

Trường Đại Học Sư Phạm Kỹ Thuật Tp.HCM Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam

Khoa Công Nghệ Thông Tin

Độc lập – Tự do – Hạnh phúc

-----***-----

-----***-----

Tp. Hồ Chí Minh, ngày ... tháng 06 năm 2015

NHIỆM VỤ KHÓA LUẬN TỐT NGHIỆP

Họ và tên sinh viên: **VÕ THÁI NGỌC**

MSSV: 11110089

NGUYỄN DUY TẤN PHÁT

MSSV: 11110093

Chuyên ngành: Công nghệ Thông tin.....Lớp: 111101.....

Giáo viên hướng dẫn: **TS ĐẶNG THANH DŨNG**.....

Ngày giao đề tài: 09/03/2015Ngày nộp đề tài: 26/06/2015

1. Tên đề tài:

WEB APP HỖ TRỢ VẼ, LƯU TRỮ VÀ CHIA SẺ CONCEPTMAP DÙNG SPRING

2. Các số liệu, tài liệu ban đầu

3. Nội dung thuyết minh và tính toán

Xây dựng được trang web hỗ trợ : Vẽ Concept Map, lưu trữ Concept Map, Share Concept Map cho 1 hay nhiều User khác, load 1 Concept Map đã vẽ trước đó, xem danh sách Concept Map, Xem danh sách task, Xem danh sách Share, login hệ thống, logout hệ thống, upload document, tạo Task, chấm điểm Concept Map, login thông qua Facobook, hỗ trợ lưu trữ ở amazon s3

4. Sản phẩm

Website chia vẽ, lưu trữ, chia sẻ conceptmap

[Http://springtilesopenshift-conceptmapjwd.rhcloud.com](http://springtilesopenshift-conceptmapjwd.rhcloud.com).....

Trưởng khoa

Giáo viên hướng dẫn

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày..... Tháng..... Năm.....

Giáo Viên Hướng Dẫn

(Ký, ghi rõ họ và tên)

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày..... Tháng..... Năm.....

Giáo Viên Phản Biện

(Ký, ghi rõ họ và tên)



LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn khoa Công nghệ thông tin, Bộ môn

Tiểu Luận Chuyên Ngành, Trường Đại học Sư Phạm Kỹ Thuật Tp.HCM đã tạo điều kiện thuận lợi cho chúng em thực hiện đề tài này.

Chúng em cũng xin được gửi lời cảm ơn chân thành nhất đến thầy Đặng Thanh Dũng, người đã tận tình trực tiếp chỉ bảo và hướng dẫn em thực hiện đề tài này.

Bên cạnh đó, chúng em xin gửi lời cảm ơn đến các thầy cô giáo, những người đã giảng dạy, tạo điều kiện cho em tích lũy được những kiến thức quý báu trong những năm học qua.

Dù đã cố gắng hoàn thành đề tài đúng yêu cầu, nhưng do thời gian hạn hẹp và khả năng còn hạn chế nên chắc chắn sẽ có những thiếu sót không tránh khỏi. Chúng em mong nhận được sự thông cảm và tận tình chỉ bảo của các thầy cô và các bạn

TP. Hồ Chí Minh, tháng 6, năm 2015

Nhóm sinh viên thực hiện

Võ Thái Ngọc

Nguyễn Duy Tấn Phát

Mục lục

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....	2
NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN	3
LỜI CẢM ƠN	4
Mục lục	5
Mục lục danh sách bảng	7
Mục lục danh sách hình ảnh	8
Phần A: MỞ ĐẦU	10
Giới thiệu đề tài	11
Chương 1. Tính cấp thiết	12
1.1 Khảo sát hiện trạng	12
1.2 Mục đích nghiên cứu	15
1.3 Đối tượng và phạm vi nghiên cứu	16
1.4 Ý nghĩa khoa học và thực tiễn.....	16
Phần B: NỘI DUNG	17
Chương 1. Cơ sở lý thuyết và Công nghệ	18
1.1 Các công nghệ	18
1.2 Thư Viện	40
Chương 2. Thiết kế.....	42
2.1 Kiến trúc hệ thống.....	42
2.2 Thiết kế sơ đồ User case.	43
2.3 Thiết kế Sequence Diagram.....	47
2.4 Thiết Kế Cơ Sở Dữ Liệu.....	60

2.5 Thiết Kế Class Diagram	61
2.6 Thiết kế giao diện và xử lý	65
Chương 3. Cài đặt và kiểm thử.....	75
3.1 Cài Đặt.....	75
3.2 Test.....	77
3.3 Kết quả khảo sát từ người dùng.	89
3.4 So sánh với các sản phẩm tương đương	90
Phần C: KẾT LUẬN.....	92
Tìm hiểu dịch vụ cloud storage của amazon (Simple storage service - S3)	93
Những chức năng đã hoàn thành	93
Ưu điểm.....	93
Nhược điểm.....	94
Giải pháp cho hướng phát triển cho tương lai.....	94
Tài liệu tham khảo.....	95

Mục lục danh sách bảng

Table 1.1: Bảng khảo sát.....	12
Table 2.1: Các xử lý trên giao diện Login.....	65
Table 2.2: các xử lý trên giao diện Add account	66
Table 2.3: Các xử lý trên giao diện Share Concept Map.....	67
Table 2.4: Danh sách các xử lý trên giao diện Show List Task.....	68
Table 2.5: Danh sách xử lý trên giao diện Show List Share	69
Table 2.6: Danh sách xử lý trên giao diện Draw ConceptMap	70
Table 2.7: Danh sách xử lý trên giao diện Save	71
Table 2.8: Danh sách xử lý trên giao diện Make Task.....	74
Table 3.1: Test Login	77
Table 3.2: Test Trang List map student	82
Table 3.3. Test Trang Create	86
Table 3.4: Test trang admin	88
Table 3.7: So sánh sản Phẩm tương đương	90

