

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



Bộ môn Kỹ thuật lập trình

**Báo cáo Bài tập lớn số 2: Chương trình quản lý thư viện
LIBPRO**

GVHD: TS. Lê Thành Sách
Sinh viên: Nguyễn Anh Khoa - 1611617
Nguyễn Minh Khôi - 1611657
Phạm Quốc Nam - 1612128
VSTS: https://thaoxkhoa.visualstudio.com/_git/LIBPRO
(VSTS là một dịch vụ private, chỉ những thành viên mới được phép truy cập)
Github: <https://github.com/nganhkhoa/LIBPRO>
(github là được chuyển lại từ VSTS, nên không có lịch sử đầy đủ)

Chương 1

Tổng quan về LIBPRO

LIBPRO. được sự gợi ý và thúc đẩy của các Giảng viên Môn Kỹ Thuật Lập Trình, là một phần mềm chuyên biệt quản trị các hoạt động trong các thư viện thông thường. LIBPRO tuy chưa phải phần mềm hoàn hảo nhất về các tiện ích và giao diện người dùng, nhưng đã đáp ứng được các yêu cầu cơ bản cho những người sử dụng. LIBPRO là sản phẩm của SIMple Group, gồm các thành viên Nguyễn Anh Khoa, Nguyễn Minh Khôi và Phạm Quốc Nam cùng phát triển.

Chương 2

Phần mềm Quản lý Thư viện LIBPRO

2.1 Đôi nét cơ bản về LIBPRO

LIBPRO là chương trình quản lý thư viện được xây dựng trên nền tảng ngôn ngữ C++ kết hợp với GUI Qt..., lưu dữ liệu người dùng, sách và yêu cầu mượn ở dạng file .json. LIBPRO được phát triển nhằm mục đích nghiên cứu và rèn luyện, nhiều tính năng mở rộng vẫn chưa được hoàn thiện, chỉ mới ở mặt ý tưởng.

2.2 Hướng dẫn cài đặt phần mềm

2.3 Hướng dẫn sử dụng phần mềm

2.3.1 Đăng nhập

2.3.2 Đăng ký

Khi chưa có hồ sơ người dùng LIBPRO, chọn **Profile->Log In** và chọn vào ô chưa đăng ký để đăng ký.

Các thông tin yêu cầu bắt buộc khi đăng ký là họ và tên người đăng ký, tên người dùng trong LIBPRO và số CMND (hoặc số thẻ căn cước).

Sau khi đăng ký, người dùng sẽ được tạo một mật khẩu mới bất kỳ để đăng nhập vào LIBPRO, sau đó người dùng có thể chỉnh sửa lại mật khẩu.

2.3.3 Đăng nhập

Khi đã có hồ sơ người dùng LIBPRO, chọn **Profile->Log In** và nhập tên người dùng LIBPRO và mật khẩu để đăng nhập.

Nếu quên mật khẩu, hãy nhấn vào ô tạo mật khẩu mới, và lúc này, hồ sơ người dùng sẽ được khóa lại, chờ khi người dùng nhận được mật khẩu mới từ QUẢN LÝ NGƯỜI DÙNG.

2.3.4 Đăng xuất

Nếu đã hoàn thành công việc trên hồ sơ của mình, người dùng có thể đăng xuất bằng cách chọn trên thanh menu **Profile->Log Out**.

2.3.5 Tùy chỉnh khác

Tạo mới tài khoản : các chức năng trong thư viện được cấp phát thông qua tài khoản của người dùng, vì vậy cần thiết tạo tài khoản cho người dùng bằng cách: trên thanh menu, chọn **Profile->New Account**.

Để chuyển đổi giữa các tài khoản, chọn trên thanh menu **Profile->Choose Account**.

Để chỉnh sửa các thông tin người dùng, bao gồm cả tên tài khoản, và mật khẩu người dùng, chọn **Profile->Edit->...** tùy theo nhu cầu của người dùng.

2.3.6 Xem thông báo

Có hai mục để xem là xem theo lịch sử và xem theo các thông báo chưa xem bao giờ.

2.3.7 Tùy chỉnh

Chủ đề của LIBPRO

(tính năng mở rộng chưa được cập nhật)

Ngôn ngữ của LIBPRO

(tính năng mở rộng chưa được cập nhật)

2.3.8 Duyệt sách

Trên thanh menu, chọn **Library->View All Book** để xem toàn bộ sách trong thư viện.

Nếu muốn tìm sách, chọn **Library->View All Book** và nhập tên sách người dùng muốn tìm vào thanh Search và chọn **Find** để tìm sách.

Sách yêu thích

Nếu muốn thêm sách vào mục yêu thích, chọn **Library->View All Book** và chọn **Add to Favorites** tại quyển sách mà người dùng thích.

Để xem các sách yêu thích đã chọn, trên thanh menu, chọn **Library->Favorites**.

Nếu không thích quyển sách nào trong mục yêu thích, người dùng nhấp vào nút **Out Favorites** để xóa sách đó khỏi mục yêu thích.

Mượn sách

Tại màn hình duyệt sách hoặc màn hình mục yêu thích hoặc màn hình tìm sách, nhấp vào nút **Add to Cart** để thêm sách vào giỏ.

Nếu người dùng đã có tài khoản với chức năng READER, trên thanh menu, chọn **Role->READER->Cart** để gửi yêu cầu mượn sách đến thủ thư và chờ sự xác nhận của thủ thư.

2.3.9 Chức năng

Chọn trên thanh menu, **Role**, và lựa chọn các chức năng bên dưới. Lưu ý là chỉ những chức năng của tài khoản hiện thời mới khả dụng.

READER - ĐỘC GIẢ

1. xem giỏ hàng: vào **READER->Cart** để xem giỏ hàng của người dùng, để đăng ký mượn sách.
2. Photocopy: có thể photo các sách cho phép photo (tính năng mở rộng chưa được cập nhật)

REVIEWER - NHÀ PHÊ BÌNH

Có thể thêm review vào sách, (tính năng mở rộng chưa được cập nhật)

CD - XEM SÁCH NÓI

Có thể duyệt vào các sách nói, (tính năng mở rộng chưa được cập nhật)

MOVIE - XEM PHIM TƯ LIỆU

Có thể duyệt các phim tư liệu, các video học thuật, (tính năng mở rộng chưa được cập nhật)

EBOOK - NHÀ PHÊ BÌNH

Có thể duyệt các sách điện tử, (tính năng mở rộng chưa được cập nhật)

VIP

Có thể duyệt tất cả các loại sách, tư liệu và phê bình, (tính năng mở rộng chưa được cập nhật)

LIBRARIAN - THỦ THƯ

1. Duyệt yêu cầu mượn sách: vào **Role->LIBRARIAN->Browse Borrow** để duyệt các yêu cầu mượn sách từ người dùng.
2. Quản lý sách: vào **Role->LIBRARIAN->Book Management** để thêm sách, xóa sách và thay đổi các thông tin của sách.

USER ADMINISTRATOR - QUẢN LÝ NGƯỜI DÙNG

1. Quản lý hồ sơ người dùng: vào **Role->USER ADMINISTRATOR->Profile Management** để thêm, xóa người dùng và tạo mật khẩu mới cho người dùng quên mật khẩu.
2. Thêm vào danh sách đặc biệt : (tính năng mở rộng chưa được cập nhật).
3. Khóa hồ sơ người dùng: **Role->USER ADMINISTRATOR->Lock** để khóa các hồ sơ vi phạm.

ACCOUNTANT - QUẢN LÝ TÀI CHÍNH

Quản lý chi tiêu của thư viện và của người dùng. (tính năng mở rộng chưa được cập nhật)

2.3.10 Trợ giúp

1. Đọc hướng dẫn sử dụng thư viện.
2. Đọc hướng dẫn sử dụng LIBPRO.
3. Về người phát triển phần mềm.
4. Liên hệ.

Chương 3

Tiến trình thực hiện phần mềm

3.1 Phân tích hoạt động của thư viện

3.1.1 Vai trò người sử dụng

LIBPRO cung cấp 4 vai trò chủ yếu cho người sử dụng:

1. KHÁCH
2. ĐỘC GIẢ
3. THỦ THƯ
4. QUẢN LÝ NGƯỜI DÙNG

KHÁCH

KHÁCH được định nghĩa là người được truy cập vào phần mềm LIBPRO, tuy nhiên, KHÁCH chỉ có thể xem phần trợ giúp **Help** trên thanh Menu. Số lượng KHÁCH truy cập vào phần mềm là không giới hạn

ĐỘC GIẢ

ĐỘC GIẢ được định nghĩa là người được truy cập vào phần mềm LIBPRO, có thể xem sách, xem thông báo, gửi yêu cầu mượn sách đến thủ thư, điều chỉnh các thông tin cá nhân, giao diện LIBPRO. Số lượng ĐỘC GIẢ truy cập vào phần mềm là có giới hạn, tùy theo quy định của thư viện.

THỦ THƯ

THỦ THƯ được định nghĩa là người được truy cập vào phần mềm LIBPRO, có khả năng quản lý sách, quản lý việc mượn và trả sách, gửi thông báo về sách mới, sách được mượn đến người dùng. Số lượng THỦ THƯ truy cập vào phần mềm là rất ít, do QUẢN LÝ THƯ VIỆN cấp phát.

