính ra sao cũng tùy quy định, vân vân.

Trên đây là những ví dụ về yêu cầu. Trong bản mô tả của bài tập lớn này đã **không nêu ra cụ thể**. Các nhóm có thể vận dụng những cách sau đây để có được yêu cầu cụ thể hơn.

- **Phỏng vấn:** Bằng những mối quan hệ của các thành viên trong nhóm (nếu có), các nhóm tự tìm ra người phù hợp có thể giải đáp những thắc mắc liên quan nghiệp vụ của một thư viện cụ thể. Nếu không có mối quan hệ nào, các nhóm có thể liên hệ (email và hoặc gặp trực tiếp) cô **Trần Thị Thu Trang** (nhân viên thư viện của Khoa để hỏi thêm thông tin). Nếu phỏng vấn, phải có bảng câu hỏi trước, ghi lại các ý hỏi được. Trong báo cáo có ghi rõ họ tên người trả lời, chức vụ và đơn vị cộng tác; cộng thêm là thời gian và địa điểm phỏng vấn.
- **Khảo sát các ứng dụng tương tự về thư viện:** Các bạn có thể tìm hiểu về tính năng và quy định của các tổ chức thư viện bằng cách tìm trên mạng. Ví dụ như, **Thư viện Trung tâm DHQG-Tp.HCM**. Trong báo cáo phải trích dẫn nguồn gốc thông tin đọc được.
- **Bằng cách thảo luận nhóm:** Các thành viên có thể đứng trên quan điểm của độc giả cũng có thể cung cấp các tính năng tốt cho vai trò độc giả. Cũng từ đó suy ra các tính năng của các vai trò khác để hỗ trợ độc giả. Báo cáo phải thể hiện rõ nội dung thảo luận và kết quả thống nhất của nhóm trong buổi họp đó.

3.2.2 Thiết kế phần mềm

Ở khâu này, các nhóm phải tạo ra những bản thiết kế như sau:

1. Thiết kế các cấu trúc dữ liệu dùng để phát triển phần mềm thư viện:

Một số ví dụ cụ thể sau.

- Để lưu trữ được thông tin về người dùng \Rightarrow phải có cấu trúc (**struct**) và hoặc lớp (**class**) mô tả kiểu dữ liệu mới - kiểu người dùng. Mã nguồn mẫu có thể là,

```
1 typedef struct {  
2     string user_no;  
3     string user_name;  
4     //more  
5 } User;
```

Hoặc có thể,

⁷Với tinh thần có điểm còn hơn không :)

```
1  class User{
2  private:
3      string user_no;
4      string user_name;
5      //more
6  };
```

Nội dung mô tả thông tin người dùng ở file dạng văn bản có thể như sau:

```
001; NGUYEN MINH NHAT; (các trường có thể cách nhau bởi dấu ;)
002; LE THI MINH NGUYET;
```

- Một người dùng có nhiều tài khoản, nhưng một tài khoản phải thuộc một và chỉ một người dùng \Rightarrow phải lưu kèm danh sách tài khoản vào người dùng. Mã nguồn mẫu có thể là,

```
1  class Account{
2  private:
3      string account_no;
4      string password;
5      bool active;
6      string user_no; //link to user
7  };
8  class User{
9  private:
10     string user_no;
11     string user_name;
12
13     vector<Account*> account_list;
14     //vector<Account> account_list; //possibly
15 };
```

Nội dung mô tả thông tin tài khoản ở file dạng văn bản có thể như sau. Ở đó, "xxx" là mật khẩu - các nhóm có thể mã hoá hay không, và "002" là mã số nhân viên "Lê Thị Minh Nguyệt" trong hệ thống.

```
minh.nguyet.a; xxx; true; 002
```

