
Báo cáo #3: Báo cáo cuối kỳ

Ứng dụng quản lí thư viện

Mô hình hoá phần mềm
(CTT530)

Tên nhóm: Nhóm 17 - Team tàn tạ

1. 1612197 – Khâu Thanh Hiếu
2. 1612209 – Nguyễn Hữu Hòa
3. 1612210 – Phạm Ân Hòa
4. 1612263 – Phạm Quang Huy
5. 1612285 – Dương Văn Khang

Thành phố Hồ Chí Minh, 28/12/2019

Lịch sử phiên bản

Ngày	Phiên bản	Mô tả	Tên tác giả
28/12/2019	1.0	Hoàn thiện báo cáo	Nguyễn Hữu Hòa

Link đến dự án :

- https://github.com/hhpr98/QLTV_TTT
- <https://trello.com/b/4Qy5PwWy/mhhpm-team-tan-ta>

Bảng phân công công việc

MSSV	Họ tên	Công việc	Tỉ lệ đóng góp	Ghi chú
1612197	Khâu Thanh Hiếu	- Biểu đồ deploy - Code Back end	20%	
1612209	Nguyễn Hữu Hòa	- Biểu đồ package - Tổng hợp PA3 - Code Front end	20%	
1612210	Phạm Ân Hòa	- Biểu đồ package - Code Front end	20%	
1612263	Phạm Quang Huy	- Biểu đồ deploy - Code Back end	20%	
1612285	Dương Văn Khang	- Biểu đồ component - Code Front end	20%	

Mục lục

Lịch sử phiên bản	2
Bảng phân công công việc	3
Không gian vấn đề.....	6
Không gian thiết kế.....	15
Không gian kiến trúc.....	19
1. Biểu đồ package.....	19
2. Biểu đồ component	20
3. Biểu đồ triển khai.....	21
Cài đặt và kiểm thử	22
Kết luận	23
Tài liệu tham khảo	24

Nội dung chính của báo cáo

Tổng hợp lại 3 không gian : không gian vấn đề (báo cáo 1), không gian thiết kế (báo cáo 2) và không gian kiến trúc (báo cáo 3).

Trình bày quá trình xây dựng ứng dụng của nhóm.

Không gian vấn đề

* Phát biểu bài toán

Thư viện trường đại học Khoa học tự nhiên sau một khoảng thời gian dài hoạt động cần nâng cấp hệ thống quản lý thư viện của mình. Nhóm 17 quyết định đề xuất thiết kế một phần mềm giúp cho việc quản lý thư viện trở nên dễ dàng hơn. Ở môn học Phân tích và quản lý yêu cầu phần mềm, nhóm đã phỏng vấn được một số cán bộ của thư viện trường Đại học Khoa học tự nhiên TP.HCM, qua đó nhận được một số yêu cầu như sau:

Phần mềm quản lý thư viện được thiết kế với vai trò giúp cho việc cán của thư viện quản lý dễ dàng hơn. Phần mềm yêu cầu được viết trên nền tảng web, chạy được với mọi trình duyệt, bao gồm cả Internet Explorer 11. Ứng dụng phải hỗ trợ hầu hết các loại hệ điều hành hiện nay, hỗ trợ các hệ cơ sở dữ liệu phổ biến (Microsoft SQL Server, MySQL,...).

Phần mềm phải thể hiện rõ một số chức năng chính như sau:

- Quản lý thông tin sách: Sách chính là vấn đề cốt lõi nhất của ứng dụng này. Việc quản lý thông tin sách cần phải lưu trữ thông tin vào cơ sở dữ liệu, cần bảo mật cơ sở dữ liệu để không ai có thể truy cập và sửa đổi trực tiếp được. Quản lý sách bao gồm việc thêm, xóa, sửa, cập nhật, độc giả chỉ có quyền xem sách, không có các quyền được nêu ở trên. Sách phải được phân loại theo từng mục, như thể loại, tác giả, thời gian, loại sách, thậm chí là giá tiền để tiện quản lý.

- Quản lý thông tin độc giả: Độc giả, hay người dùng là những người thường xuyên sử dụng ứng dụng, thường xuyên truy cập vào ứng dụng. Việc quản lý thông tin độc giả phải được đồng nhất, không nên dễ xảy ra lỗi vì sẽ ảnh hưởng đến trải nghiệm của người dùng. Quản lý thông tin độc giả bao gồm thêm, sửa, xóa (xóa theo dạng đánh dấu, chứ không xóa hoàn toàn khỏi cơ sở dữ liệu), cập nhật thông tin độc giả, hạn chế sử dụng chức năng hoặc cấm vĩnh viễn tài khoản của độc giả nếu vi phạm các nội quy của thư viện.

- Quản lý nhân viên: Nhân viên thư viện được cấp một tài khoản, không thể tự đăng kí như độc giả được. Nhân viên có thể thực hiện các quyền quản lý sách (thêm, chỉnh sửa thông tin, cập nhật sách), thực hiện quản lý độc giả, quản lý việc mượn trả sách.

- Tìm kiếm sách: Tìm kiếm sách theo đầu mục, theo tác giả, theo thể loại, theo ngày xuất bản. Thông tin tìm kiếm được phải hiển thị cho người dùng thấy, thực hiện phân trang nếu kết quả tìm kiếm quá nhiều.

- Thống kê, báo cáo sách: Hàng tháng hoặc hàng tuần, nhân viên thư viện có thể tổng hợp, thống kê tình trạng sách, tình trạng độc giả, tình trạng mượn trả sách của thư viện. Nếu có yêu cầu đột xuất, nhân viên thư viện có thể sử dụng ngay chức năng này bất cứ lúc nào. Sau khi tổng hợp và thống kê, có thể xuất báo cáo thành một file excel hoặc pdf để tiện theo dõi.

- Tạo phiếu mượn, trả sách tự động: phiếu mượn trả sách có thể được thực hiện thủ công bằng cách cung cấp/quét mã số sinh viên, hoặc thực hiện thông qua tài khoản. Độc giả có thể thực hiện việc này một cách tự động bằng thao tác trên tài khoản của mình: tìm kiếm sách, chọn mượn (nếu số lượng còn), nhận được mã, đến thư viện cung cấp mã cho nhân viên để mượn sách và lấy sách.

- Phân quyền người dùng: người dùng được phân làm administrator (có quyền quản lý mọi vấn đề), nhân viên (quản lý sách, độc giả và mượn trả sách), độc giả (chỉ có các quyền cơ bản: xem, tìm kiếm, mượn sách,...)

- Đăng kí: chức năng đăng kí bao gồm các hình thức đăng kí phụ
 - Đăng kí sử dụng tài khoản facebook, google.
 - Đăng kí sử dụng thẻ sinh viên (tự tạo tài khoản, cập nhật thông tin sau).
 - Đăng kí bằng hình thức thông thường (tên đăng nhập, mật khẩu), cần có nhập lại mật khẩu, mã xác nhận người dùng, hoặc captcha để tăng tính bảo mật.