QUẢN LÝ NGƯỜI DÙNG

QUẢN LÝ NGƯỜI DÙNG được định nghĩa là người được truy cập vào phần mềm LIBPRO, có khả năng quản lý hồ sơ người dùng, thêm, khóa hoặc xóa người dùng, gửi thông báo đến người dùng các thông tin liên quan đến hồ sơ người dùng. Số lượng QUẢN LÝ NGƯỜI DÙNG truy cập vào phần mềm rất ít, do QUẢN LÝ THƯ VIỆN cấp phát.

Ngoài ra, LIBPRO còn cung cấp thêm một số vai trò khác

1. NHÀ SƯU TẦM: có thể mua sách
2. NHÀ PHÊ BÌNH: có thể bình luận và đánh giá sách
3. SÁCH NÓI: truy cập vào kho sách nói
4. TƯ LIỆU PHIM ẢNH: truy cập vào kho phim tư liệu
5. SÁCH ĐIỆN TỬ: truy cập vào kho sách điện tử
6. VIP: có thể xem tất cả các loại sách, tư liệu
7. QUẢN LÝ TÀI CHÍNH: quản lý các chi tiêu của thư viện và người dùng

3.1.2 Cấp độ quản lý và thực thi

Cấp Người dùng

Đây là cấp lớn nhất và khái quát nhất trong môi trường LIBPRO.

Một số đặc điểm của cấp Người dùng:

1. Cấp người dùng sẽ được cấp phát cho người dùng sau khi QUẢN LÝ NGƯỜI DÙNG chấp thuận yêu cầu đăng ký của người dùng.
2. Tên gọi cấp Người dùng là duy nhất và không trùng lặp, xét cả trường hợp chữ in và chữ thường
3. Tên gọi cấp Người dùng có độ dài không quá 32 ký tự và không chứa các ký tự đặc biệt (về các ký tự đặc biệt, xem kỹ hơn ở mục...)
4. Một tên Người dùng có thể đăng ký nhiều tài khoản phân biệt và không trùng lặp
5. Cấp người dùng chỉ được xóa khi người dùng gửi yêu cầu đến QUẢN LÝ NGƯỜI DÙNG hoặc do người dùng vi phạm các quy định của thư viện ở mức nào đó trở lên.

Cấp Tài khoản

Đây là cấp thứ hai sau cấp Người dùng trong môi trường LIBPRO.

Một số đặc điểm của cấp Tài khoản:

1. Cấp Tài khoản sẽ có thể cấp phát cho người dùng sau khi QUẢN LÝ NGƯỜI DÙNG chấp thuận yêu cầu đăng ký của người dùng.
2. Tên gọi cấp Tài khoản là duy nhất và không trùng lặp, xét cả trường hợp chữ in và chữ thường
3. Tên gọi cấp Tài khoản có độ dài không quá 32 ký tự và không chứa các ký tự đặc biệt (về các ký tự đặc biệt, xem kỹ hơn ở mục...)
4. Một tên Tài khoản có thể đăng ký nhiều vai trò phân biệt và không trùng lặp
5. Cấp Tài khoản sẽ bị khóa nếu tài khoản đó được liệt vào danh sách đen. (Chi tiết về danh sách đen, xem mục...)
6. Cấp tài khoản chỉ được xóa khi người dùng gửi yêu cầu đến QUẢN LÝ NGƯỜI DÙNG hoặc do người dùng vi phạm các quy định của thư viện ở mức nào đó trở lên.

Cấp vai trò

Như đã giới thiệu ở phần 2.1.1 về các vai trò người dùng, phần này xin bổ sung thêm một số quy định

1. Số vai trò trong 1 tài khoản tối đa có thể là 3
2. Mỗi vai trò đặc trưng cho khả năng khai thác thư viện của người dùng với từng mức phí khác nhau
3. Trong cùng một tài khoản, không thể cùng lúc thao tác ở 2 vai trò khác nhau. BUỘC phải chuyển từ vai trò A sang vai trò B để thực hiện công việc của B chứ không thể thực hiện công việc của B khi đang ở vai trò A.

3.1.3 Thủ tục đăng ký và đăng nhập

Việc truy cập vào LIBPRO hoàn toàn chưa cần đến việc đăng ký, như đã nêu ở trên, nhưng để nâng cao mức thao tác, người sử dụng cần đăng ký người dùng và tài khoản LIBPRO.

Về việc Đăng ký

Người dùng sẽ nhấp vào ô đăng ký để thực hiện yêu cầu đăng ký của mình. Thông tin cần điền vào sẽ có những mục bắt buộc và không bắt buộc. Sau khi điền xong, người dùng sẽ nhấn nút gửi và chờ duyệt từ QUẢN LÝ NGƯỜI DÙNG. Vì thành viên của thư viện chỉ có giới hạn, nên không phải mọi người đều đăng ký thành công.

Việc đăng ký thường chỉ xảy ra khi muốn tạo thêm tài khoản mới cho cấp Người dùng hiện tại. Nếu muốn đăng ký cấp Người dùng mới, phải đăng xuất khỏi cấp người dùng hiện tại để đăng ký.

Về thông tin cần phải đăng ký:

1. Tên User (Là tên để khai báo trong LIBPRO)
2. Họ và Tên người dùng (Là tên trong khai sinh)
3. Ngày tháng năm sinh
4. Giới tính
5. Quốc tịch
6. Dân tộc
7. Số chứng minh nhân dân
8. CMND cấp ngày
9. Địa chỉ hiện tại
10. Số điện thoại
11. Công việc
12. Cơ quan làm việc
13. Địa chỉ cơ quan

Về việc Đăng nhập

Người dùng sẽ nhập vào ô đăng nhập để đăng nhập vào hồ sơ người dùng của mình. Khi đăng nhập, chỉ cần điền tên người dùng và password của người dùng đó. Sau khi đăng nhập, mọi tùy chỉnh trước lần đăng xuất cuối cùng đều sẽ được giữ nguyên.

Khi người dùng muốn thoát chương trình, cần thực hiện bước đăng xuất trước khi thoát.

Nếu người dùng không đăng xuất trước khi thoát khỏi chương trình, ở lần mở chương trình tiếp theo, chương trình sẽ hiện lên tên tài khoản chưa đăng xuất và yêu cầu nhập password để truy cập vào tài khoản; hoặc truy cập như là KHÁCH.

Về các ký tự đặc biệt

Phần này chủ yếu dành cho việc đặt tên người dùng hoặc tên tài khoản.

Các ký hiệu đặc biệt là: !@#\$%^&* () <>? và dấu cách

3.1.4 Quản lý sách

Các đặc tính của sách

Sách trong LIBPRO được quản lý dựa trên các thông tin sau:

1. Mã ISBN
2. Tên Đầu sách
3. Tên tác giả (một hoặc nhiều tác giả)
4. Nhà xuất bản
5. Thể loại (một hoặc nhiều thể loại)
6. Năm xuất bản
7. Nội dung chính

Sách trong thư viện sẽ được quản lý trực tiếp qua mã sách (ISBN), mỗi cuốn sách có một mã sách riêng biệt và không trùng lặp.

Thể loại: Các thể loại sách được phân ra như sau:

- Hội họa và Nhiếp ảnh
- Sách Audio
- Tự truyện và Hồi ký
- Sách trên CD
- Tài chính và Tiền tệ
- Lịch
- Trẻ em
- Kinh Thánh và Sách Kitô giáo
- Truyện tranh
- Công nghệ và Máy tính

- Ẩm thực
- Sở thích và...
- Deals in Books
- Giáo dục và Đào tạo
- Cơ khí và Vận tải
- Đồng tính
- Sức khỏe
- Lịch Sử
- Hải hước
- Luật
- Văn học
- Y học
- Phiêu lưu mạo hiểm
- Làm cha mẹ
- Chính trị và Khoa học xã hội
- Tham khảo
- Tôn giáo và Tín ngưỡng
- Lãng mạn
- Khoa học và Toán học
- Khoa học viễn tưởng
- Kỹ năng sống
- Thể thao và Hoạt động Ngoài trời
- Lứa tuổi thiếu niên
- Ôn luyện các kỳ thi
- Sách giáo khoa
- Du lịch và Khám phá

Sắp xếp sách

Sách được sắp xếp theo các tiêu chí giảm dần tính ưu tiên sau:

- Theo Thời điểm xuất bản (ưu tiên sách xuất bản gần đây)
- Theo Bảng chữ cái (A-Z hoặc Z-A)
- Theo Tên tác giả
- Theo Thể loại sách
- Theo Nhà xuất bản

Nhập sách

Việc nhập sách được diễn ra định kỳ hoặc không định kỳ. Định kỳ 3 tháng hoặc 6 tháng sẽ có đợt sách mới về; sách mới có thể là:

- Giống sách đã có trong thư viện, việc nhập sách mới làm tăng số lượng sách chỉ của đầu sách đó
- Là sách hoàn toàn mới chưa có trong thư viện, việc nhập sách mới làm tăng số lượng các đầu sách

Việc nhập sách mới được quản lý và thông qua bởi quyết định của thủ thư.

Sau khi nhập sách mới về, thủ thư cần chủ động gửi thông báo đến các độc giả còn quyền người dùng.

Mượn sách

1. Ai có thể mượn sách

Chỉ cấp vai trò ĐỘC GIẢ có quyền gửi yêu cầu mượn sách.