Mục lục danh sách hình ảnh

Figure 1.1: Spring Architechure	19
Figure 1.2: Dependency Injection	20
Figure 1.3: Cách thức hoạt động của Denpendency Injection	21
Figure 1.4: AOP	23
Figure 1.5: Các thuật ngữ trong AOP	24
Figure 1.6: Spring MVC	25
Figure 1.7: DispatcherServlet	26
Figure 1.8: Spring Security	28
Figure 1.9: Hibernate	29
Figure 1.10: Cấu trúc Hibernate	31
Figure 2.1: Hình User case admin.	43
Figure 2.2: Hình User case Guset.	44
Figure 2.3: Hình User case học sinh.	45
Figure 2.4: Hình Use case Teacher.	46
Figure 2.5: Senquence View Detail ConceptMap.	47
Figure 2.6: Senquence Upload Document.	48
Figure 2.7: Senquence Show List Task.	49
Figure 2.8: Senquence Show List Share.	50
Figure 2.9: Senquence Share ConceptMap.	51
Figure 2.10: Senquence Save ConceptMap.	52
Figure 2.11: Senquence Edit ConceptMap.	53
Figure 2.13: Senquence Make Task.	55
Figure 2.14: Senquence Chấm Điểm.	56
Figure 2.15: Sequene Add Friend.....	57
Figure 2.16: Sequence Download Document	58
Figure 2.17: Sequence Create Planner	59
Figure 2.18: Database.	60
Figure 2.19: Class Diagram Model	61
Figure 2.20: Class Diagram Task	62
Figure 2.21: Class Diagram File Upload	63
Figure 2.22: ClassDiagram Share	64
Figure 2.23: Giao diện đăng nhập	65
Figure 2.24: Giao diện thêm tài khoản.....	66
Figure 2.25: Giao diện Share ConceptMap	67
Figure 2.26: Giao diện Show List Task	68

Figure 2.27: Giao diện Show List Share.....	69
Figure 2.28: Giao diện Draw ConceptMap	70
Figure 2.29: Giao diện Lưu ConceptMap.....	71
Figure 2.30: Giao diện Make Task.....	73
Figure 3.1: Open Shift.....	75
Figure 3.2: Jboss	76
Figure 3.3: Khảo sát từ người dùng.....	89

Phần A: MỞ ĐẦU

Giới thiệu đề tài

Trước nay, chúng ta ghi chép thông tin bằng các ký tự, đường thẳng, con số. Với cách ghi chép này, chúng ta mới chỉ sử dụng một nửa của bộ não - não trái, mà chưa hề sử dụng kỹ năng nào bên não phải, nơi giúp chúng ta xử lý các thông tin về nhịp điệu, màu sắc, không gian và sự mơ mộng. Hay nói cách khác, chúng ta vẫn thường đang chỉ sử dụng 50% khả năng bộ não của chúng ta khi ghi nhận thông tin. Với mục tiêu giúp chúng ta sử dụng tối đa khả năng của bộ não, Tony Buzan đã đưa ra Bản đồ tư duy để giúp mọi người thực hiện được mục tiêu này.

Hiểu được ý muốn này mọi người trên mạng Internet hiện nay có rất nhiều trang web vẽ “Sơ đồ tư duy”. Ưu điểm của nó là giúp người ta nhìn ra vấn đề một cách toàn diện. Nhưng chúng ta lại tốn quá nhiều thời gian để vẽ vời.

Nhận thấy nhược điểm đó chúng em quyết định tạo ra một sản phẩm có thể vẽ được “Sơ đồ tư duy” mà tốn ít thời gian để vẽ hơn. Sau thời gian nghiên cứu tìm tòi, chúng em đã tạo ra một trang web có thể vẽ, lưu trữ và chia sẻ ConceptMap. So với “Sơ đồ tư duy” thì ConceptMap tạo ra các “khái niệm” và logic hơn. Giúp người dùng tiết kiệm được thời gian tạo ra nó hơn.

Mục tiêu của chúng em với đề tài này là xây dựng một website có đầy đủ tính năng vẽ, lưu trữ và chia sẻ ConceptMap. Đặc biệt chúng em còn có thể tạo ra một môi trường cho giáo viên có thể giao bài tập cho học sinh để học sinh có thể làm quen với “Sơ đồ ConceptMap” này hơn. Khi đã quen thuộc sử dụng vẽ ConceptMap, nó rất giúp ích cho chúng ta: sáng tạo hơn, tiết kiệm thời gian, ghi nhớ tốt hơn, nhìn thấy bức tranh tổng thể của một vấn đề, và tổ chức và phân loại suy nghĩ của bạn ...

Chương 1. Tĩnh cấp thiết

1.1 Khảo sát hiện trạng

Hiện nay có rất nhiều website chuyên về vẽ conceptmap. Trong chương này sẽ xây dựng bảng thống kê khảo sát về một số website nổi bật.