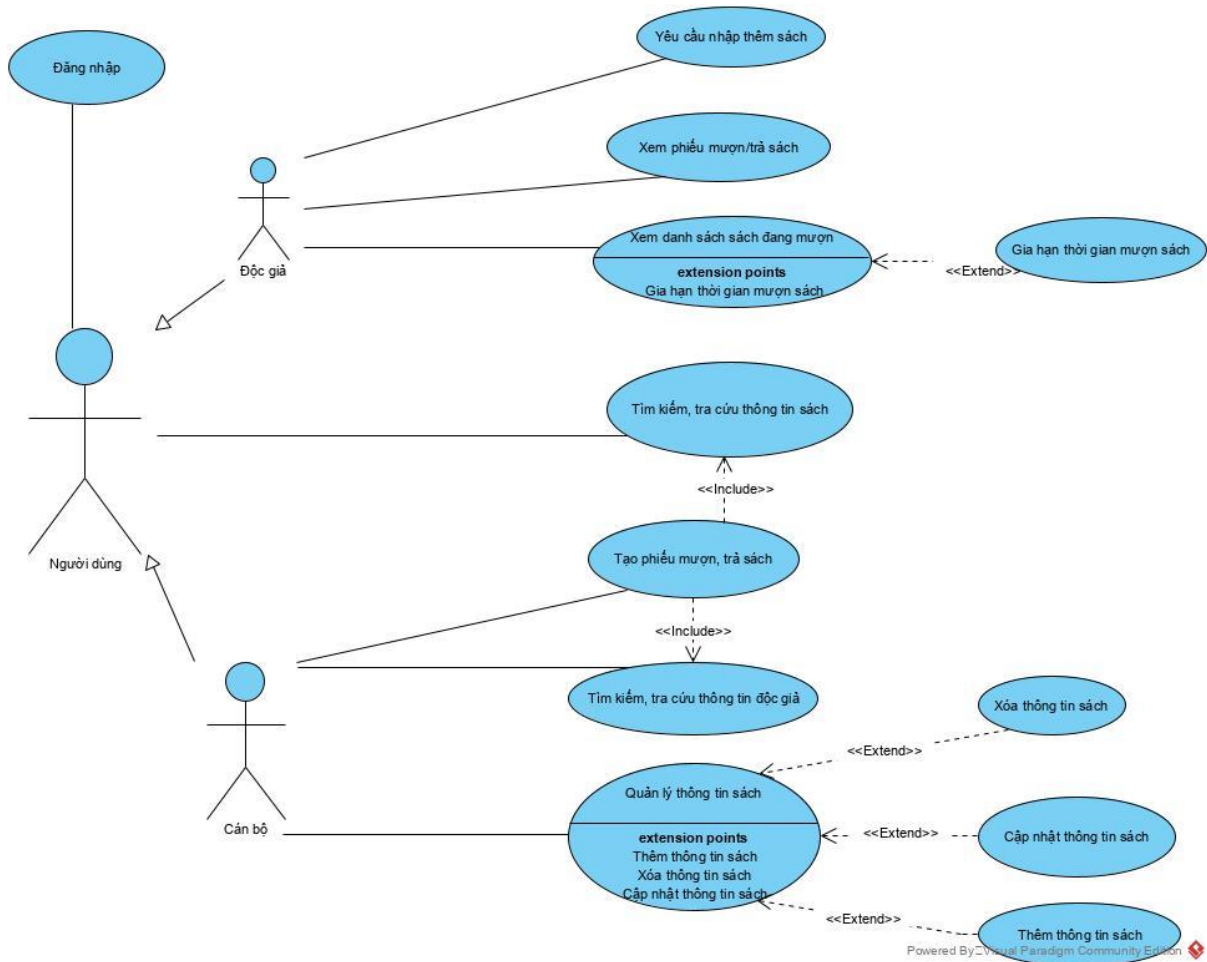
Đối với hình thức đăng kí 1 và 3, cần phải có sự kiểm duyệt, xác nhận tài khoản của nhân viên thư viện thì độc giả mới mượn được sách.
 - Đăng nhập: việc đăng nhập cũng được thực hiện thông qua 3 hình thức
 - Đăng nhập bằng tài khoản mạng xã hội: facebook, google.
 - Đăng nhập thông qua tài khoản thông thường.
 - Đăng nhập bằng quét thẻ: ứng dụng cần phải có cơ sở dữ liệu của sinh viên trường, để đảm bảo rằng nhận diện được sinh viên khi sử dụng máy quét thẻ.
 - Các vấn đề về đăng nhập:
 - Quên mật khẩu: cần có chức năng quên mật khẩu, người dùng muốn lấy lại mật khẩu thì phải yêu cầu đến hệ thống, hệ thống sẽ gửi mã khôi phục về email/điện thoại đã đăng kí. Nhập mã khôi phục và đặt lại mật khẩu để truy cập.
 - Reset mật khẩu: trong trường hợp không có số điện thoại hoặc email, độc giả có thể trực tiếp đến thư viện để nhờ nhân viên thư viện xác nhận danh tính, qua đó reset lại mật khẩu.
 - Đổi mật khẩu: độc giả thực hiện đổi mật khẩu và xác nhận lại mật khẩu, cần nhập mật khẩu cũ trước khi muốn đổi sang mật khẩu mới.
 - Tương tác với độc giả: ở mỗi cuốn sách sẽ có mục yêu thích (đánh giá sách hay), subscribe (theo dõi tác giả), comment (nhận xét về sách), rating (đánh giá sách ở thang điểm 5).
 - Liên kết với các thư viện khác: Chức năng nâng cao, đồng bộ sách giữa các thư viện, đồng bộ người dùng, người mượn có thể mượn sách ở thư viện này và trả sách ở thư viện kia.
- Ngoài ra, để đảm bảo chương trình hoạt động tốt, ứng dụng cần chạy liên tục 24/24 (dĩ nhiên có thể nghỉ để bảo trì), giao diện thân thiện với người dùng, phù hợp với đối tượng trẻ tuổi (người dùng chính), thời gian truy xuất của các yêu cầu nhanh, không chậm trễ quá 5s đối với các thao tác, đảm bảo không bị quá tải. Mật khẩu của người dùng phải được bảo mật tuyệt đối, kể cả khi lưu trong cơ sở dữ liệu phải được hash hoặc đặt password.
- * Yêu cầu chức năng:**

ID	Trọng số	Mô tả
REQ-1	Quản lí sách	Thêm, sửa, xóa, cập nhật sách
REQ-2	Thống kê, báo cáo sách	Thống kê xem sách nào đã mượn, còn lại bao nhiêu, mượn bao sách trong khoảng thời gian nào, đã trả chưa,...
REQ-3	Tìm kiếm sách	Tìm kiếm sách hiện có trong thư viện
REQ-4	Tạo phiếu mượn/trả sách tự động	Phiếu mượn trả sách chỉ cần quét thẻ sinh viên/hoặc nhập thẻ sinh viên.
REQ-5	Đăng kí độc giả	Đăng kí tài khoản cho độc giả (chủ yếu là sinh viên, công nhân viên trong trường)
REQ-6	Quản lí thông tin độc giả	Sửa, cập nhật thông tin (tất cả), xóa, ban (cấm) độc giả, vô hiệu hóa tài khoản (admin)
REQ-7	Phân quyền	Phân quyền tài khoản (dành cho admin)
REQ-8	Tương tác với độc giả	Thông qua các hình thức comment, rating, like, subscribe, tùy chọn hình thức.
REQ-9	Liên kết với các thư viện khác	Chức năng nâng cao: đồng bộ sách giữa các thư viện, đồng bộ người dùng, người mượn có thể mượn ở thư viện này và trả ở thư viện kia.
REQ-10	Chức năng đăng nhập	Đăng nhập sử dụng tài khoản, đăng nhập facebook, google, đăng nhập bằng quét thẻ,...
REQ-11	Đổi mật khẩu	Đổi mật khẩu
REQ-12	Quên mật khẩu/reset mật khẩu	Recover lại mật khẩu
REQ-13	Quản lí nhân viên	Thêm, sửa, xóa