Việc xác nhận yêu cầu mượn sách từ ĐỘC GIẢ do THỦ THƯ xử lý.

2. Có thể mượn sách trong bao lâu

Theo quy định hiện hành và phổ thông, mỗi đầu sách chỉ được mượn nhiều nhất là 2 tuần, kể từ ngày ghi trên yêu cầu mượn sách.

Nếu có nhu cầu gia hạn, cấp vai trò THỦ THƯ sẽ quyết định nếu có thể gia hạn, và chỉ được gia hạn thêm 1 tuần là tối đa.

3. Một lần mượn được bao nhiêu quyển sách

Một lần mượn (tại thời điểm bất kỳ của thời gian mượn) thì một ĐỘC GIẢ chỉ có thể mượn tối đa 4 đầu sách.

Tránh việc mượn hơn 1 quyển của cùng 1 đầu sách.

Trả sách

1. Khi nào phải trả sách

Việc trả sách được thực hiện thông qua hai quy định sau:

- (a) Bất kỳ quyển sách nào được mượn đều phải trả lại thư viện trong vòng 2 tuần kể từ ngày ghi trên yêu cầu mượn sách, hoặc trong vòng 3 tuần nếu được gia hạn.
- (b) Thư viện có yêu cầu trả sách trước thời hạn vì một số lý do cụ thể.

Trường hợp ĐỘC GIẢ không trả sách cho thư viện theo hai quy định trên, sẽ có mức xử phạt tương ứng.

2. Tình trạng sách

Tình trạng sách là một đại lượng có tính tương đối tùy thuộc vào quy định thư viện.

Nếu sách bị hư hại thì thủ thư có quyền phạt tiền người dùng, bằng cách thông báo và trừ tiền trong tài khoản của người dùng.

Nếu làm mất sách, thì phải xử phạt theo quy định thư viện.

3.1.5 Quản lý người dùng

Việc quản lý người dùng được thực hiện thông qua vai trò QUẢN LÝ NGƯỜI DÙNG.

Tạo cấp Người dùng

Khi nhận được yêu cầu đăng ký hồ sơ người dùng LIBPRO, QUẢN LÝ NGƯỜI DÙNG sẽ quyết định chấp thuận yêu cầu đăng ký hay không.

Cập nhật tình trạng người dùng

Dựa theo lịch sử đăng nhập và đăng xuất của người dùng để đánh giá và xếp loại người dùng theo các tiêu chí (tính năng mở rộng chưa được cập nhật)

Danh sách Đỏ

Các tài khoản vi phạm quy định thư viện từ mức 3 đến mức 1 sẽ được liệt kê vào một danh sách gọi là danh sách Đỏ.

Khóa tài khoản và Danh sách Đen

Việc khóa tài khoản được thực hiện theo yêu cầu từ THỦ THƯ khi tài khoản này vi phạm các quy định từ mức 4 trở xuống.

Đồng thời, khi bị khóa lần đầu, các tài khoản đó sẽ được liệt kê trong một danh sách, gọi là Danh sách Đen.

Sau một khoảng thời gian (tùy vào thái độ của người dùng tài khoản), tài khoản từng bị khóa sẽ được mở, nhưng tên tài khoản vẫn nằm trong danh sách Đen.

Các tài khoản bị liệt vào danh sách Đen sẽ chịu một số hạn chế khi sử dụng thư viện.

Xóa tài khoản và Xóa người dùng

Các tài khoản vi phạm quy định thư viện từ mức 6 trở xuống sẽ bị xóa tài khoản cùng mọi thông tin về tài khoản đó, trừ danh sách Đỏ và danh sách Đen

Các người dùng vi phạm quy định thư viện từ mức 8 trở xuống sẽ bị xóa tài khoản cùng mọi thông tin về tài khoản đó, trừ danh sách Đỏ và danh sách Đen.

3.2 Vẽ sơ đồ thiết kế dữ liệu

3.2.1 Sơ đồ quản lý hồ sơ người dùng

Hồ sơ người dùng được chia ra 4 tập tin:

1. Tập tin chính quản lý người dùng

```
{
  "UserList": [
    {
      "AccountList": [
        {
          "AccountName": string,
          "AccountRoleMap": [
            numbers
          ],
          "Lock": bool
        }
      ],
      "Balance": number double,
      "Borrowed": [
        {
          "Submit ID": number,
          "ISBN": string
        }
      ],
      "Cart": [
        strings
      ],
      "DateOfBirth": string,
      "Ethnic": string,
      "First": bool,
      "Gender": string,
      "IDNumber": string,
      "IDValidDate": string,
      "Institute": string,
      "InstituteAddress": string,
      "Nationality": string,
      "Occupation": string,
      "Password": string,
      "PhoneNumber": string,
      "UserAddress": string,
      "UserFirstName": string,
      "UserID": string,
      "UserLastName": string,
      "Username": string
    }
  ]
}
```

2. Tập tin quản lý việc đăng ký

```
{
  "SignUp": [
    {
      "AccountList": [
        {
          "AccountName": string,
          "AccountRoleMap": [
            numbers
          ],
          "Lock": true
        }
      ],
      "Balance": number double,
      "DateOfBirth": string,
      "Ethnic": string,
      "First": true,
      "Gender": string,
      "IDNumber": string,
      "IDValidDate": string,
      "Institute": string,
      "InstituteAddress": string,
      "Nationality": string,
      "Occupation": string,
      "Password": string,
      "PhoneNumber": string,
      "UserAddress": string,
      "UserFirstName": string,
      "UserID": string,
      "UserLastName": string,
      "Username": string
    }
  ]
}
```

3. Tập tin quản lý việc đăng ký không thành công

```
{
  "Rejected": [
    {
      "AccountList": [
        {
          "AccountName": string,
          "AccountRoleMap": [
            numbers
          ],
          "Lock": true
        }
      ],
      "Balance": number double,
      "DateOfBirth": string,
      "Ethnic": string,
      "First": true,
      "Gender": string,
      "IDNumber": string,
      "IDValidDate": string,
      "Institute": string,
      "InstituteAddress": string,
      "Nationality": string,
      "Occupation": string,
      "Password": string,
      "PhoneNumber": string,
      "UserAddress": string,
      "UserFirstName": string,
      "UserID": string,
      "UserLastName": string,
      "Username": string
    }
  ]
}
```


4. Tập tin quản lý việc xóa người dùng

```
{
  "Deleted": [
    {
      "AccountList": [
        {
          "AccountName": string,
          "AccountRoleMap": [
            numbers
          ],
          "Lock": true
        }
      ],
      "Balance": number double,
      "DateOfBirth": string,
      "Ethnic": string,
      "First": true,
      "Gender": string,
      "IDNumber": string,
      "IDValidDate": string,
      "Institute": string,
      "InstituteAddress": string,
      "Nationality": string,
      "Occupation": string,
      "Password": string,
      "PhoneNumber": string,
      "UserAddress": string,
      "UserFirstName": string,
      "UserID": string,
      "UserLastName": string,
      "Username": string
    }
  ]
}
```

3.2.2 Sơ đồ quản lý sách

Các sách trong thư viện sẽ được lưu trữ trong cùng một tập tin có sơ đồ như sau:

```
{
  "BookLibrary": [
    {
      "Author": [
        strings
      ],
      "BookNumber": number,
      "Borrowed": number,
      "Category": [
        strings
      ],
      "Content": string,
      "ISBN": string,
      "Publisher": string,
      "Title": string,
      "Year": string
    }
  ]
}
```

3.2.3 Sơ đồ quản lý việc mượn sách

Việc lưu trữ các hoạt động mượn trả sách được lưu thành 2 tập tin, có sơ đồ như sau:

1. Tập tin quản lý duyệt mượn sách

```
{
  "Checked": [
    {
      "ISBN": string,
      "Request Date": string,
      "Submit ID": number,
      "User": string
    }
  ],
  "Submission": [
    {
      "ISBN": string,
      "Request Date": string,
      "Submit ID": number,
      "User": string
    }
  ]
}
```

2. Tập tin quản lý mượn trả sách

```
{
  "Accepted": {
    "Pending": [
      {
        "Browsed Date": string,
        "Permission Granted by": string,
        "Submit ID": number
      }
    ],
    "Received": [
      {
        "Browsed Date": string,
        "Permission Granted by": string,
        "Received Date": string,
        "Submit ID": number
      }
    ],
    "Returned": [
      {
        "Browsed Date": string,
        "Permission Granted by": string,
        "Received Date": string,
        "Returned Date": string,
        "Submit ID": number
      }
    ]
  }
  "Rejected": [
    {
      "Browsed Date": string,
      "Permission Granted by": string,
      "Submit ID": number
    }
  ]
}
```

3.3 Quá trình làm việc

3.3.1 Tạo thư mục làm việc trên visualstudio.com

Visual Studio gần đây vừa triển khai một ứng dụng mới, và Visual Studio Team Service, giúp các lập trình viên làm việc nhóm chung với nhau thông qua mô hình Agile. Để bắt đầu, bạn cần có một tài khoản của Microsoft, sau đó truy cập theo link sau <https://www.visualstudio.com/team-services/> và chọn "Get started for free" để tạo một tên miền của mình. Tên miền sẽ có dạng <https://tênmiền.visualstudio.com>.