Table 1.1: Bảng khảo sát

STT	Tên Website	Nội dung đánh giá	Ưu điểm	Khuyết điểm
1	mindmup.com	Giao diện	Hiển thị đầy đủ các chức năng trên 1 trang.	Người mới sử dụng khó khăn trong việc bắt đầu vẽ
		Chức năng	Có thể thay đổi màu sắc các node Tự động chỉnh kích thước của node Tự động canh chỉnh vị trí của các node Cho phép chèn một node ở giữa 2 node khác một cách dễ dàng Kéo thả để thay đổi parent/child của các node	Không có các ghi chú cho từng chức năng

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

			<p>Có thể chèn hình ảnh làm nền cho 1 node</p> <p>Có thể đính kèm một tệp văn bản cho một node</p> <p>Lưu theo drive google, facebook, twitter...</p>	
2	examtime.com	Giao diện	- màu sắc sinh động	Không thể hiện các node trên giao diện
		Chức năng	<p>-Có thể thay đổi font chữ cho các node</p> <p>-Có thể thay đổi màu sắc các node bằng scroll bar</p> <p>-Thêm note mới dễ dàng bằng popup dưới node</p> <p>-kéo thả để tạo node mới</p> <p>-tự lưu khi trang page thoát.</p> <p>-Lưu theo drive google</p>	Nếu có quá nhiều node. Khung nhìn sẽ trở nên chật hẹp và khó nhìn
3	wisemapping.com	Giao diện	<p>-Cấu trúc sẵn các node</p> <p>-Có thể thêm các icon cho các node</p>	Nếu có quá nhiều node. Khung nhìn sẽ trở nên chật hẹp và khó nhìn

			-Tự động canh chỉnh vị trí của các node	
		Chức năng	<ul style="list-style-type: none"> - Cho phép chèn một node ở giữa 2 node khác một cách dễ dàng - Kéo thả để thay đổi parent/child của các node -Nhấn phím “enter” để thêm node -Lưu trữ theo drive google, facebook,... 	<ul style="list-style-type: none"> - Không có khả năng đồng bộ hóa dữ liệu. - Khi dùng phải mất nhiều thao tác.
4	text2mindmap.com	Giao diện	<p>Vui nhộn, dễ nhìn, dễ sử dụng</p> <ul style="list-style-type: none"> -Có hiệu ứng khi di chuyển các node 	- Thể hiện ít nút chức năng trên giao diện
		Chức năng	<ul style="list-style-type: none"> -Tự động chỉnh kích thước của node -Tự động canh chỉnh vị trí của các node Chỉnh sửa text của các node khó khăn -Lưu trữ theo gmail 	Không thể thay đổi hình dạng node

5	softschools.com	Giao diện		Giao diện xấu Hình dạng, màu sắc các node không đa dạng
		Chức năng		- Thao tác thêm node và xóa node không thuận tiện - Không thể đính kèm hình ảnh hay thay đổi hình nền của node - Chức năng khá ít

1.2 Mục đích nghiên cứu

Mục tiêu của chúng em với đề tài này là xây dựng một website có đầy đủ tính năng vẽ, lưu trữ và chia sẻ ConceptMap. Đặc biệt chúng em còn có thể tạo ra một môi trường cho giáo viên có thể giao bài tập cho học sinh để học sinh có thể làm quen với “Sơ đồ ConceptMap” này hơn.

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

1.3 Đối tượng và phạm vi nghiên cứu

- Các ứng dụng vẽ ConceptMap.
- Nghiên cứu lập trình Spring

1.4 Ý nghĩa khoa học và thực tiễn

- Có thể áp dụng vào thực tiễn.

Phần B: NỘI DUNG

Chương 1. Cơ sở lý thuyết và Công nghệ

1.1 Các công nghệ

1.1.1 Spring

Spring là một trong những framework của Java được sử dụng phổ biến nhất tính đến thời điểm này.

Như chúng ta đã biết, một ứng dụng thiết kế hướng đối tượng bao giờ cũng đặt việc tạo ra các thành phần riêng rẽ độc lập và có tính sử dụng lại cao lên làm ưu tiên hàng đầu. Bởi vì một khi đã tạo ra các thành phần như vậy, chúng ta sẽ tiết kiệm được rất nhiều thời gian và công sức trong quá trình tạo và bảo trì hệ thống sau này. Tuy nhiên, điều này quả là không dễ dàng bởi vì tính độc lập của các thành phần sẽ bị giảm đi do cách các thành phần liên kết với nhau. Với Spring, có được tiếng tăm như ngày hôm nay, chính là do framework này có cách tạo và liên kết các thành phần rất riêng và hữu ích.

Spring nổi bật ở cách nó liên kết các thành phần trong hệ thống lại với nhau. Sự liên kết được thực hiện bởi hai đặc tính cơ bản và quan trọng nhất của Spring là Dependency Injection và Aspect oriented programming (AOP)