- Một ví dụ khác, một tài khoản thì có nhiều vai trò, một vai trò cũng được cấp cho nhiều tài khoản \Rightarrow không thể lưu kèm như nói trên (trường hợp của [Account](#) và [User](#)), mà phải lưu riêng một bảng ánh xạ giữa tài khoản và vai trò được ánh xạ. Điều này cũng có nghĩa nên tạo riêng một cấu trúc ([struct](#)) hoặc lớp ([class](#)) cho bảng ánh xạ này. Lớp này có thể có tên "AccountRoleMap". Trong cấu trúc hay lớp này phải có ít nhất hai trường thông tin là "tên tài khoản" và "mã vai trò" để thể hiện việc ánh xạ cho nhau. Mã nguồn mẫu có thể là,

```
1  class Account{
2  private:
3      string account_no;
```



```
4         string password;  
5         bool active;  
6         string user_no; //link to user  
7     };  
8     class User{  
9     private:  
10        string user_no;  
11        string user_name;  
12  
13        vector<Account*> account_list;  
14        //vector<Account> account_list; //possibly  
15    };  
16    class Role{  
17    private:  
18        string role_id;  
19        string role_desc;  
20    };  
21    class AccountRoleMap{  
22    private:  
23        string account_no;  
24        string role_id;  
25    }
```

Nội dung mô tả thông tin vai trò và bảng ánh xạ tài khoản và vai trò ở file dạng văn bản có thể như sau. Tập tin danh mục vai trò có thể là,

THU_THU; là thủ thư trong hệ thống
QUAN_LY_NGUOI_DUNG; có trách nhiệm quản lý người dùng và tài khoản

Tập tin ánh xạ tài khoản và vai trò có thể là,

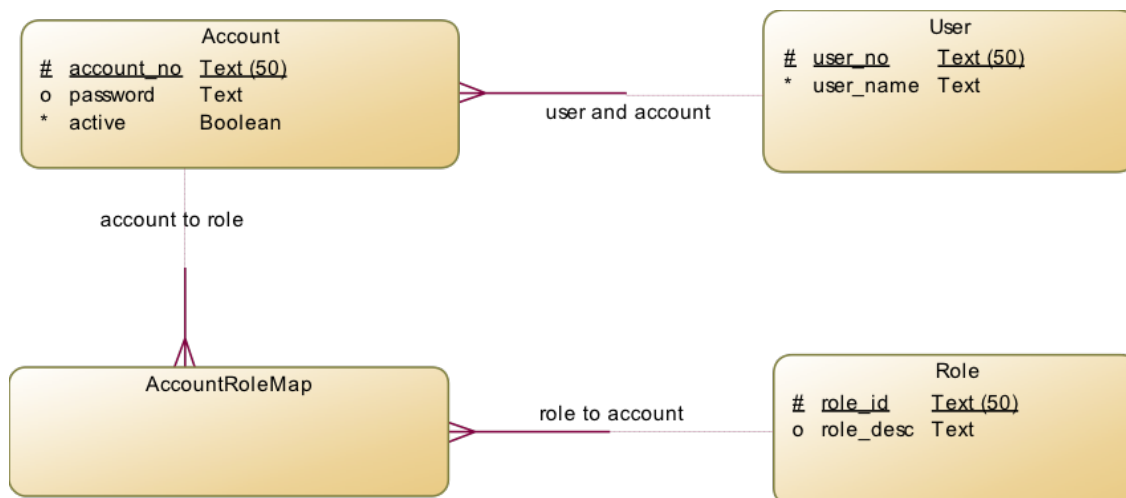
minh.nguyet.a; THU_THU
minh.nguyet.a; QUAN_LY_NGUOI_DUNG

Lưu ý, các nhóm có thể sử dụng công cụ Visio của Microsoft, PowerDesigner, hay Rational Rose để tạo sơ đồ thiết kế chuyên nghiệp hơn. Xem xét ví dụ sau. Quan hệ giữa các khái niệm **User**, **Account**, **Role**, và **AccountRoleMap** về mặt các trường dữ liệu cần thiết phải lưu xuống tập tin được thể hiện bằng sơ đồ trong các Hình 1 và Hình 2. Tổng lược các biến thành viên và các phương thức thành viên của các lớp **User** và **Account**, cũng như quan hệ giữa chúng được thể hiện trong Hình 3.