Một tên miền có thể có nhiều dự án (project) và một dự án thì có thể có nhiều nguồn (repository hay repo). Mặc định chúng ta sẽ có một dự án là My Project. Bỏ qua nó và vào trang chủ chọn New Project. Lúc này bạn sẽ có các lựa chọn về tên dự án, chỉ dẫn cho dự án và mô hình làm việc và các phương thức quản lý phiên bản (version control). Về mô hình làm việc, nhóm không biết nhiều về các mô hình này, nên để mặc định là Agile. Về quản lý phiên bản, nhóm thống nhất dùng Git.

Sau khi đã tạo xong, một màn hình yêu cầu upload code lên theo git hiện ra. Sau khi upload code lên, thì chúng ta xem như đã hiện thực xong bước đầu của việc tạo dự án trên Visual Studio Team Service.

3.3.2 Sử dụng Git để quản lý và cập nhật tài liệu

Sơ lược về Git

Tổng quan sơ về Git. Git được hình thành từ khá lâu, bắt nguồn từ những vấn đề với những phần mềm quản lý phiên bản trước đó khi hướng theo code mà không hướng theo tài liệu. Lâu dần, Git được phát triển để từ nay thành một trong những phần mềm quản lý phiên bản được nhiều người sử dụng nhất.

Để bắt đầu với Git, bạn cần lên trang <https://git-scm.com/downloads> và tải về Git. Sau khi cài đặt, Git đã ở trên PATH trong máy, bạn có thể hiện thực các câu lệnh trên Git ở trên chính Shell, hoặc sử dụng Git Bash hoặc Git GUI, là những công cụ hỗ trợ riêng của Git. Ở đây nhóm không sử dụng Git GUI, nên thao tác hiện thực là trên Shell.

Git cơ bản

Cùng điểm qua các câu lệnh cơ bản của Git.

Để thực hiện sao lưu thay đổi trong một thư mục (kích hoạt git) thì sẽ sử dụng câu lệnh:

```
git init
```

Lúc này thư mục sẽ được git quan sát các thay đổi. Để lấy về một thư mục đã có, ta có hai cách:

- 1: `git remote add <tên ngắn> <link>`
- 2: `git clone <link>`

Với clone, tên ngắn được mặc định là origin. Trước khi upload, chúng ta phải kiểm tra cho máy cập nhật mới nhất với repo.

```
git fetch
```

Để lấy về cập nhật và sau đó là

```
git merge
```

Để ghép các thay đổi vào, xung đột sẽ được đề cập sau. Hoặc dùng câu lệnh

```
git pull
```

Là một hiện thực của hai câu lệnh trên. Sau khi đã có code, ta thực hiện thay đổi, git sẽ tự nhận biết thay đổi. Để biết chúng ta đã thay đổi gì,

```
git status
```

sẽ hiện ra các file chưa được stage, hoặc đã stage nhưng chưa commit. Mọi quá trình trên Git đều diễn ra qua 3 giai đoạn,

```
Stage -> Commit -> (Fetch -> Merge -> Xung đột -> Xử lý) ->Push
```

Stage là để git biết rằng nó sẽ lưu file vào. Commit là để git biết rằng chúng ta muốn lưu lại lúc này, và Push là upload lên repo. Khái niệm dùng Git cơ bản khép lại ở đây, và phần nâng cao ngay phía sau.

Git nâng cao

Khi gặp xung đột lúc merge. Git sẽ yêu cầu xử lý xung đột, và cho chúng ta biết file và chỗ nào trong file. Lúc đó, trên bash sẽ hiện ra tên các file đã xảy ra xung đột, (xung đột thường là khi nhiều người sửa cùng file ở cùng một chỗ, vì cách lưu của Git là theo các bit nên nếu chỗ bit đó bị thay đổi bởi nhiều người thì sẽ nhận biết, ở chỗ khác sẽ không bị xung đột). Chúng ta với file đó sẽ hiện ra các dòng thế này:

```
<<<<<<< HEAD
code của ta
=====
code trên repo
>>>>>>> master
```

Có thể không phải là master mà là một chuỗi số, chuỗi số tượng trưng cho lần commit nào đó, hoặc là một nhánh nào đó. Tìm chuỗi đó sẽ cho chúng ta biết được tên của người đã commit lần đó. Từ đó liên hệ với nhau và sửa chỗ đó. Khi sửa xong, chúng ta vẫn push như bình thường.

Tiếp theo là khái niệm cần thiết trong việc làm việc Git. Nhánh. Nhánh (Branch) tương tự như việc khi chúng ta copy code ra USB rồi đưa cho một người khác sửa rồi sau đó ghép lại. Điều tốt hơn việc copy ra USB là khi Git tự động biết được những file nào mà chúng ta đã thay đổi. Đồng thời cho chúng ta xem lại code trước khi ghép lại.

Để bắt đầu tạo nhánh mới, có thể hiện thực trên repo. Hoặc trên Bash:

```
git checkout -b <tên nhánh mới>
```

Nhánh mới sẽ là một copy của nhánh chúng ta đang ở.

```
git branch -a
```

Là câu lệnh để biết được tất cả các nhánh, (trước khi hiện thực câu lệnh này, sẽ là một điều không ngoan khi lấy dữ liệu từ repo về trước, git fetch). Với dấu sao và màu chữ màu xanh chính là nhánh chúng ta đang ở. Một lưu ý, tạo nhánh trên Bash vẫn chưa được thấy trên repo cho tới khi ta push.

Khi làm việc với nhánh đã xong, chúng ta sẽ bắt đầu ghép vào nhánh gốc hoặc nhánh cha, để thực hiện, chúng ta cần lên repo và tìm pull request. Lúc này sẽ có màn hình về các thông tin đồng thời một danh sách các thanh đối trong các file để chúng ta xem xét.

Nếu làm việc đơn lẻ, chính chúng ta có thể đồng ý cho duyệt, nhưng khi làm việc nhóm, cần có một Code reviewer, thường là admin, sẽ xem xét code và đưa ra các comment các câu hỏi nếu có thắc mắc trước khi bấm chấp nhận hoặc từ bỏ lần ghép code này.

Git nâng cao còn có nhiều điều nữa nhưng nhóm vẫn chưa thể dùng hết được. Một phần không thuộc Git nhưng lại là một tính tăng quan trọng trong việc lập trình theo một nhóm. Công nghệ tích hợp tự động, Continuous Integration.

3.3.3 Continuous Integration

Tích hợp tự động là khi có một sự thay đổi ở nhánh thì sẽ lập tức build, hoặc tích vào đối với ứng dụng web. Nhóm đã thực hiện được trên Visual Studio Team Service và build các lần push lên gần đây. Lợi thế của công cụ này là có thể biết được lỗi xảy ra khi nào và do ai push lên. Đồng thời, tạo một tag để lưu code tại thời điểm đó để có thể quay về.

Điều đầu tiên cần làm là phải biết cách build trên máy của ta. Sau đó hiện thực lại công thức đó trên các máy host hoặc cài một agent về máy xem như là host. Lúc này ta chỉ việc tạo công thức và nếu đúng, với mỗi lần chạy thì sẽ có thể tự động build và tự động tạo tag để ta quản lý.

Sau đây là một ví dụ. Có thể xem thêm thông tin chi tiết tại: <https://www.visualstudio.com/team-services/continuous-integration/>

Ví dụ:

3.3.4 Hiện thực phần mềm trên CommandPrompt

Bắt đầu thực hiện và thiết kế

Khi nhận được tin báo dự án, cả nhóm đã lên lịch họp và đã bắt đầu code các phần của sổ. Vào lúc này, do không có kiến thức nên việc compile được thực hiện một cách không đúng đắn khi include các file cpp vào để khi chạy

```
g++ -Wall -std=c++11 -c main.cpp -o LIBPRO.exe
```

Thì có thể ra được file chạy hoàn chỉnh.

Sau nhiều ngày nghiên cứu, đã tìm được cách include đúng và compile đúng. Nhưng từ đó lại mở ra một vấn đề khác khi không thể từng tay chạy g++ ra các file o rồi link lại với nhau. Từ đó nhóm ngừng lại trong việc code và tiếp tục nghiên cứu thêm.

Phân chia thư mục

Cùng lúc đó, nhóm cũng học cách phân chia thư mục, nhận thấy trên mạng có hai cách để phân chia:

```
include
src
và
project
    module A
        src
        include
```

```
module B
    src
    include
```

Và quyết định chọn theo cách ở trên. Vì nhóm chưa có kiến thức và kinh nghiệm trong việc code theo module.

Sử dụng JSON cho dữ liệu

Ban đầu với việc đọc dữ liệu từ file nhóm vẫn hiện thực rất tốt, nhưng khi tới vấn đề rằng sử dụng các file txt để lưu dữ liệu, một là tạo một hàm chức năng để hiện thực cách lấy dữ liệu trong file; lựa chọn hai là với mỗi file, tạo một hàm mới cho việc đọc và ghi file.

Với cả hai cách, không cách nào là dễ dàng và ngược lại khó nhận dạng thông tin để chỉnh sửa bằng tay. Từ đó, nhóm quyết định sử dụng một loại dữ liệu khác.