1.1.2 Spring Architecture

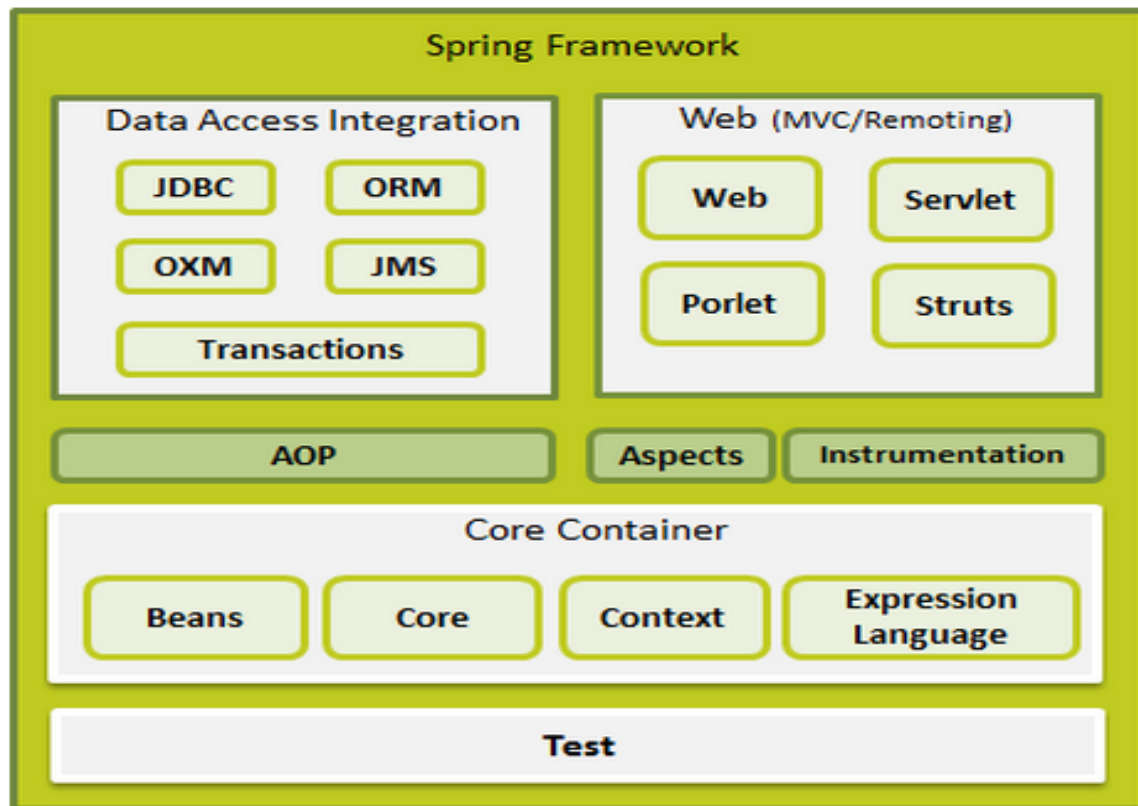


Figure 1.1: Spring Architecture

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

1.1.3 Dependency Injection

Đầu tiên ta hãy nói về việc nếu không sử dụng Dependency Injection.

Một class A phụ thuộc vào class B nếu class A sử dụng class B như là một biến.

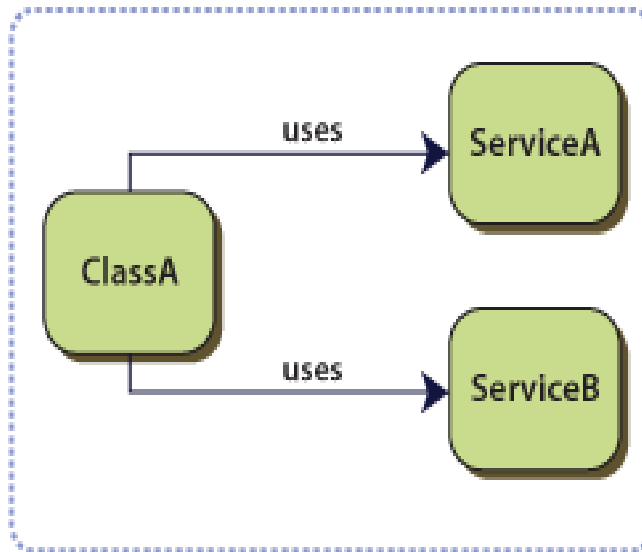


Figure 1.2: Dependency Injection

Ở đây ta thấy một số hạn chế. Thứ nhất là để thay thế hay cập nhật các lớp Service, ta cũng phải thay đổi code ở ClassA. Thứ hai là các lớp Service phải có sẵn lúc compile chương trình. Thứ ba là sẽ gây có khăn cho việc kiểm tra một các độc lập, vì có sự phụ thuộc trực tiếp giữa các lớp. Cuối cùng là sự lặp đi lặp lại của code để tạo, định vị và quản lý những sự phụ thuộc đó.

Dependency Injection

Khi viết 1 ứng dụng Java phức tạp, các class nên càng độc lập càng tốt, để làm tăng tính tái sử dụng và kiểm tra chúng 1 cách độc lập của các lớp khác trong khi làm đơn vị kiểm thử.

Dependency Injection giúp gắn các class với nhau và đồng thời giữ cho chúng độc lập

Nếu Dependency Injection được sử dụng ở lớp B được cấp cho lớp A thông qua: constructor của lớp A - thì đây được gọi là constructor injection