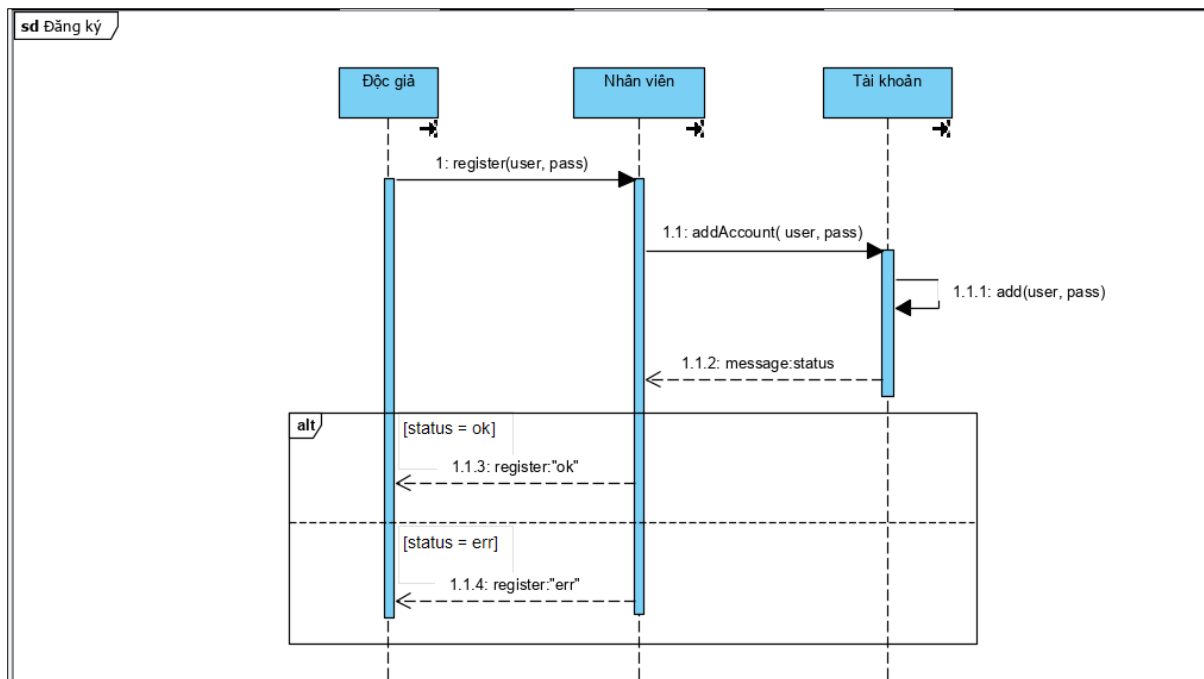
*** Yêu cầu phi chức năng:**

ID	Trọng số	Mô tả
REQ-1	Thời gian truy xuất nhanh (hiệu xuất ứng dụng)	Dưới 5 giây cho lần lần truy xuất CSDL, refresh, tìm kiếm, load,...
REQ-2	Xử lí ổn định trên cả windows, linux và macOS	Tương thích mọi hệ điều hành
REQ-3	Hỗ trợ IE11	Tương thích mọi trình duyệt phiên bản cao nhất.
REQ-4	Tính ổn định	Hoạt động 24/24, 7 ngày/tuần, thời gian ngưng hoạt động không quá 10%.
REQ-5	Giao diện đẹp, thân thiện	Dễ làm quen, dễ học sử dụng
REQ-6	Loại CSDL	MS SQL/MySQL/MongoDB/ProgestSQL