Sau khi tìm hiểu, nhóm rút lại các cách gồm có:

1. SQL
2. MongoDB (do thấy nhiều trên mạng)
3. thuần JSON

Với SQL và MongoDB, học cách để liên kết với dữ liệu trở nên khó khăn và về sau có thể cản trở. Vì thế nhóm quyết định tìm một thư viện để parse JSON nhanh và dễ sử dụng.

Sau khi tìm kiếm, một số cái tên được đưa vào sự chú ý như Rapid JSON, JSON CPP, và đặc biệt nhất là Nlohmann JSON, với cấu trúc dễ dàng cài đặt nhanh chóng chỉ bằng include một file hpp.

Và nhóm quyết định sử dụng JSON, sau đây là một số câu lệnh để sử dụng cơ bản:

```
using json = nlohmann::json;
json A = json::object(); // tạo một json object {}
json B = json::array(); // tạo một json array []
```

Để tạo một key trong A với giá trị item:

```
A["key"] = item;

{
    "key": item
}
```

Để lấy từ file, câu lệnh dùng như đọc file:

```
ifstream file(path, ios::in);
json C = NULL;
file >> C;
và xuất ra:
file << C.dump(4);
```

Để truy cập vào các phần tử,


```
D.at("key")
Và thêm vào [chỉ số] nếu nó là array.
D.at("key")[0].at("F00") = bar;

{
    "key": [
        "F00": bar
    ]
}
```

Build bằng Makefile

Quay lại vấn đề build dự án, nhóm đã tìm được cách thức build dự án nhanh hơn khi dùng makefile. Nhóm đã viết các công thức cho makefile sử dụng g++ và build được dự án bằng việc viết.

```
make tênfile.o
để build các file cpp thành o và bỏ vào trong thư mục build.

make libpro hoặc make
để link các file trong build lại.
```

Makefile hiện đã lỗi thời và nhóm đã xóa đi, tuy nhiên vẫn còn nằm trong lịch sử dự án trong các lần build thành công với tag từ 18 trở xuống.

Tạo dự án bằng CMake

Với việc xử lý dự án ở trên nhiều công cụ build khác nhau, Makefile phải được thay đổi. Từ đó nhóm lại đi tìm một hình thức khác để tự động hoá dự án. Và đã tìm được CMake.

CMake là công cụ sinh file project, với những ai dùng GNU, CLANG thì sẽ sinh ra Makefile, với những ai dùng Visual Studio thì sẽ sinh ra file sln cùng các file project khác, và tương tự với CodeBlocks, Eclipse, XCode, hay các IDE và các bộ biên dịch khác. Ở đây, nhóm có thiết lập trước với hai compiler là GNU và MSVC (cho Visual Studio).

Sau một khoảng thời gian tìm hiểu, nhóm đã viết được CMakeLists với mục tiêu là lấy tất cả file trong dự án và tạo ra Makefiles có thể build được dự án tốt nhất. Và cũng đã thử với Visual Studio để tạo được dự án với chia thư mục đúng như thư mục đã chia.

Để bắt đầu tạo dự án và build, theo các bước dưới đây. Giả sử đang ở thư mục gốc của dự án.

```
mkdir build & cd build
cmake ..
make
../bin/./LIBPRO
```

Nhóm đã tích hợp unit test vào nên nếu muốn build để test:

```
cmake .. -DTEST=ON
make
../bin/./LIBPRO_TEST
```

Build cho Linux

Với tính năng hỗ trợ Bash cho Windows 10, Linux Subsystem for Windows 10, nhóm đã nghiên cứu để tiến hành build trên Linux. Tuy có gặp một vài trở ngại trong việc sử dụng thư viện, nhưng vẫn có thể build được chương trình cho Unix.

3.3.5 Thiết kế giao diện trên Qt GUI

Thiết kế màn hình chính

Màn hình chính được thiết kế đơn giản với thanh menu gồm các nút: Profile-Announcement-Library-Settings-Role-Help.

Khu vực bên dưới thanh menu sẽ trình bày các sách có trong thư viện.

Các chức năng khác được thực hiện trên các cửa sổ con.

Thiết kế phần cửa sổ con

Mỗi cửa sổ con thực hiện một chức năng nhất định, và có các nút tương ứng với chức năng đó.

3.4 Thực hiện báo cáo về phần mềm

Bản báo cáo này được thực hiện trong suốt quá trình phát triển phần mềm, nhằm cập nhật liên tục các ý tưởng của nhóm. Bản báo cáo cùng với phần báo cáo trên trực tiếp sẽ khái quát rõ hơn về phần mềm của nhóm.

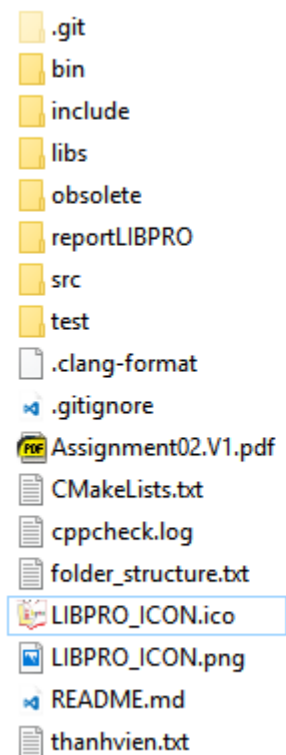
3.5 Lập tài liệu phát triển phần mềm

3.5.1 Tổ chức dự án

SIMple được triển khai thành 2 mảng: CommandPrompt và Qt

CommandPrompt

- bin: chứa các tập tin dữ liệu
- include: chứa các file header
- libs: chứa thư viện của test
- src: chứa các file cpp
- test: chứa các file test
- .clang-format: đồng bộ code giữa các thành viên
- .gitignore: tránh update những tập tin tạm
- CMakeList.txt: tập tin chuyên biệt để sử dụng CMake biên dịch chương trình
- cppcheck.log: tập tin kiểm tra sai sót cơ bản trong code
- folder_structure: tập tin ghi lại cây thư mục của dự án
- LIBPRO_ICON.png: tập tin hình ảnh chứa icon của chương trình
- README.md: tập tin hướng dẫn sử dụng chương trình
- README_INSTALL.md: tập tin hướng dẫn build và cài đặt chương trình
- thanhvien.txt: tập tin ghi lại thông tin người phát triển chương trình

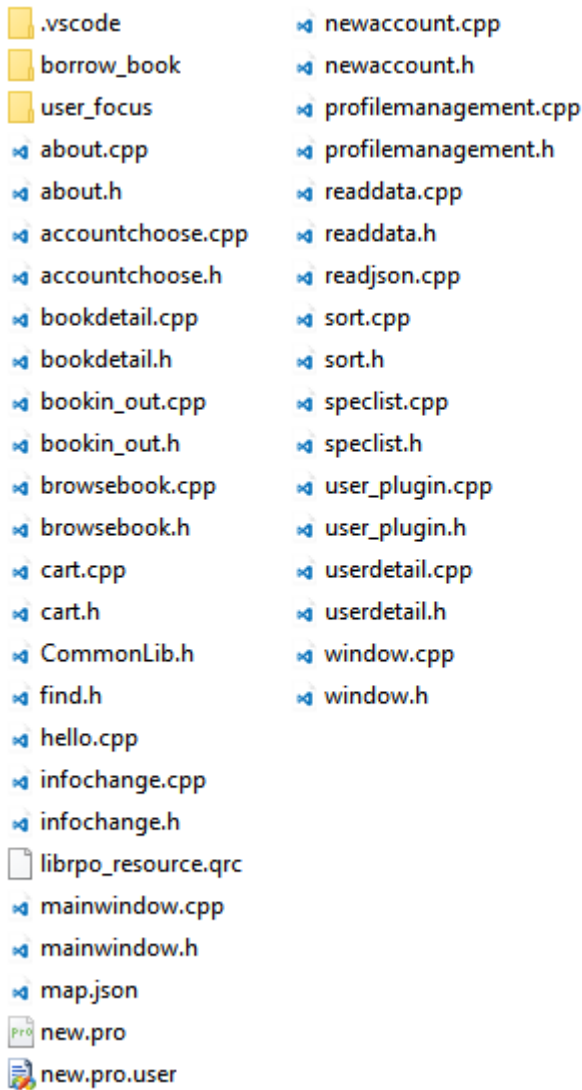


Hình 3.1: Các thư mục làm việc của CommandPrompt

3.5.2 Qt GUI

Vì đặc điểm phân vùng tập tin trên Qt, các tập tin trong thư mục làm việc trên Qt đều được lưu trên cùng một thư mục

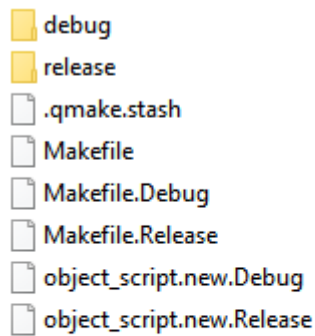
- user_focus: chứa các thư mục làm việc trên CommandPrompt, và đã có một số chỉnh sửa sao cho phù hợp với Qt GUI (như đã nêu ở trên)
- Tập tin new.pro là tập tin gốc để thực thi trên Qt
- Tập tin libpro_resource.qrc là tập tin để thêm biểu tượng (icon) cho phần mềm
- Còn lại là các tập tin header và source (.h và .cpp)