một setter - thì đây được gọi là setter injection

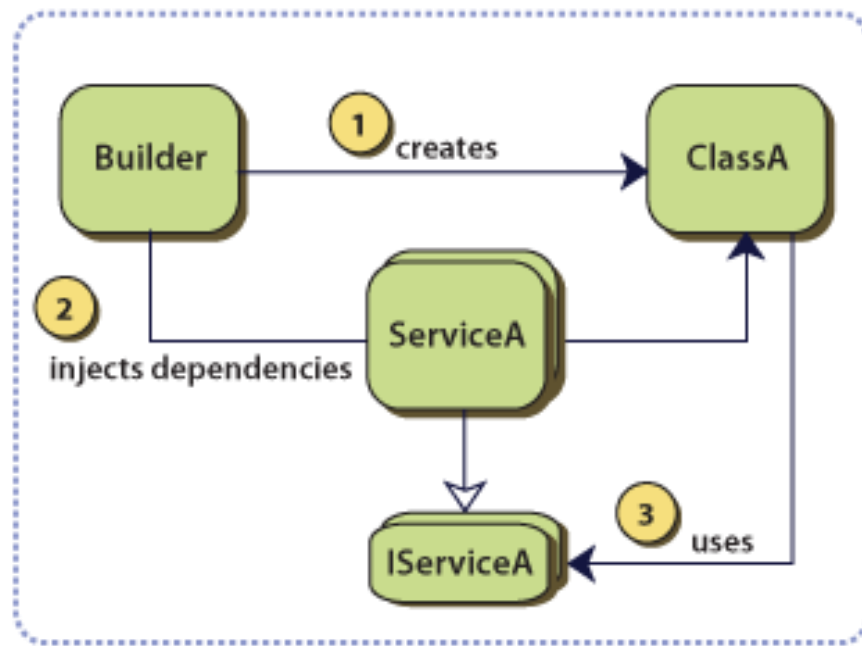


Figure 1.3: Cách thức hoạt động của Dependency Injection

Ở đây Builder chịu trách nhiệm bơm sự phụ thuộc cho ClassA. Không tồn tại sự phụ thuộc trực tiếp giữa ClassA và ServiceA. Và ta có thể dễ dàng thêm một Service mới bằng cách implement Interface IServiceA.

Ưu điểm:

Làm cho client độc lập hơn dễ dàng hơn để kiểm tra đơn vị.

Thúc đẩy tái sử dụng, kiểm thử và bảo trì.

Giảm mã boilerplate trong các đối tượng áp dụng từ tất cả các công việc để khởi tạo hoặc thiết lập phụ thuộc được xử lý bởi một thành phần cung cấp

Dependency injection cho phép phát triển đồng thời hoặc độc lập. Hai nhà phát triển độc lập có thể phát triển các lớp học sử dụng mỗi khác, trong khi chỉ cần phải biết giao diện các lớp sẽ giao tiếp thông qua. Plugins thường được phát triển bởi các cửa hàng bên thứ ba mà thậm chí không bao giờ nói chuyện với các nhà phát triển đã tạo ra các sản phẩm có sử dụng các plugin.

Nhược điểm:

Các nhà phát triển phải tham khảo nhiều tập tin theo cách một hệ thống thực hiện

Dependency injection thường đòi hỏi nhiều dòng code để thực hiện cùng một code thừa kế

Dependency injection tăng khối nói bằng cách yêu cầu người sử dụng một hệ thống con để cung cấp cho nhu cầu của các hệ thống con

1.1.4 Aspect oriented programming (AOP)

Aspect-Oriented Programming (AOP) còn được gọi là Aspect-Oriented Software Development (AOSD) là một nguyên tắc thiết kế giúp tách rời các yêu cầu hay các vấn đề được quan tâm (separation of concerns) trong chương trình thành các thành phần độc lập và từ đó tăng tính uyển chuyển cho chương trình. “Separation of concerns” là một trong những kỹ thuật được quan tâm nhất trong ngành phần mềm.

Khi viết phần mềm, ta thường gặp hai loại “concern” (khái niệm): core concerns và cross-cutting concerns:

+ Core concern: là requirement chính của chương trình, ví dụ như upload file/folder, đọc danh sách user, ...

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

+ Cross-cutting concerns: là những xử lý phụ cần được thực hiện khi “core concern” được gọi. Cross-cutting concerns thường xảy ra nhiều nơi trong chương trình, nó có thể xảy ra trong nhiều layer của ứng dụng, nhiều class, nhiều method; chẳng hạn như tính toán thời gian chạy các hàm đọc ghi database, ghi log lại mỗi lần cập nhật thông tin user, ...