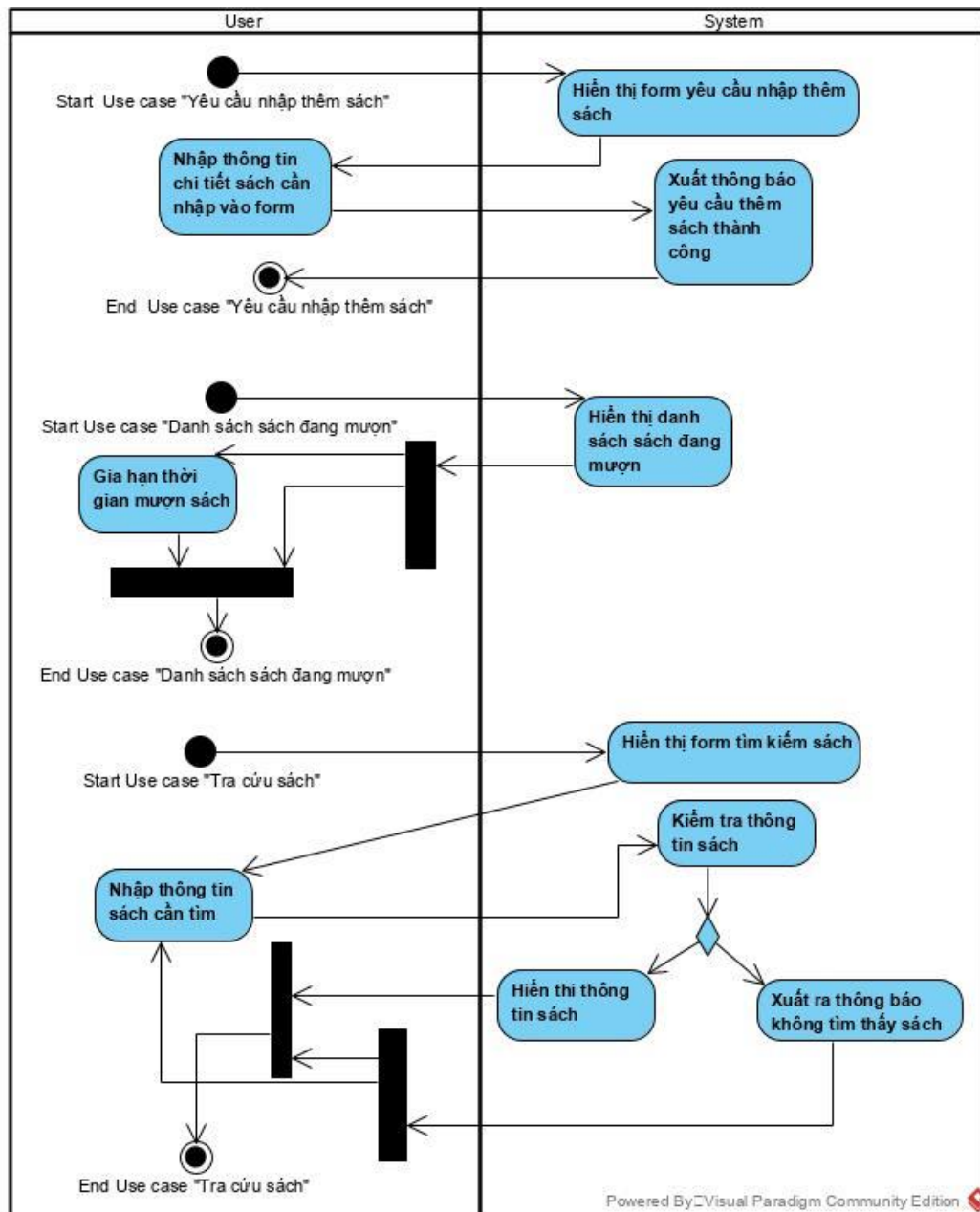
*** Use case:**

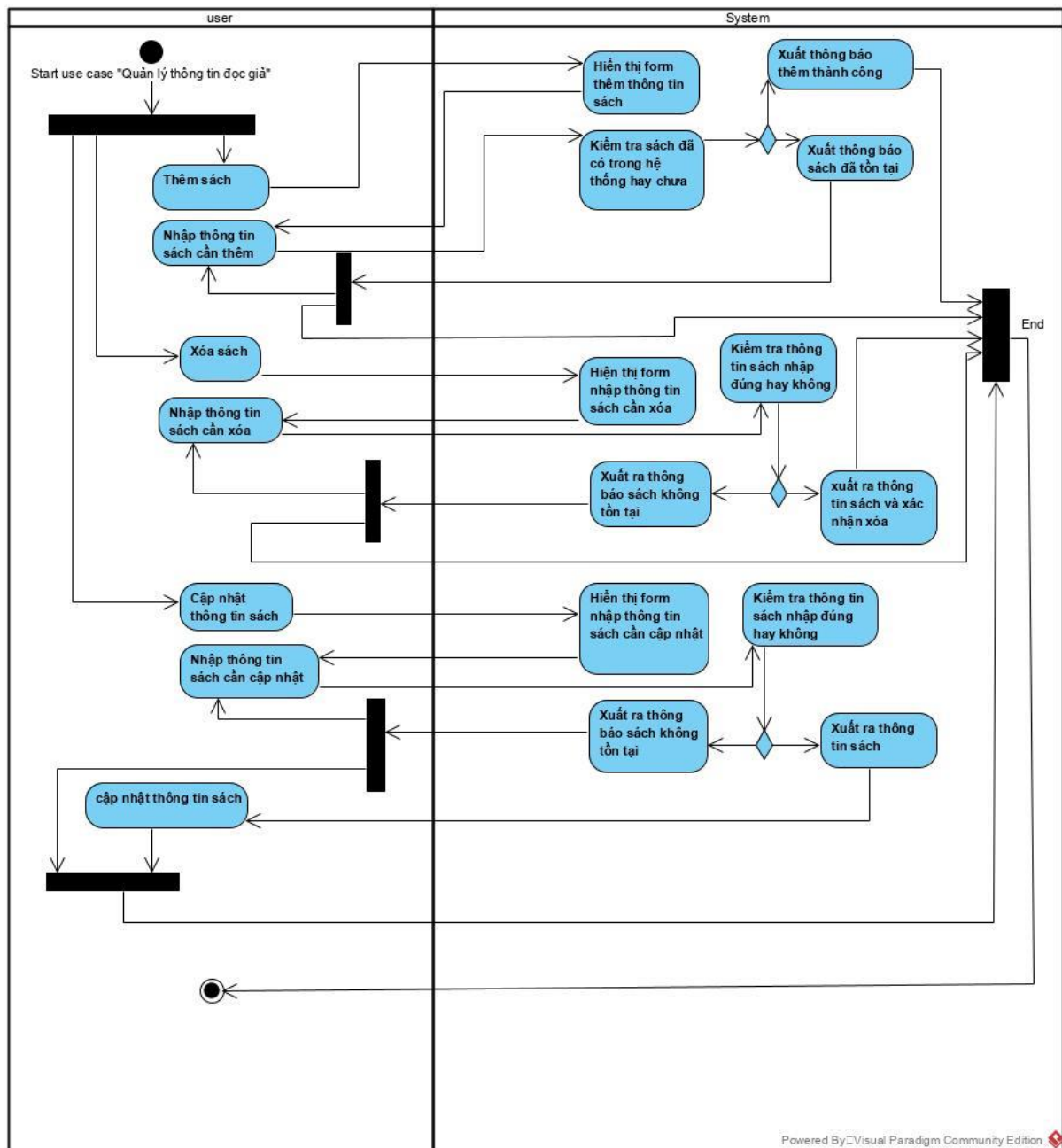


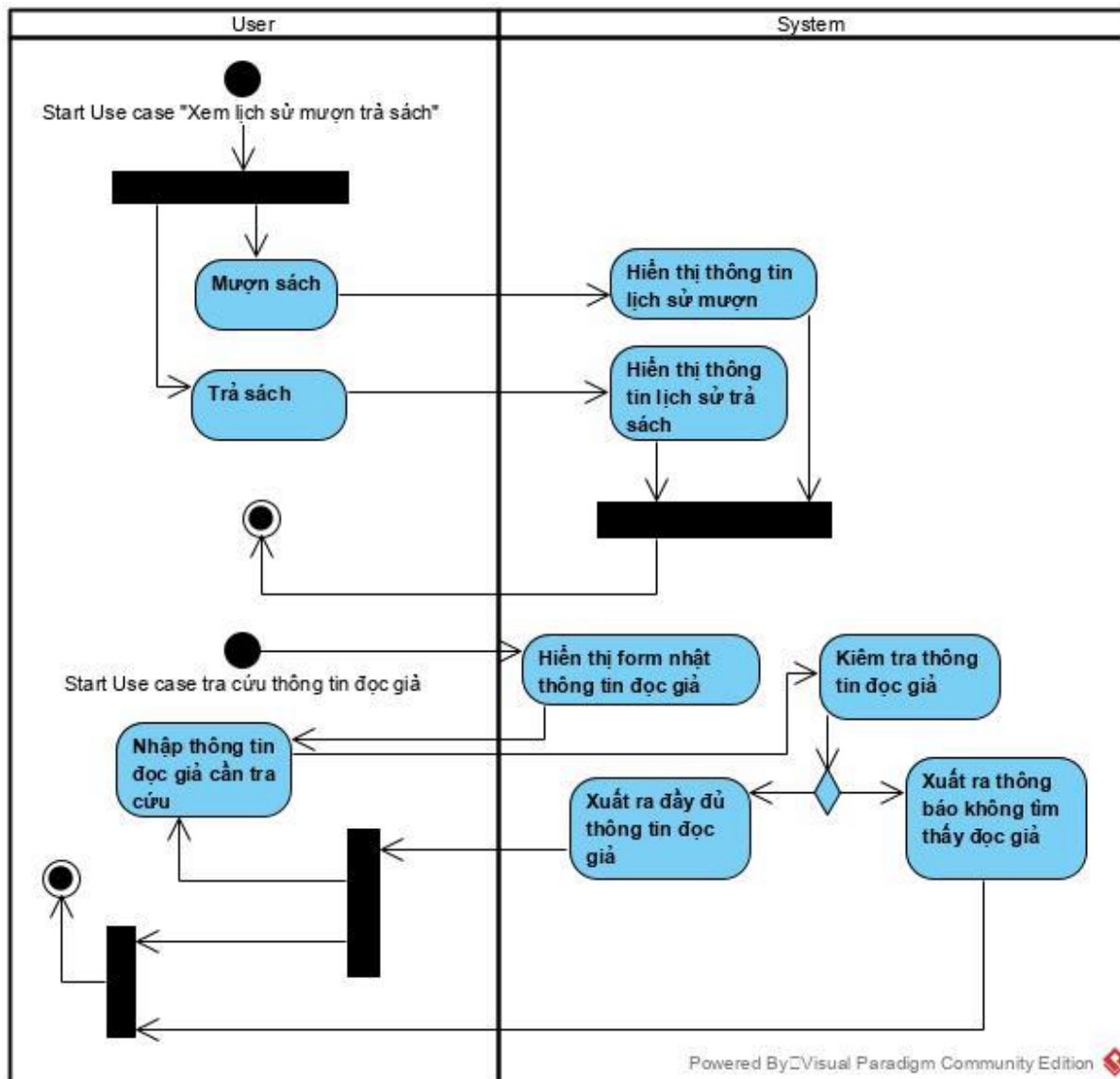
*** Biểu đồ tuần tự:**