Hình 3.2: Các thư mục trong thư mục làm việc Qt

Trong quá trình biên dịch sẽ tạo ra thư mục mới là *build-new-Desktop_Qt_5_8_0_MinGW_32bit-Release*, trong đó chứa:

- debug: chứa các tập tin link (.o) và các tập tin moc(meta-object compiler) dùng để liên kết với các phần mở rộng của Qt trong quá trình sửa lỗi
- release: chứa các tập tin link (.o) và các tập tin moc(meta-object compiler) dùng để liên kết với các phần mở rộng của Qt trong quá trình thực thi
- .qmake.stash: ghi lại path các bộ qmake compiler
- Makefile: tập tin được tạo ra từ qmake trong quá trình biên dịch
- object_script.new.Debug ghi lại các tập tin liên kết (.o) khi chọn cách biên dịch Debug
- object_script.new.Release ghi lại các tập tin liên kết (.o) khi chọn cách biên dịch Release



Hình 3.3: Các thư mục trong thư mục thực thi Qt

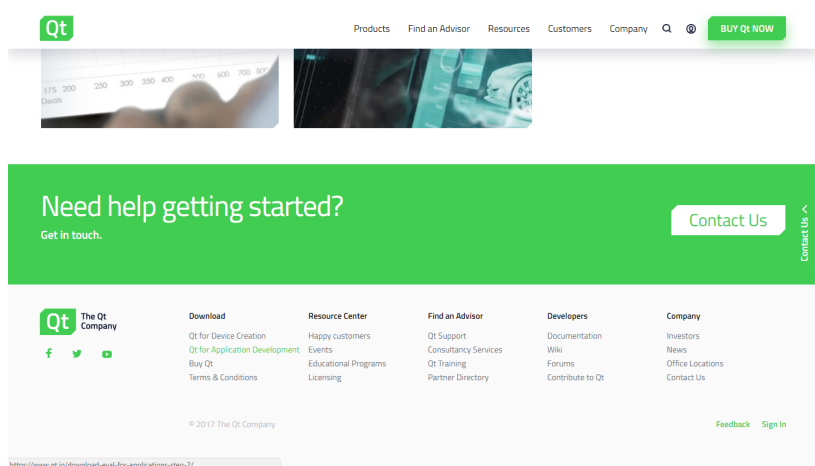
SIMple thực hiện đồng bộ cả hai mặt nên nhiều hàm trong Qt được thi triển từ các hàm trong CommandPrompt và sử dụng các tập tin header trong CommandPrompt.

3.5.3 Thư viện hỗ trợ

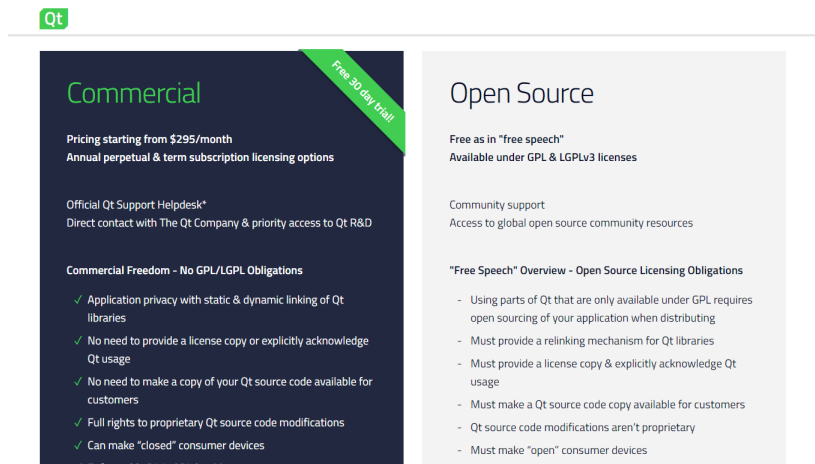
Qt GUI

Hướng dẫn cài đặt:

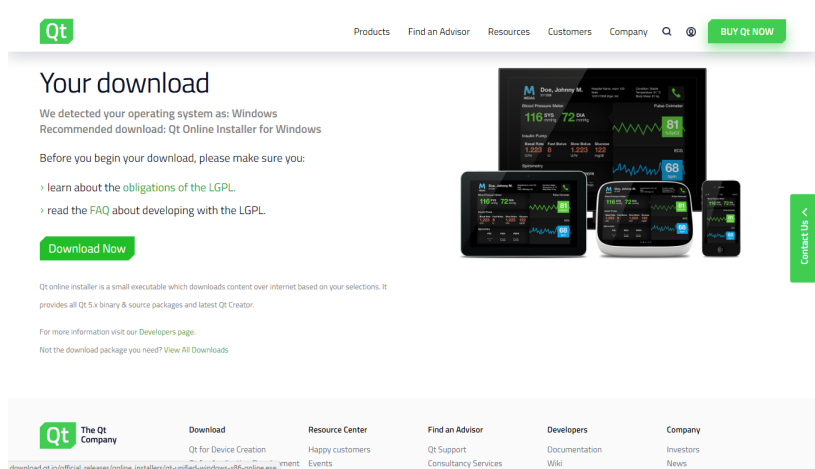
1. Vào trang web của Qt và chọn:



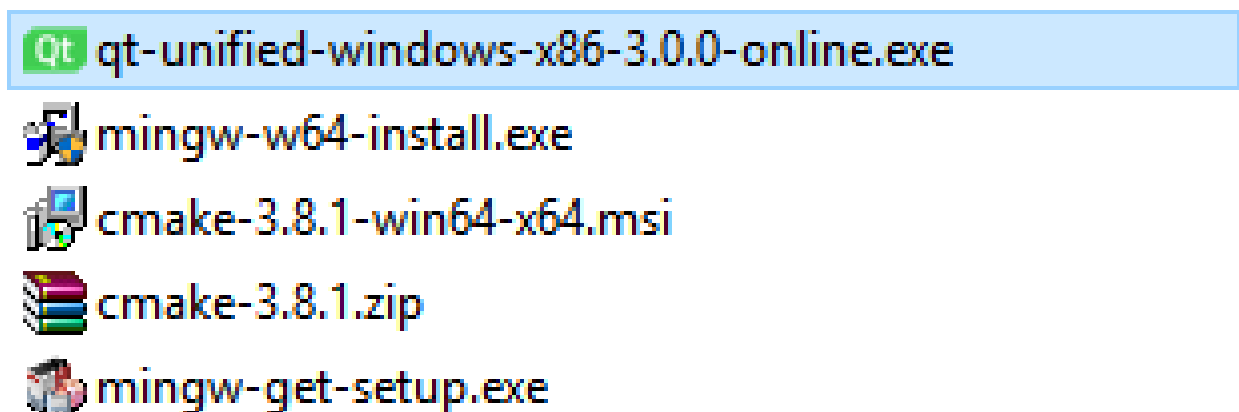
2. Sau đó, chọn ô bên phải, Open Source để tải bản miễn phí



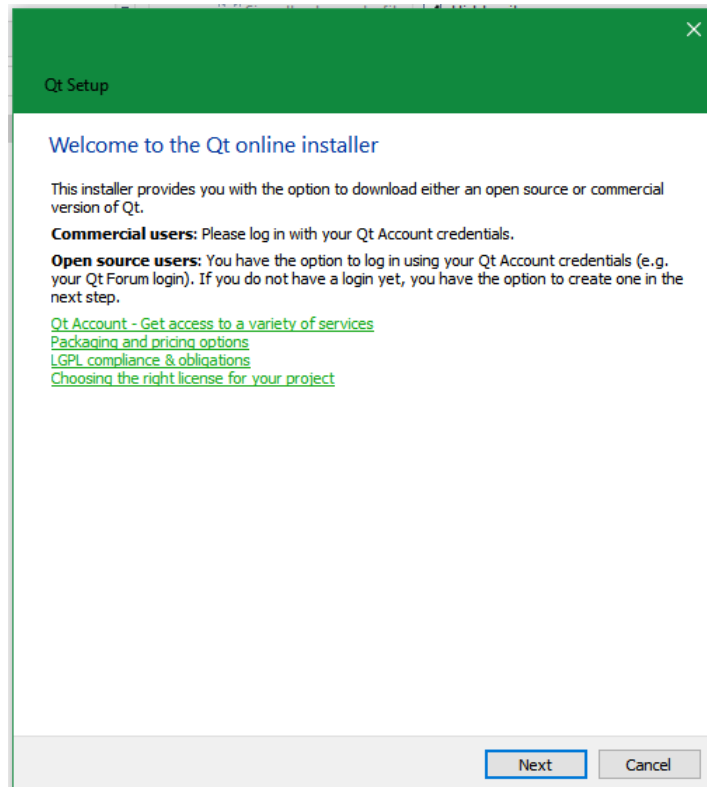
3. Nhấp nút Download và Qt sẽ hỏi thư mục muốn tải về, chọn thư mục và OK