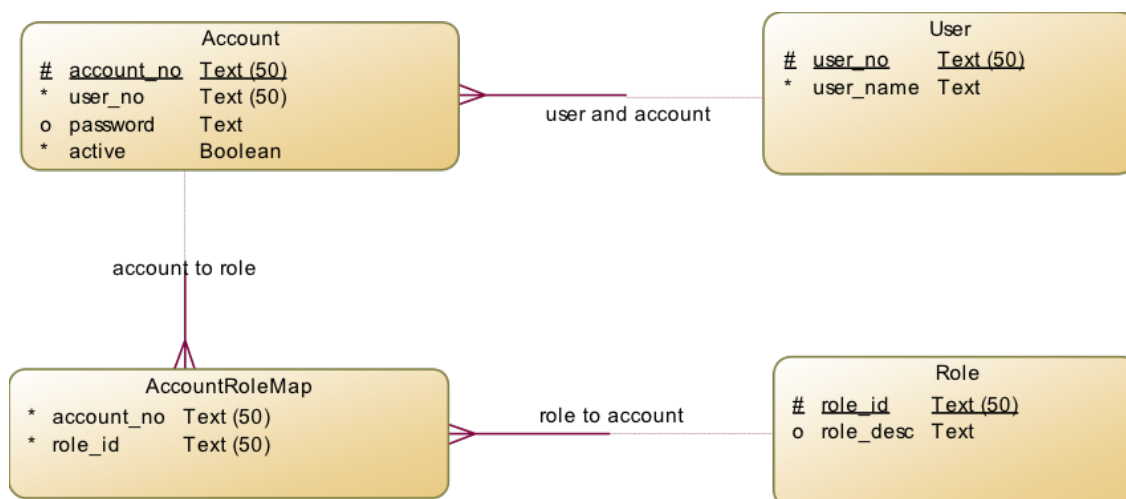
Nếu nhóm nào không thể tạo các hình như vậy thì thể hiện bằng code hay một dạng nào đó mà nhóm nghĩ ra để người đọc có thể hiểu được: (a) các nhóm có dùng đến các lớp (cấu trúc) nào; (b) các lớp đó có biến và phương thức thành viên nào, được khai báo ra sao; (c) quan hệ giữa các lớp là gì; (d) về mặt dữ liệu cần lưu xuống tập tin, các trường dữ liệu nào được lưu cùng nhau vào file (xem Hình 2).

2. Thiết kế các khối chức năng và hệ thống con:

- Nhiều ứng dụng trong thực tiễn về quản lý thư viện, các nhóm vai trò sẽ có hệ thống con riêng để phục vụ. Ví dụ, các bạn ở nhà có thể truy cập Web hay dùng điện thoại