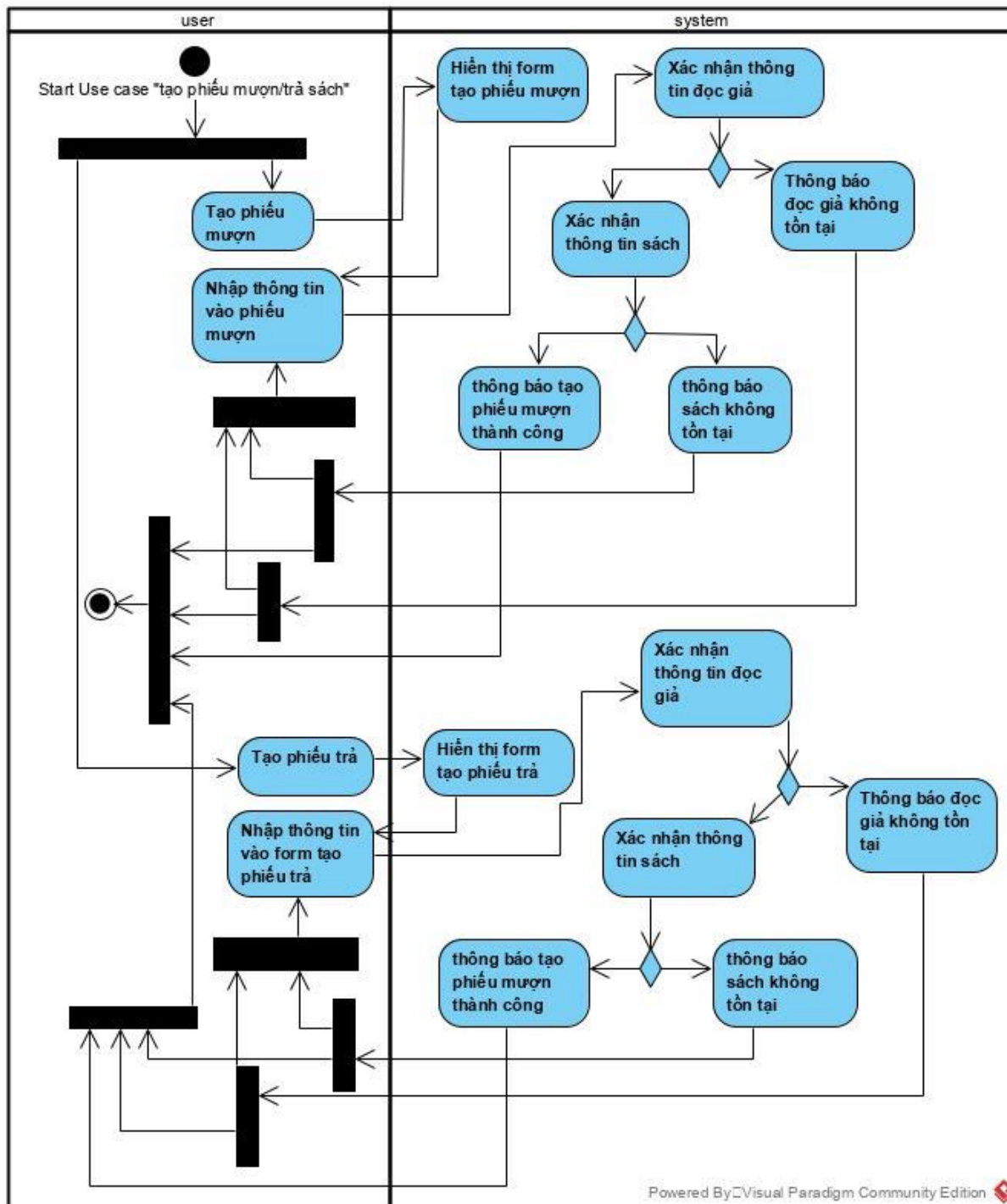


*** Biểu đồ hoạt động:**

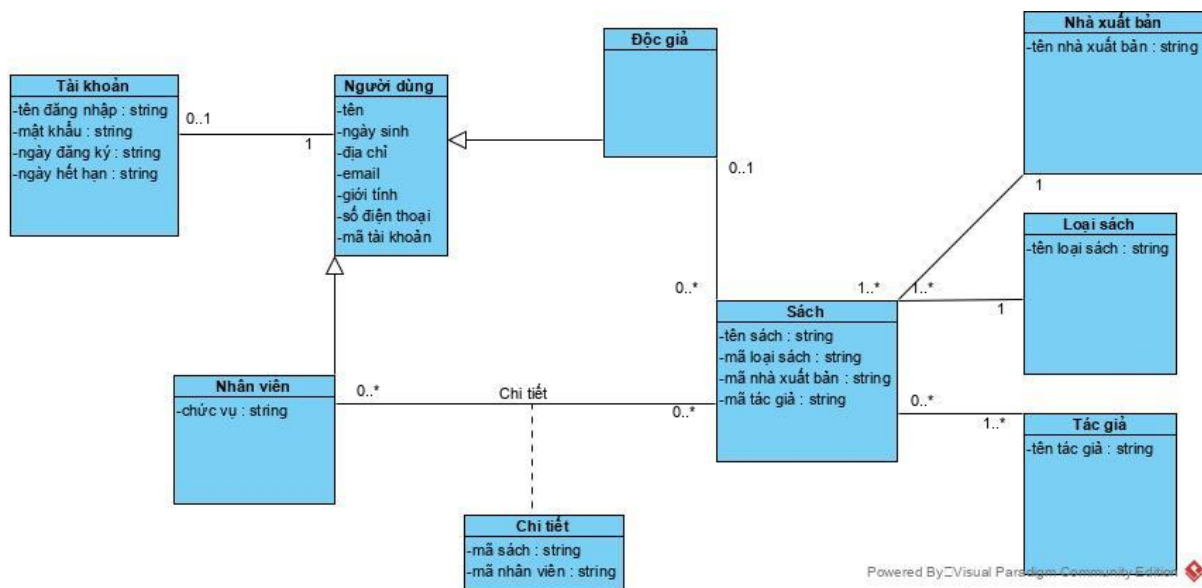








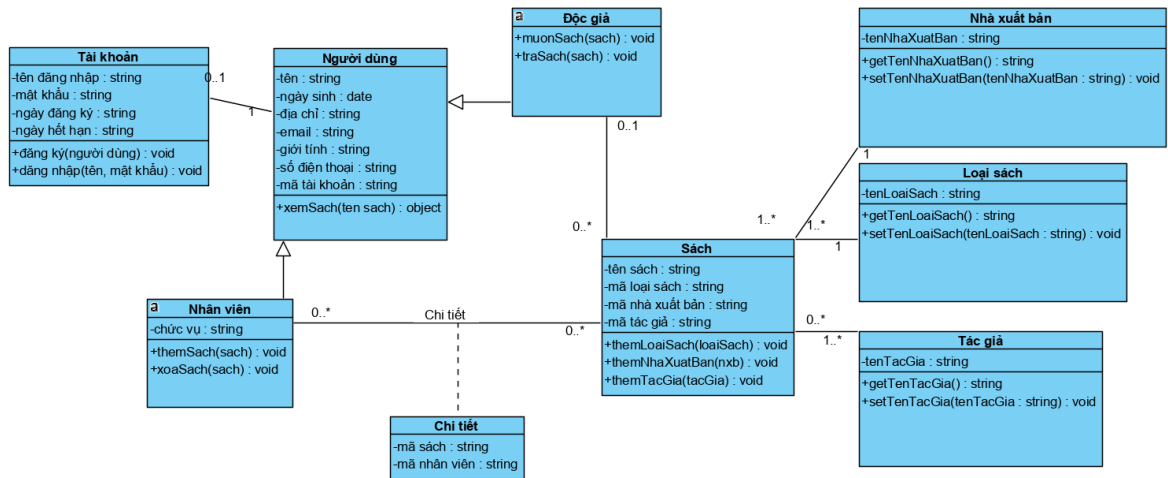
* Biểu đồ lớp:



Powered By Visual Paradigm Community Edition

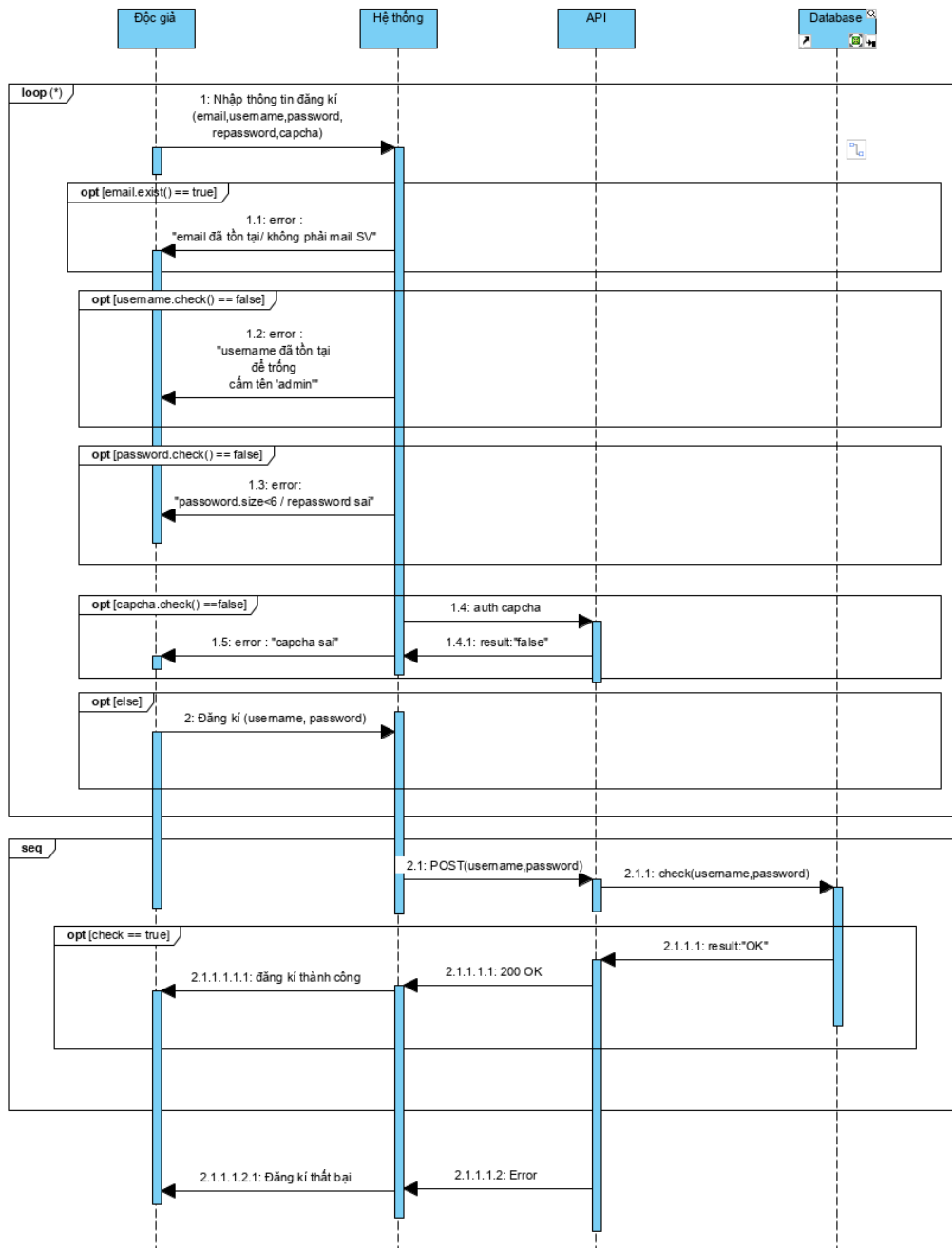
Không gian thiết kế

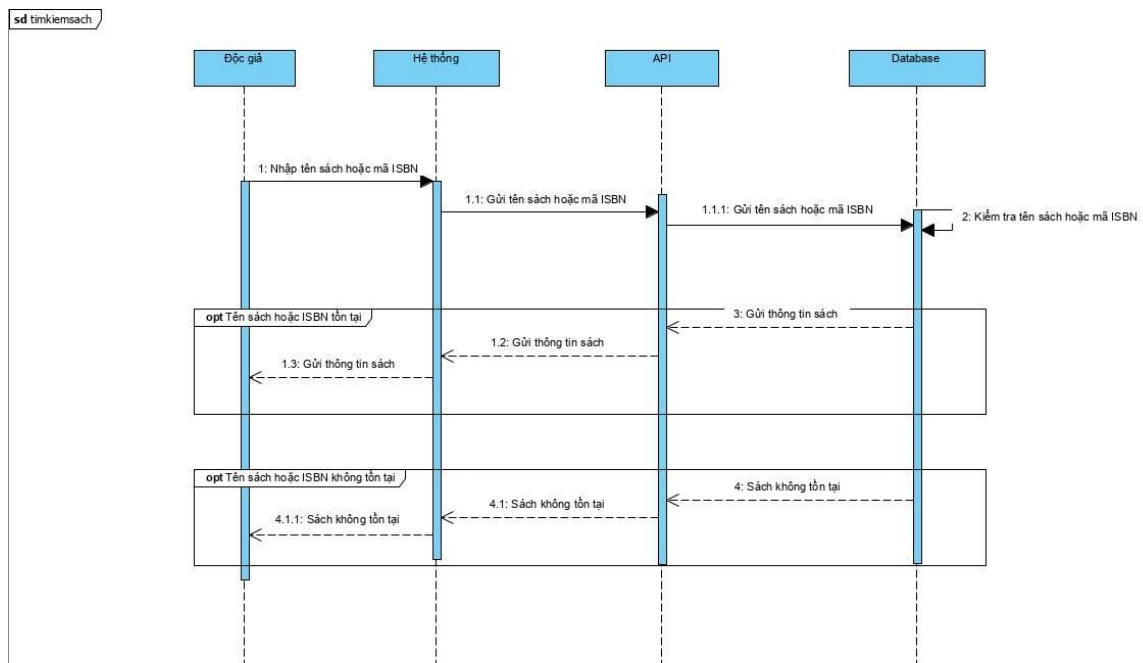
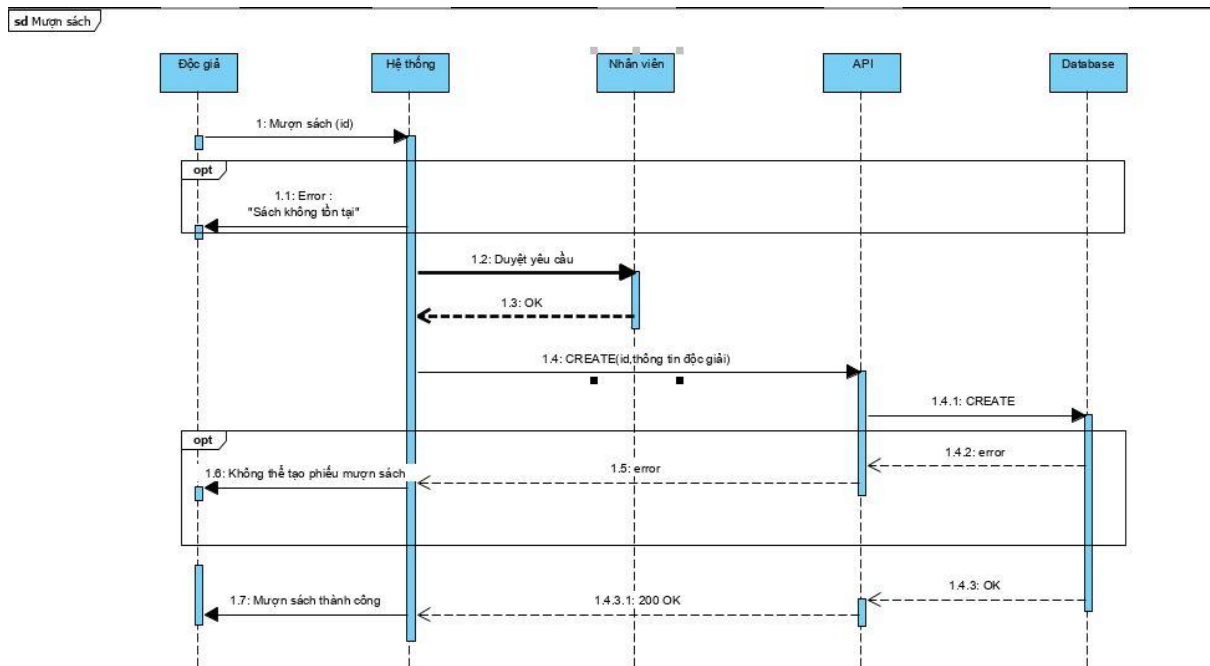
* Biểu đồ lớp ở mức chi tiết:



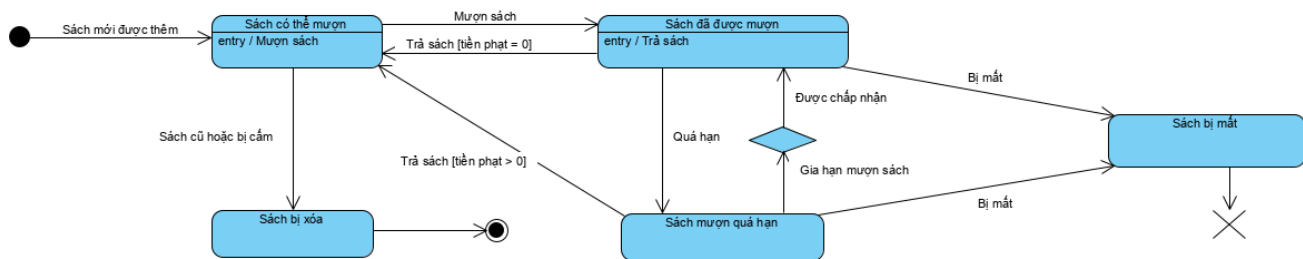
* Biểu đồ tuần tự:

sd Đăng kí

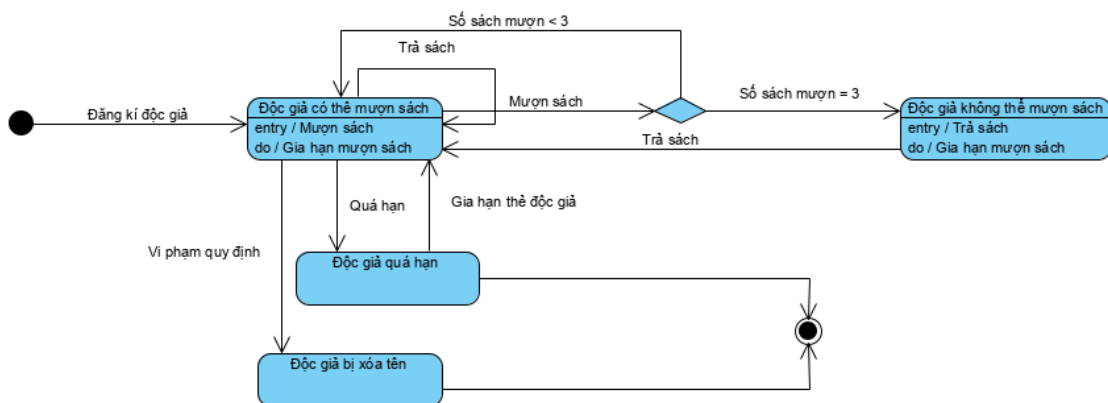




*** Biểu đồ trạng thái:**



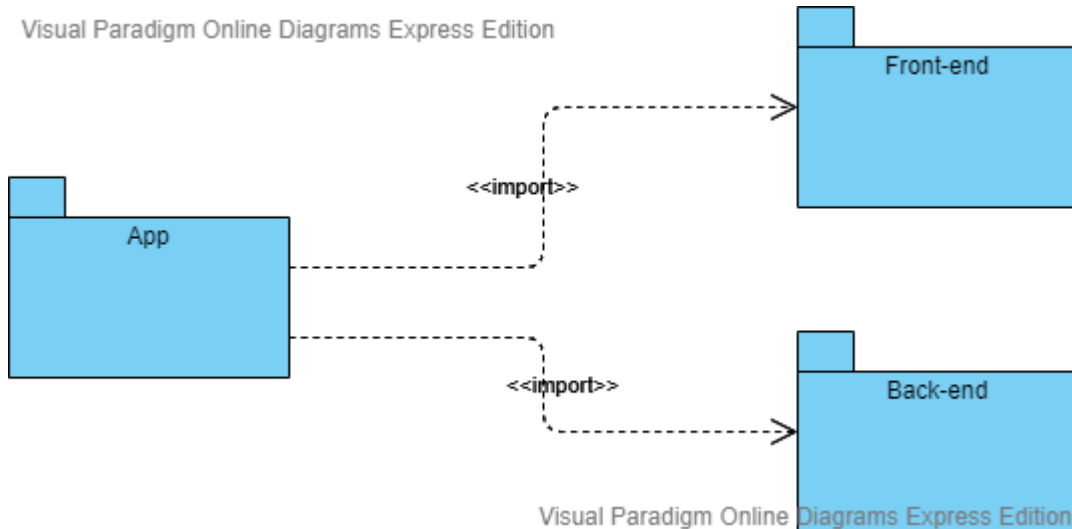
H1. State diagram Sách



H2. State diagram Độc giả

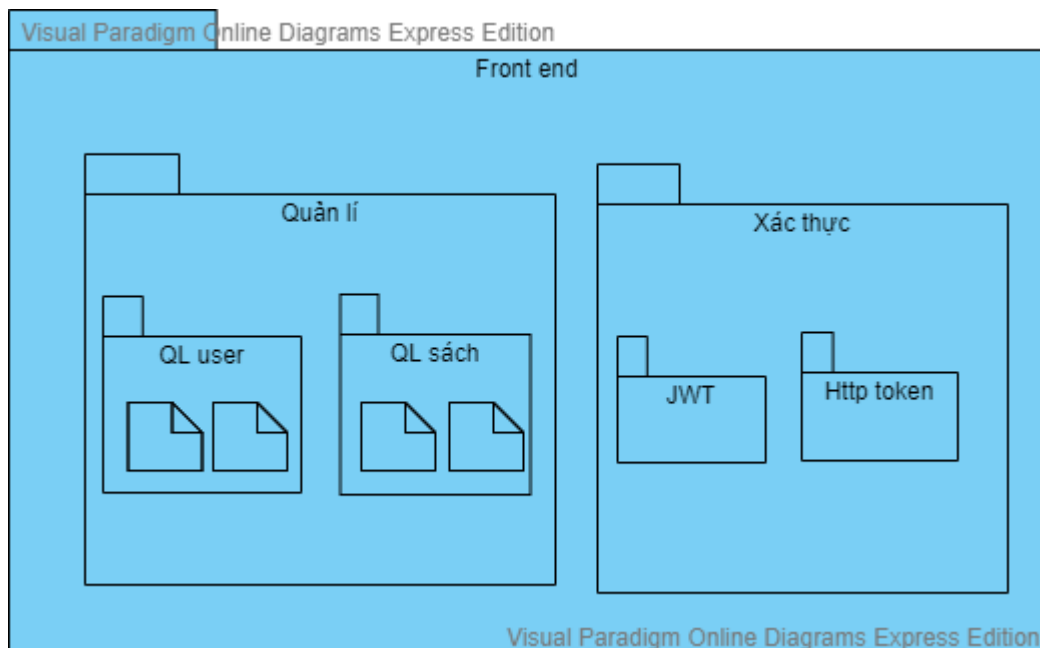
Không gian kiến trúc

1. Biểu đồ package

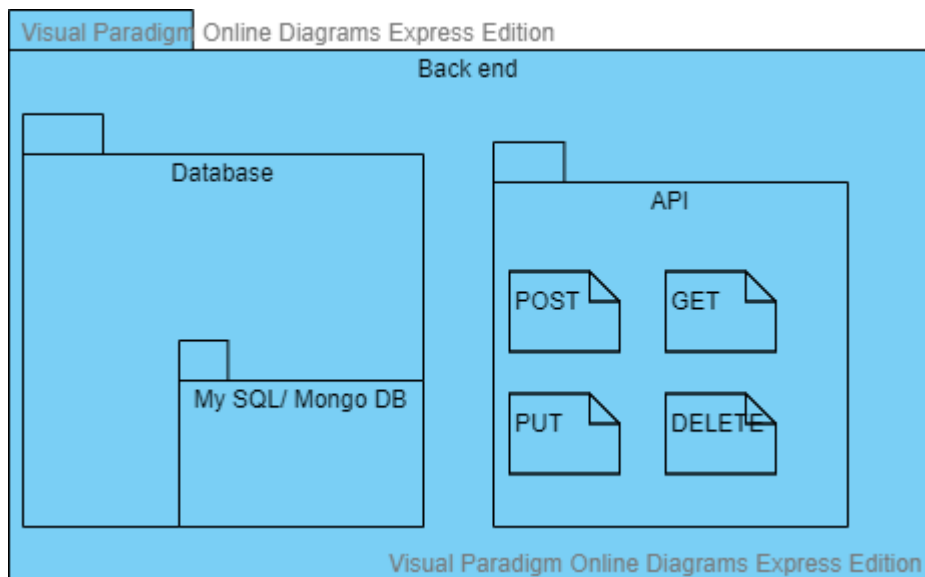


Package App gồm 2 package:

- Back end : quản lí các package về phần xử lí.
- Front end : quản lí các package về phần giao diện.

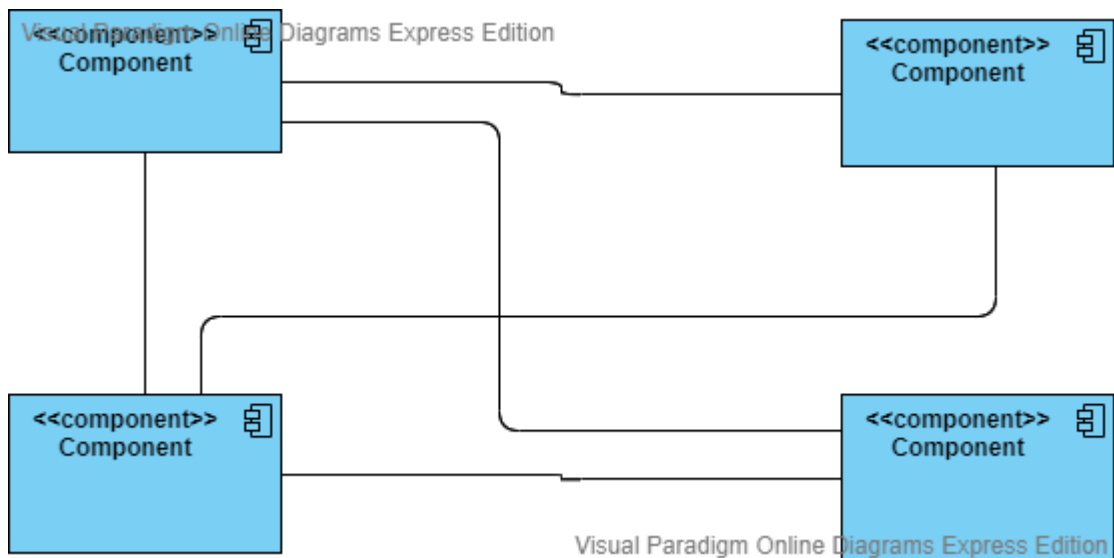


- Package quản lí :
 - + Quản lí user : tương ứng với class user.
 - + Quản lí sách : tương ứng với class book.
- Package xác thực: thực hiện xác thực người dùng và xác thực quyền.