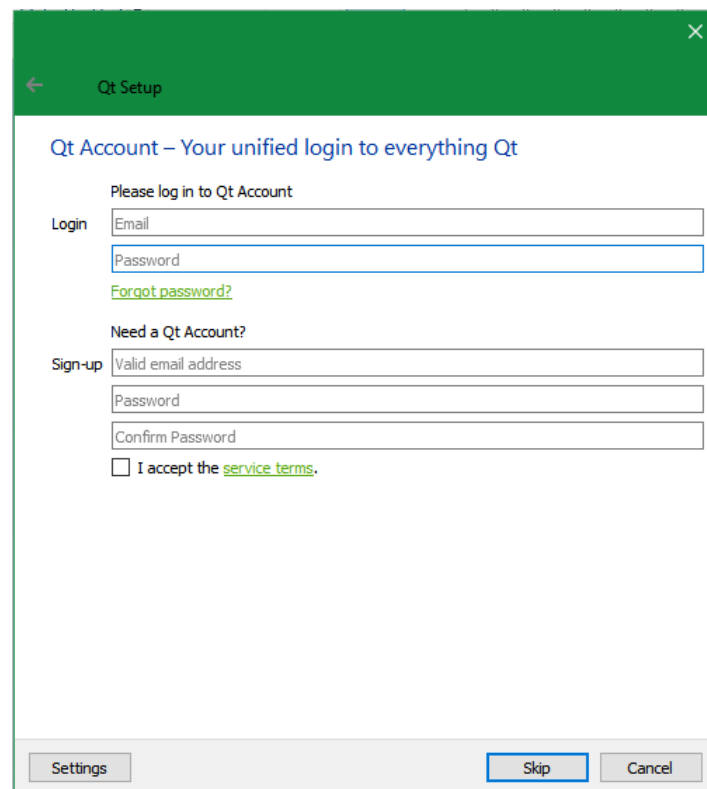
4. Sau khi tải về sẽ có tập tin như thế này



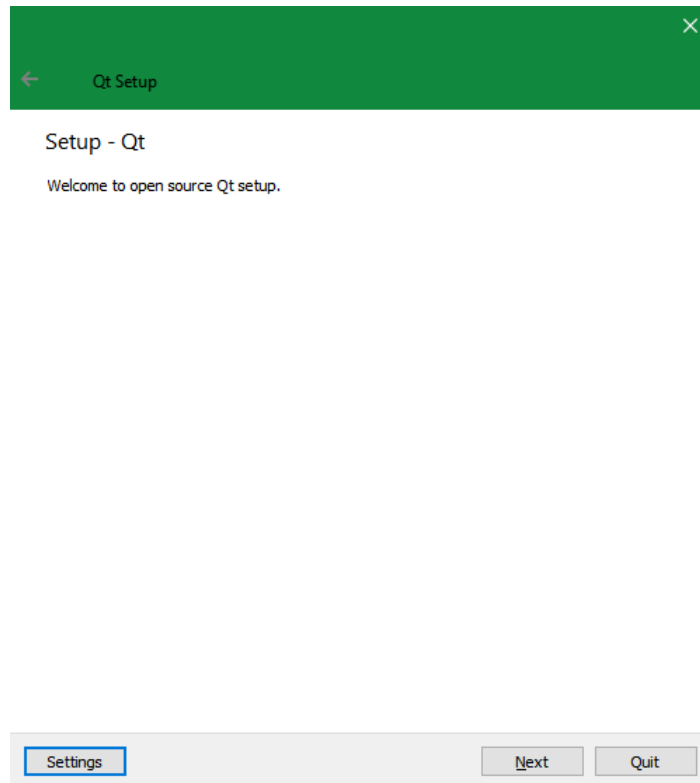
5. Nhấp vào tập tin, sẽ hiện ra cửa sổ như bên dưới, chọn Next



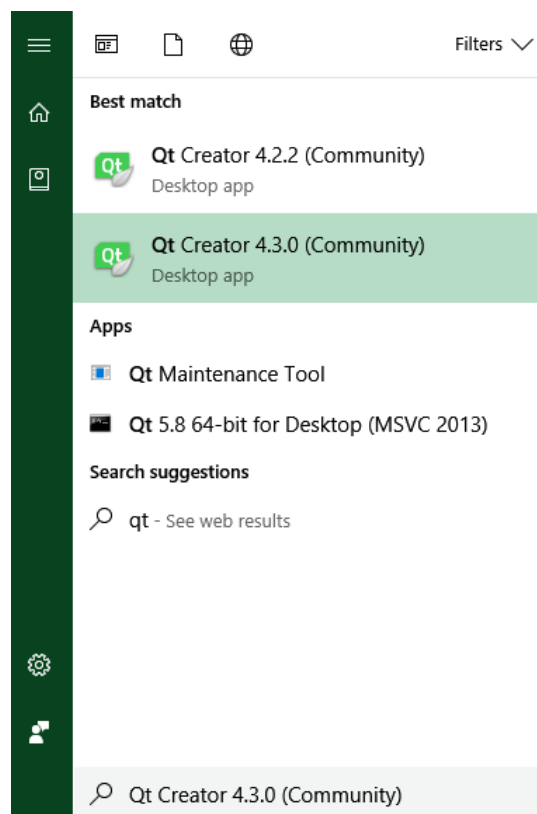
6. Bây giờ, Qt muốn người dùng phải có tài khoản trước khi cài đặt, chọn Login nếu đã có tài khoản hoặc Sign-up nếu chưa có tài khoản



7. Sau khi đã đăng nhập tài khoản, giờ tới cửa sổ Setup, chọn Next để cài đặt



8. Phần cài đặt có thể kéo dài vài giờ đồng hồ, nên người dùng có thể tranh thủ làm việc khác
9. Sau khi cài đặt xong, tại màn hình Windows, nhấn **Start->Qt** và chọn **Qt Creator 4.3.0 (Community)** (có thể tùy theo phiên bản Qt). Đây là môi trường để người dùng thực hiện lập trình giao diện.



GTest

Khi nói đến Unit Test trong C++, nhiều người có thể nghĩ tới assert.h. Nhưng nhóm đã không sử dụng thư viện mặc định đó, mà đã sử dụng Google Test hay còn gọi tắt là GTest. GTest được Google phát

triển và để tự do sử dụng khi để Github public với repo <https://github.com/google/googletest>. Để dùng, vào thư mục google test, chạy CMakeLists để ra makefile và chạy tiếp makefile, kết quả cho chúng ta 2 file .a, là file thư viện tĩnh, chúng ta chỉ việc link thư viện vào để sử dụng.

Và để tiện cho việc sử dụng trên chương trình, ta tạo một symbols là UNITTEST, và chính lại hàm main. Vì mọi hàm unit test đều phải được chạy trên một hàm main riêng và sử dụng các định nghĩa hàm trong dự án. Việc còn lại là tạo các hàm test và để trong thư mục test/src và link vào khi chạy CMakeLists với các định nghĩa IF (TEST) và dùng chức năng option(TEST OFF).

3.5.4 Biên dịch và chạy chương trình

Đối với CommandPrompt

Để biên dịch chương trình của nhóm, yêu cầu:

1. Một bộ biên dịch hỗ trợ c++11,
2. Hoặc g++ phiên bản 4.9 trở lên (vì thư viện nlohmann::json yêu cầu)
3. CMake phiên bản 3.8 trở lên (nếu sử dụng build cho Visual Studio, khuyến khích 3.8.1, vì nhóm đã thực hiện trên phiên bản này để build và thành công)
4. Nếu sử dụng Visual Studio, yêu cầu Visual Studio 2013 trở lên, vì sử dụng chức năng của C++11, phiên bản Visual Studio 2010 không thể build, build hoàn hảo trên Visual Studio 2017, các phiên bản còn lại chưa thử.

Đầu tiên là chạy cmake trên một folder mới để sinh ra các file cần thiết để chạy.

Nếu như là biên dịch theo g++ thì chạy makefile, nếu biên dịch theo Visual Studio thì chạy file sln.

Với Visual Studio, cần thực hiện chọn Set as start up project với dự án đang làm để khi build chạy được.

Đối với Qt Creator

sử dụng Build của IDE để thực hiện việc biên dịch.

3.5.5 Hiện thực các hàm chính yếu

Đăng ký và Đăng nhập

1. Đăng nhập

```
BEGIN
Load data
Choose SignIn
Enter username and password
IF (username or password empty)
    do nothing and return
ENDIF
IF (the data is empty)
    display to user that there is no data
ENDIF
IF (username is wrong)
    display cannot sign in
ENDIF
IF (password is wrong)
    display cannot sign in
ENDIF
Get user info as current user
Display Signed In
END
```

2. Đăng ký

```
BEGIN
Load Data
Choose Sign Up
Enter username
WHILE username exists
    display message
    input another username
ENDWHILE
Input normal info
Prompt advanced info
IF Yes
    Show advanced info
ENDIF
Choose Account package
IF default
    Account name = username
    Package = reader
ELSE
    input account name
    Choose package
    IF package = VIP
        create account with all package
    ENDIF
ENDIF
ENDIF
Generate random password
Show for user to take note
IF wrong
    END
ELSE
    Send Request
ENDIF

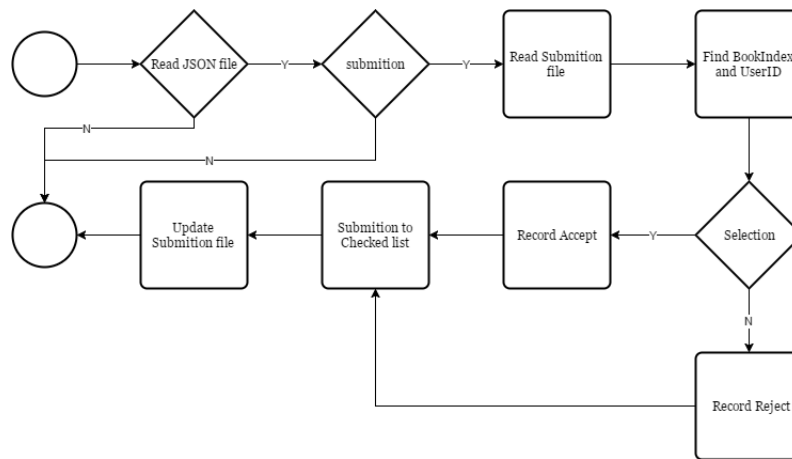
Librarian Login
Browse SignUp
IF accept
    move data to user
ELSE
    move data to rejected
END
```