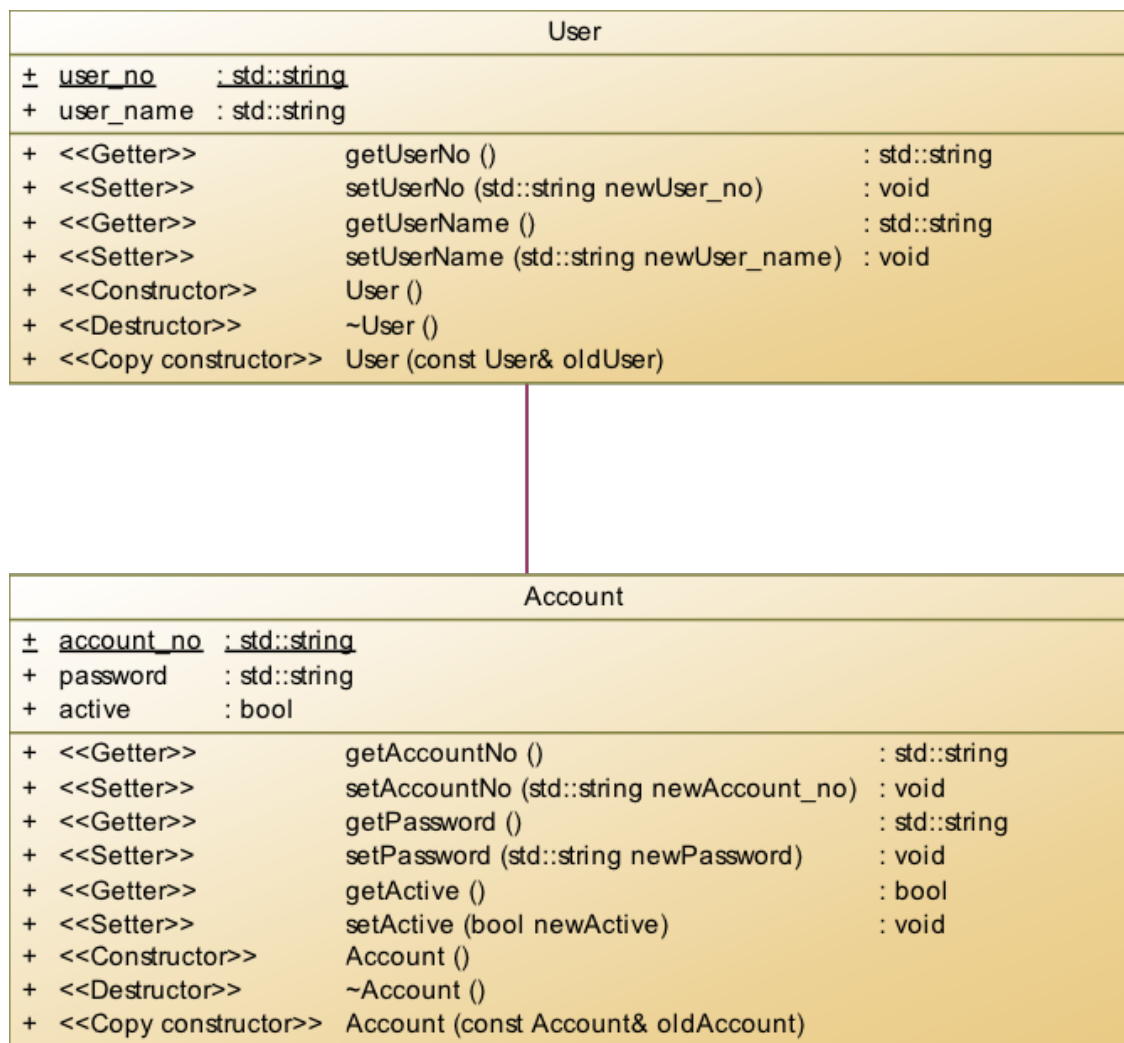
Hình 1: Sơ đồ quan hệ giữa các khái niệm: **User**, **Account**, v.v. về mặt dữ liệu sẽ được lưu xuống tập tin - nhìn ở mức cao nhất



Hình 2: Sơ đồ quan hệ giữa các khái niệm: **User**, **Account**, v.v. về mặt dữ liệu sẽ được lưu xuống tập tin - nhìn ở mức thấp nhất (giống như việc lưu xuống tập tin - xem các file dữ liệu ở trên)

để tìm sách và đăng ký mượn. Kiến thức của môn học này là chưa đủ để giúp phát triển ứng dụng với yêu cầu như vậy. Các nhóm có thể chọn hình thức giả lập. Các nhóm có thể dùng một cửa sổ chung để tất cả mọi người dùng đăng nhập và làm việc. Các nhóm cũng có thể chọn tách riêng ra các ứng dụng con khác nhau cho mỗi vai trò, hoặc nhóm các vai trò. Các ứng dụng con này sẽ giao tiếp thông qua dữ liệu được ghi nhận ở tập tin. Ví dụ, khi đọc giả đăng ký mượn sách, thì hệ thống ghi nhận vào tập tin. Khi thủ thư đăng nhập và yêu cầu, hệ thống có thể đọc từ tập tin để có danh sách các yêu cầu mượn sách.