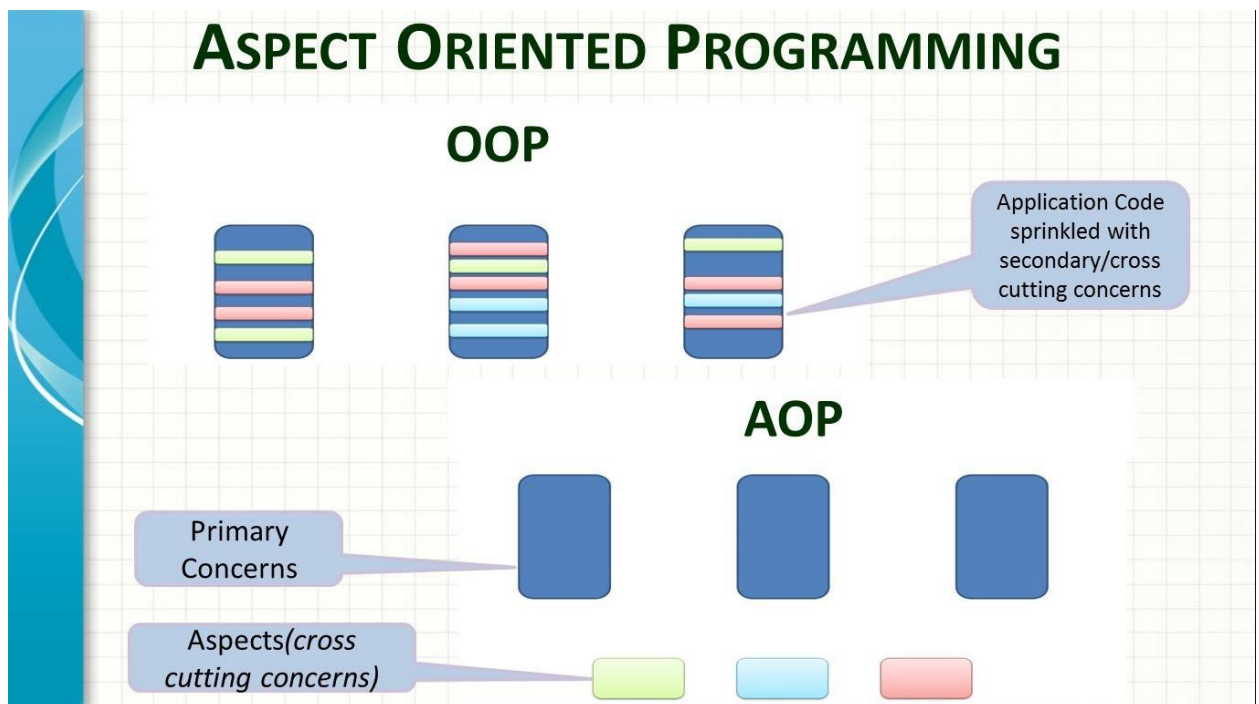


Figure 1.4: AOP

Kỹ thuật AOP thường được sử dụng để giải quyết các vấn đề như caching, tracking, security hay failure injections. Vì thế, nhiều tài liệu nói rằng AOP giúp module hóa ứng dụng, biến chương trình thành các module hoạt động độc lập, mỗi module làm một chức năng riêng, từ đó dễ bảo trì và nâng cấp.

AOP mang lại một số lợi ích

- + Chức năng chính của chương trình không cần biết đến các chức năng phụ khác
- + Các chức năng phụ có thể được thêm thắt, bật tắt lúc runtime tùy theo yêu cầu
- + Các thay đổi, sửa lỗi, nâng cấp nếu có đối với các chức năng phụ sẽ không ảnh hưởng đến chương trình chính.
- + Hệ thống sẽ uyển chuyển và giảm thiểu tính phụ thuộc lẫn nhau của các module

Các thuật ngữ thường gặp trong AOP

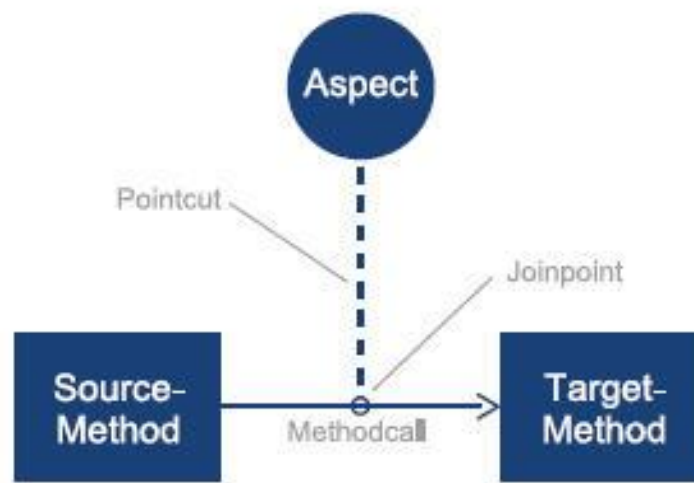


Figure 1.5: Các thuật ngữ trong AOP

+ Join point: Một điểm trong chương trình. Ví dụ như ta cần ghi log lại sau khi chạy method thì điểm ngay sau method đó được thực thi gọi là một joinpoint để có thể chạy những xử lý ghi file log. Join point theo nghĩa của nó có thể hiểu là những nơi có thể được chèn những “custom action” của bạn.

+ Pointcut: Có nhiều cách để xác định joinpoint, những cách như thế được gọi là

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

pointcut.

+ Advice: là những xử lý phụ được thêm vào xử lý chính. Code để thực hiện các xử lý đó được gọi là Advice. Vì các xử lý phụ có thể thêm vào trước hoặc sau hoặc cả hai đối với những xử lý chính nên có nhiều loại Advice khác nhau mà ta sẽ đề cập trong những phần sau.