Quản lý xuất nhập sách

```
BEGIN
Load data
User Signed In as Librarian
Choose add new book
Input ISBN
IF (ISBN is a book in data)
    Ask if add in storage
    IF true
        Book number + add in quantity
    ENDIF
ENDIF
Input info of new book
Update Book Data
END
```

Quản lý mượn trả sách

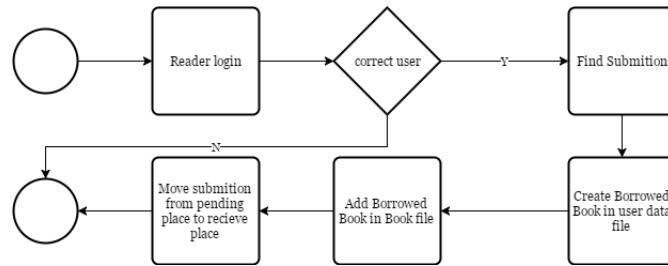
Function: BrowseBorrowBook



Hình 3.4: Sơ đồ khối hoạt động của hàm BrowseBorrowBook

```
if(!Read JSON file) escape;
if(!there is submission) escape;
else{
    Read Submissions in Submission file;
    Find BookIndex and UserID in specific Submission;
    Show selection{
        if(Accept){
            Record accepting new submission
            to pending place in BorrowLog file;
        }
        else Record accepting new submission
        to rejected place in BorrowLog file;
    }
    Move browsed submission to checked list in Submission file;
    Update Submission file;
}
```

Function: GiveBook

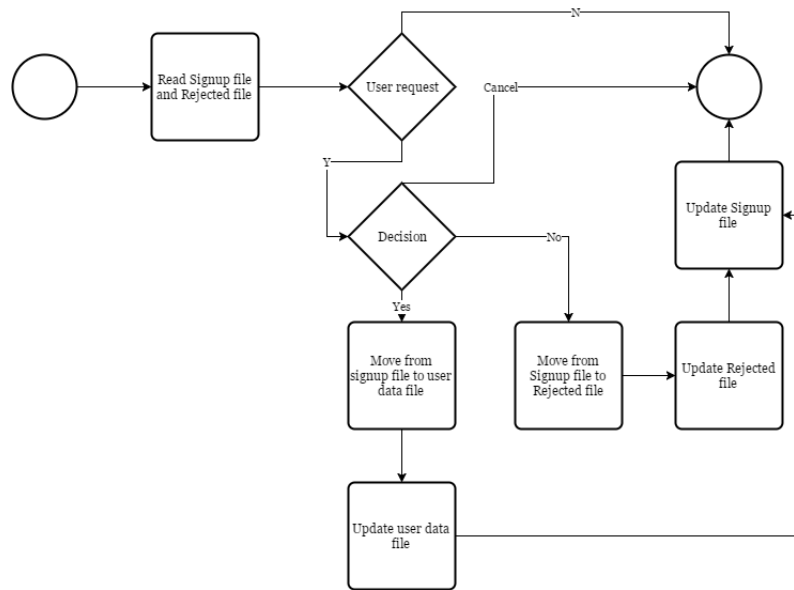


Hình 3.5: Sơ đồ hoạt động hàm GiveBook

```
Reader login before getting book (username and password);
if (correct User Submit){
    Find Submission of that Reader;
    Create Borrowed Book in Reader user profile;
    Add Borrowed Book in Book Library file;
    Move submititon from pending place to receive place;
}else escape;
```

Quản lý hồ sơ người dùng

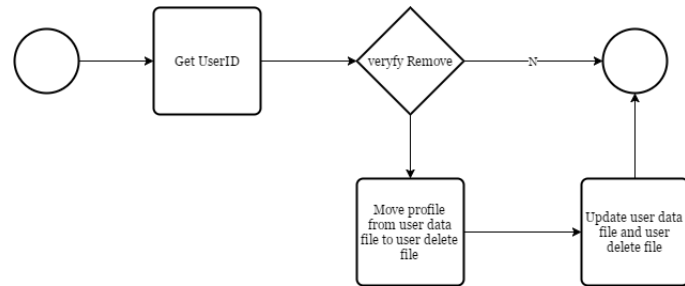
Function: VerifyNewUser



Hình 3.6: Sơ đồ hoạt động hàm VerifyNewUser

```
Read Sign-up file and user-rejected file;
if (no user request) escape;
else{
    if (user request decision is Cancel){
        escape;
    }
    else if (user request decision is Yes){
        Move user profile from sign-up file to user data file;
        update new user in user data file;
    }
    else if (user request decision is No){
        Move user profile from sign-up file to user-rejected file;
        update new user in user-rejected file;
    }
}
update signup file;
```

Function: DeleteUser



Hình 3.7: Sơ đồ hoạt động hàm DeleteUser

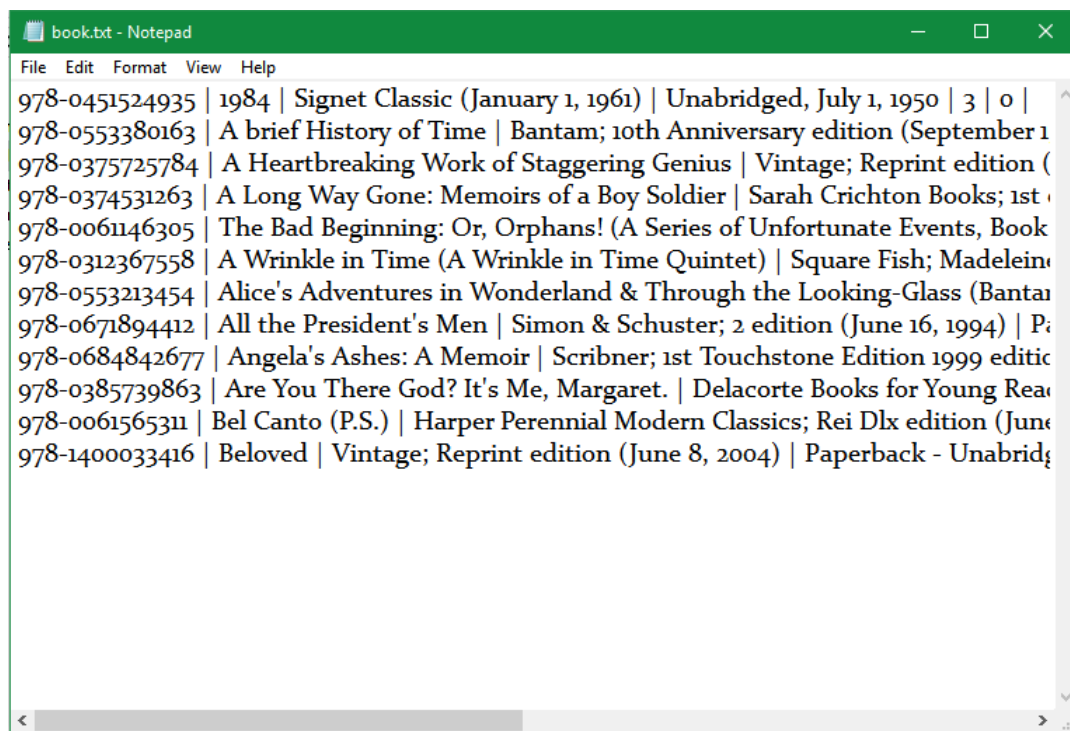
```
Get UserID to remove;  
if (verifyRemove is No) escape;  
else{  
    Move user profile from user data file to user deleted file;  
    update user data file and user deleted file;  
}
```


Chương 4

Các vấn đề nảy sinh trong tiến trình phát triển phần mềm

4.1 Lưu dữ liệu người dùng bằng file *.txt là một trở ngại

Trong quá trình làm việc và lưu dữ liệu, nhận thấy việc lưu dữ liệu trong file .txt rất khó truy cập và quản lý, cũng như rất khó hiểu cho người phát triển để kiểm tra dữ liệu. Cụ thể: nếu dữ liệu lưu như thế này:



Rõ ràng dù chỉ là dữ liệu sách nhưng lưu với .txt thì không thể nào hiểu được đâu là tên sách, đâu là nhà xuất bản, đâu là tên tác giả. Nên nhất thiết phải có cách lưu dữ liệu hữu hiệu hơn.

4.2 Việc đồng bộ hóa giữa dữ liệu và giao diện

Vì nhóm chủ yếu tập trung vào thiết kế dữ liệu, nên phần giao diện không thực sự là thế mạnh. Phần mềm chỉ mới được hiện thực các chức năng cơ bản, còn các tính năng mở rộng chỉ mới là ý tưởng.