- Các nhóm trình bày giải pháp của mình, tự thiết kế các khối và hệ thống con nếu có thể, cũng như nêu ra quan hệ giữa các khối.
- Kết quả của bước này sẽ cho ra danh sách các khối, danh sách thư mục mã nguồn (header và source) hiện thực các khối, danh sách các dự án con (nếu có), và **bản vẽ**



Hình 3: Sơ đồ quan hệ giữa các cấu trúc dữ liệu (lớp) cho trường hợp lớp **User** và **Account** sơ đồ các khối, vắn vắn.

Lưu ý - nhắc lại, các nhóm có thể sử dụng công cụ Visio của Microsoft, PowerDesigner, hay Rational Rose tạo sơ đồ thiết kế chuyên nghiệp hơn.

3. Thiết kế giao diện:

- Với kiến thức của môn học, các nhóm chỉ có thể tạo giao diện tương tác với người dùng thông qua bàn phím. Các nhóm có thể tạo danh sách menu cho mỗi tài khoản (dựa trên tài khoản đó được ánh xạ vai trò gì, và vì vậy đợt đăng nhập đó người dùng có thể dùng được tính năng gì). Mỗi khi người dùng chọn một menu, chương trình hiển thị màn hình menu (hay kết quả) mới.
- Theo cách đó, thì người dùng có thể chuyển qua lại giữa các màn hình để hoàn thành công việc. Chương trình chỉ kết thúc khi họ muốn thoát ra ở màn hình chính.
- Danh sách các màn hình tương tác và sự thân thiện với người sử dụng của nó hoàn toàn phụ thuộc vào sự sáng tạo và khả năng của nhóm. Với những nhóm mà có thành

viên **đặc biệt tốt**, nhóm đó có thể sử dụng các thư viện hỗ trợ đồ họa (GUI - Graphics User Interface). Với C++, các bạn có thể sử dụng MFC (Microsoft) hay QT (hỗ trợ nhiều nền tảng).

- Kết quả của công việc này là bản thiết kế các màn hình và minh họa (có thể chụp lại màn hình của chương trình và cắt phù hợp).

3.2.3 Tổ chức và quản lý mã nguồn trong quá trình phát triển

Tùy vào bản thiết kế, các nhóm có cách tổ chức mã nguồn khác nhau. Các nhóm cần phát họa và lập tài liệu về việc tổ chức này.

Các nhóm được khuyến khích sử dụng "[Github](#)" hoặc một dịch vụ quản lý phiên bản tương tự để quản lý và tích hợp code cho các thành viên. Nhóm nào sử dụng tính năng này sẽ có điểm cộng thêm.

3.2.4 Thu thập số liệu

Các nhóm phải tự nhập hoặc thu thập đủ số liệu để đánh giá các tính năng của hệ thống. Khi sử dụng dữ liệu từ nguồn nào thì cần trích dẫn cụ thể.

3.2.5 Kiểm tra phần mềm

Trong quá trình lập trình, các sinh viên có thể thường xuyên sử dụng tính năng "debug" đi kèm bộ biên dịch để xác định lỗi hoặc xem các kết quả trung gian khi chương trình chạy.

Khi đã có sản phẩm, các nhóm cũng phải chạy với những dữ liệu khác nhau được thiết kế cẩn thận (+ cách tương tác khác nhau) để kiểm tra tính đúng đắn cho sản phẩm của mình. Danh sách các trường hợp kiểm tra này và kết quả kiểm tra cũng phải thể hiện trong báo cáo.

Nếu nhóm nào có thành viên **đặc biệt tốt**, các nhóm đó nên sử dụng thêm thư viện về **UnitTest** trong quá trình phát triển.