1.1.5 Spring MVC

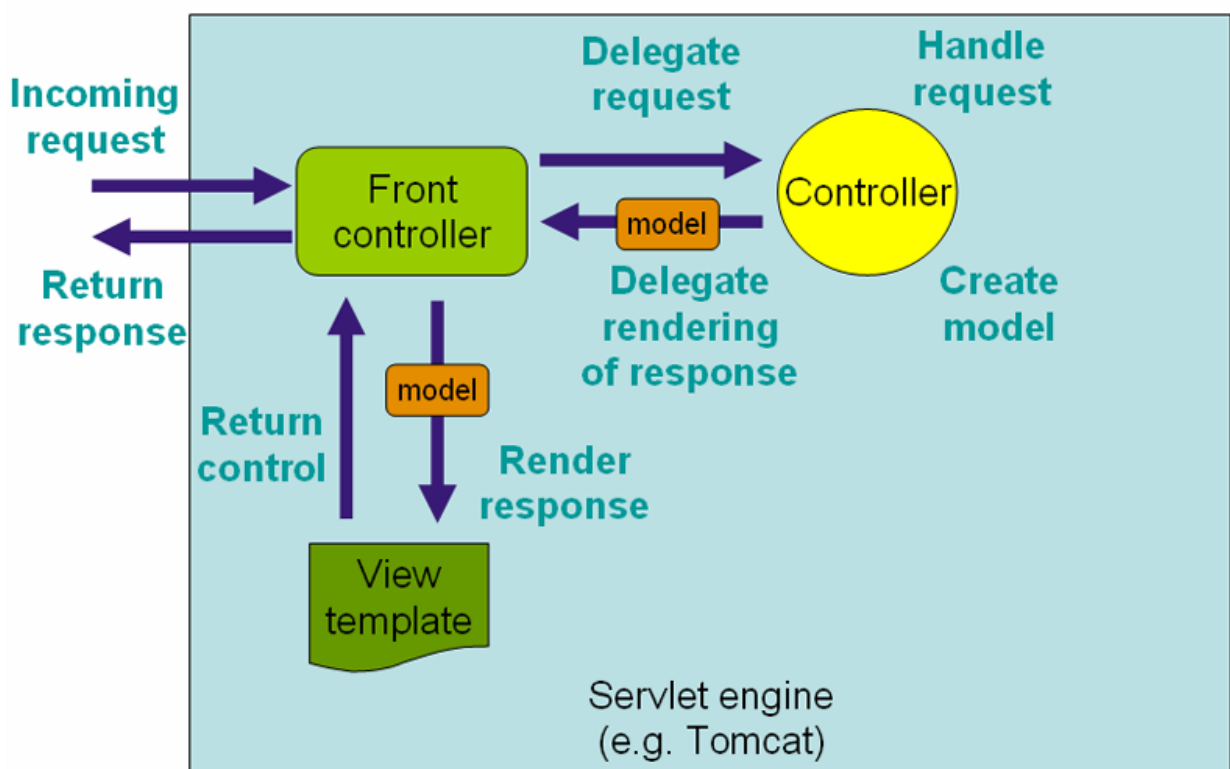


Figure 1.6: Spring MVC

Đầu tiên, có lẽ ta nói qua một chút về khái niệm MVC. MVC lần lượt là ba chữ cái đầu tiên của ba từ Model, View và Controller. MVC là một mô hình ứng dụng mà ở đó các thành phần được phân tách ra thành các lớp riêng biệt với các nhiệm vụ đặc trưng. View sẽ là lớp cho các thành phần có chức năng hiển thị, giao tiếp trực tiếp với người dùng.

Về, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

Nhiệm vụ của các thành phần trong View là trình bày các dữ liệu từ Model đến người dùng cuối. Model là các thành phần có khả năng lưu trữ và vận chuyển thông tin. Quá trình ném dữ liệu vào Model sẽ được thực hiện bởi Controller. Controller là các thành phần giúp cho việc xử lý logic các thao tác nghiệp vụ. Nhiệm vụ của Controller là lấy dữ liệu từ Model, xử lý dữ liệu, và cập nhật lại dữ liệu vào Model.

Spring MVC được xây dựng xung quanh mô hình DispatcherServlet

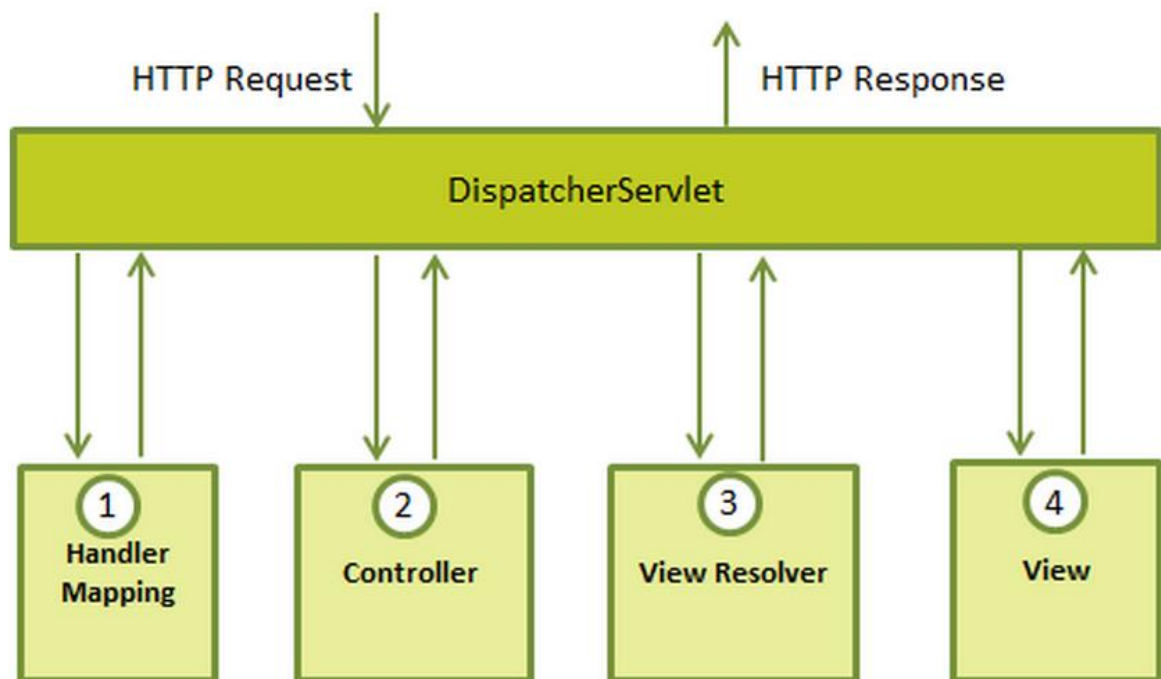


Figure 1.7: DispatcherServlet

DispatcherServlet được gọi là Front Controller, chịu trách nhiệm nhận các request sau đó điều hướng các request này và trả về các response.