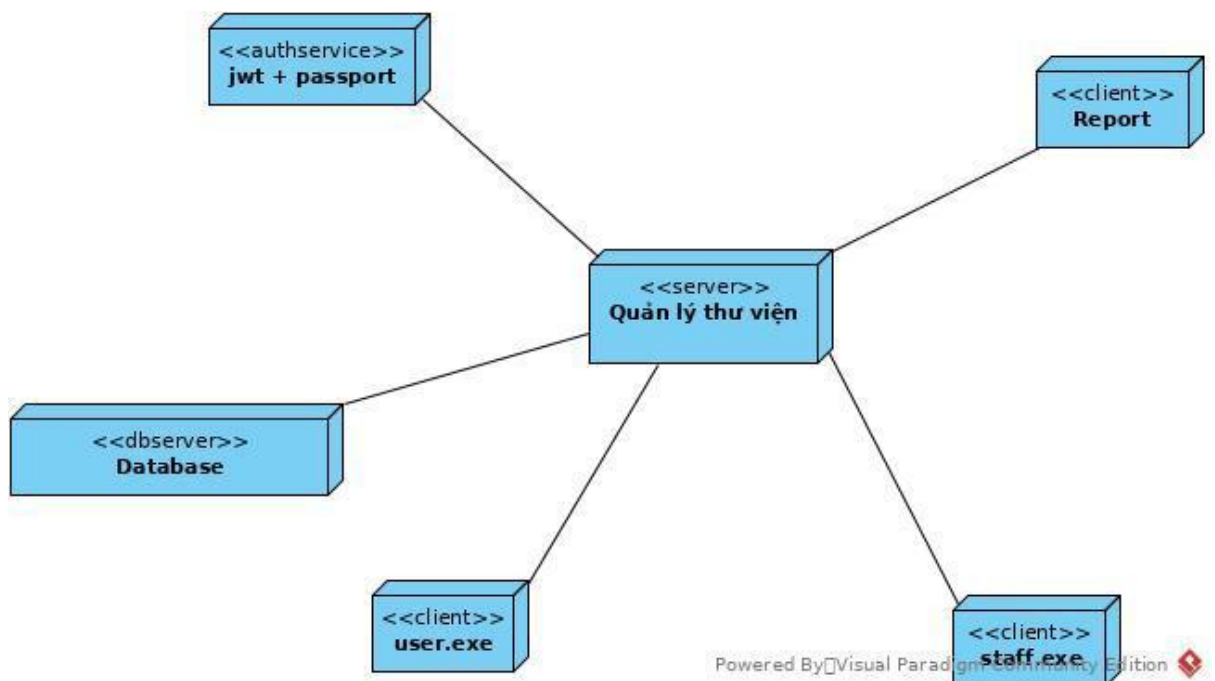
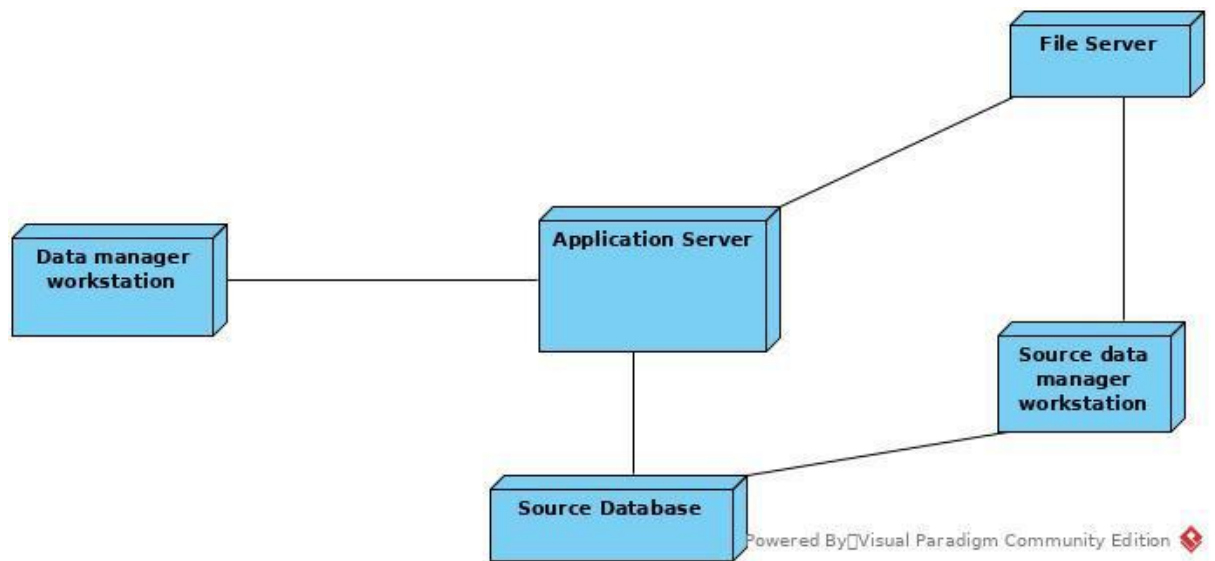
- Package database : quản lí thao tác với database.
- Package API : quản lí các API.

2. Biểu đồ component



- Component User: Chứa các component của người dùng, component này tương tác với Book và Staff. Component này chứa các thể hiện của người dùng và liên quan đến người dùng. Interface: User.
- Component Staff: Nhân viên quản lí thư viện, có quan hệ chặt chẽ với component user và quản lí các component Book. Interface: Book.
- Component Book: Quản lí sách và các component liên quan đến sách. Interface: Book.
- Component Auth: Component có nhiệm vụ duy nhất là xác thực người dùng. Interface: httpToken, jwt.

3. Biểu đồ triển khai



Cài đặt và kiểm thử

- Cài đặt:

- + Sử dụng framework NodeJS và ExpressJS view hbs.
- + Chia 2 thành viên code back-end và 3 thành viên code front-end.
- + Kiểm thử : test UI (đưa cho thành viên khác test) và test API (post man).
- + Code theo mô hình đã đề ra.

- Chưa mô hình:

- + Code chưa hoàn chỉnh nên chưa deploy lên hoàn toàn.
- + Staff và User gần giống nhau => Không nên có user, chỉ nên có staff.

Kết luận

* Thu được:

- Sản phẩm hoàn thiện.
- Kỹ năng lập trình theo mô hình.
- Ít phát sinh lỗi hơn khi kiểm thử.

* Thuận lợi:

- Đã học phát triển ứng dụng web, phân tích thiết kế phần mềm, phân tích và quản lý yêu cầu phần mềm.
- Có kiến thức lập trình căn bản.
- Hiểu biết cách sử dụng các công cụ vẽ diagram.

* Khó khăn:

- Nhiều yêu cầu, chức năng khó mô hình.
- Đồ án song song với nhiều môn khác nên dành ít thời gian.
- Nhiều biểu đồ khó vẽ.
- Visual paradigm chỉ có 30 ngày free.

* Hướng khắc phục:

- Tập trung code vào những ngày cuối.
- Thảo luận cùng nhau giải quyết vấn đề.
- Chia code, chia biểu đồ ra vẽ.
- Dùng trello, git quản lý dự án.

Tài liệu tham khảo

- FIT HCMUS
- Slide bài giảng, mô hình hóa Phần mềm.