3.2.6 Lập tài liệu hướng dẫn phát triển, triển khai và sử dụng

Các nhóm cần tiến hành các công việc theo yêu cầu:

1. **Định dạng mã nguồn:** Mã nguồn cần định dạng để tuân thủ những hướng dẫn về **phong cách lập trình** của giảng viên trong quá trình giảng dạy. Mã nguồn chương trình của các nhóm cũng phải có tính tự giải thích cao. Nghĩa là, phải có chú thích cho các hàm, các biến, hằng, và các bước quan trọng trong giải thuật.
2. **Lập tài liệu phát triển:** Tài liệu này cần thể hiện rõ các điểm sau:
 - Cấu trúc và sự tổ chức của (các) dự án, các khối, các thư mục trong (các) dự án, cùng ý nghĩa của chúng.
 - Đối với các thư viện bên ngoài (download được, ví dụ như nhóm có dùng **GUI** hay **Unitest**), cần giải thích cách cài đặt các thư viện đó.
 - Các bước để biên dịch chương trình thành công. Đối với các nhóm phát triển trên **Linux**, hãy dùng tạo **Makefile**.
 - Vẽ **flowchart** và **viết mã giả** ⁸ cho các giải thuật quan trọng trong hệ thống **LIBPRO**.

⁸Đây là điểm **BẮT BUỘC PHẢI CÓ**, nếu nhóm nào không có thì xem như không nhận điểm cho toàn bộ các phần trong báo cáo



3. **Lập tài liệu hướng dẫn triển khai phần mềm:** Tài liệu này hướng dẫn việc cài đặt và cấu hình để chạy thành công phần mềm.
4. **Lập tài liệu hướng dẫn sử dụng phần mềm:** Tài liệu này hướng dẫn sử dụng phần mềm cho các vai trò mà phần mềm hỗ trợ.

4 Báo cáo kết quả

Các nhóm thực hiện báo cáo theo mẫu đính kèm. Báo cáo của các nhóm được thực hiện với **Latex**. Các nhóm phải nộp cả bản mềm của bản báo cáo lên trang môn học và in ra cuốn để nộp trong ngày báo cáo.

5 Ngày báo cáo

Các nhóm sẽ báo cáo tập trung vào một buổi. Ngày báo cáo cụ thể sẽ được cập nhật trong phiên bản kế tiếp (dự kiến sẽ xảy ra trong tuần dự trữ - liền trước tuần thi học kỳ).

6 Tiêu chí chấm bài

Bài làm của các nhóm được chấm theo các tiêu chí nêu trong Bảng 1. Theo đó, bài làm của các nhóm được chấm ở 3 loại kết quả khác nhau (Cuốn báo cáo, Mã nguồn chương trình, và Chương trình **LIBPRO**) - tùy thuộc vào từng tiêu chí cụ thể. Tổng các hệ số trong phần bắt buộc là 1; do đó, để thành điểm các nhóm tự phải nhân 10. Ví dụ, Tiêu chí "Lập tài liệu" được chấm trong cả cuốn báo cáo và mã nguồn; tìm đọc Phần 3.2.6 để biết trong cuốn báo cáo và mã nguồn các nhóm phải làm gì để được điểm tối đa.



Bảng 1: Các tiêu chí chấm bài

Phần Bắt buộc				
STT	Tiêu chí	Báo cáo	Mã nguồn	C.Trình
1.	Phân tích	0.1		
2.	Thiết kế	0.1		
3.	Tổ chức mã nguồn		0.05	
4.	Thu thập số liệu			
5.	Kiểm tra phần mềm	0.025		
6.	Lập tài liệu	0.025	0.05	
7.	Phân chia công việc và phối hợp	0.025		
8.	Báo cáo hoàn chỉnh và chuyên nghiệp			
9.	Sử dụng thành thạo nhập xuất màn hình			0.1
10.	Sử dụng thành thạo nhập xuất tập tin			0.1
11.	Phát triển giải thuật phù hợp (∈ Báo cáo & chương trình)	0.025		0.4
Tổng phần bắt buộc:		0.3	0.1	0.6
Phần cộng thêm				
1.	Có sử dụng GUI			0.1
2.	Có sử dụng Git hay phần mềm tương đương		0.05	
3.	Có sử dụng công cụ để thiết kế	0.1		
4.	Viết được Makefile		0.05	
5.	Có nhiều tính năng hay bổ sung			0.0 → 0.15
Tổng phần cộng thêm:		0.1	0.1	0.1 → 0.25
			Tổng toàn bộ	1.3 → 1.45