Handler Mapping có nhiệm vụ kết nối một request với một Controller tương ứng.

Controller sau khi nhận request sẽ tiến hành các xử lý nghiệp vụ để tạo ra model sau đó return lại view name tương ứng để hiển thị phần model vừa được tạo.

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

DispatcherServlet sau khi nhận được view name sẽ tìm đến View Resolver để xác định view cụ thể sẽ trả về cho Client.

Spring Architecture

Ngày nay giao diện trang web thường được chia ra thành nhiều phần và các thành phần này có thể sử dụng lại giữa các trang web khác nhau. Ví dụ, một website chứa các thành phần : header, footer, menu, top... Các thành phần này cùng ý nghĩa khi ở nhiều trang web khác nhau. Rất khó khăn để làm cố định code của từng trang, và nếu sau này có 1 sự thay đổi nào thì tất cả các trang đều phải thay đổi. Do đó chúng ta sử dụng cơ chế templatzation, chúng ta tạo ra một Header, Footer, Menu chung và đưa chúng vào từng trang.

Apache Tiles cho phép tạo một mẫu (template) bao gồm các mảnh thành phần có thể thay thế được. Việc tạo ra một giao diện cụ thể giờ đây chỉ đơn giản là sử dụng lại mẫu có sẵn sau đó thay thế các thành phần mà ta muốn tùy biến.

Một cách bố trí giao diện phổ biến của các website được định nghĩa trong một file cấu hình trung tâm và giao diện này có thể được kế thừa trên tất cả các trang web của ứng dụng web.

1.1.6 Spring security

Nhắc đến từ Security, ta hiểu nó bao gồm rất nhiều thứ liên quan đến vấn đề bảo mật. Trong bảo mật, hai chủ đề hay được nhắc đến đó là cơ chế xác thực (Authentication) và cơ chế phân quyền (Authorization). Trong bản thân mỗi chủ đề này thì lại bao gồm rất nhiều cách thức và phương pháp khác nhau để thực hiện. Chẳng hạn như cơ chế xác thực, chúng ta có xác thực cơ bản (basic authentication), xác thực tập trung (central authentication), OpenID ...

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

Quan tâm đến vấn đề bảo mật, Spring đã phát triển một phần trong framework của mình để hỗ trợ cho việc thực hiện các cơ chế xác thực và phân quyền trên ứng dụng đó là Spring security.

Spring Security bắt đầu vào cuối 2003 với tên gọi “The Acegi Security System for Spring” bởi Ben Alex. Được công khai bởi Apache vào tháng 3 năm 2004.

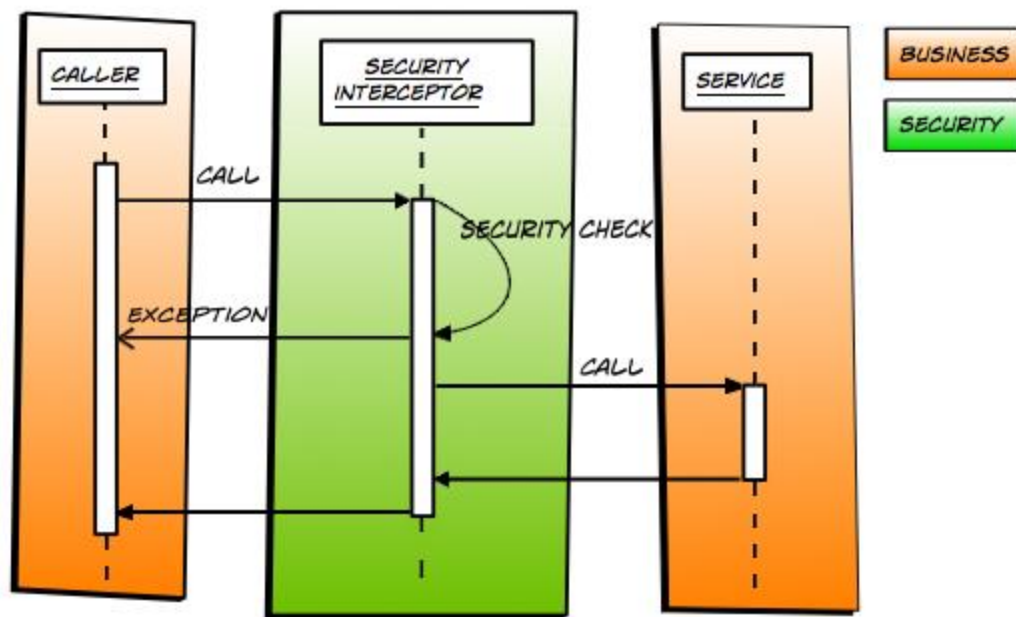


Figure 1.8: Spring Security

Spring Security hỗ trợ tối ưu cho người lập trình về bảo mật cũng như phân quyền. Chúng ta chỉ cần thực hiện một số cấu hình cần thiết thông qua bằng XML. Thậm chí Spring Security còn cung cấp sẵn một trang Login cho người sử dụng.

1.1.7 Hibernate

ORM (Object Relational Mapping) framework là một cơ chế cho phép người lập trình thao tác với database một cách hoàn toàn tự nhiên thông qua các đối tượng. Lập trình viên hoàn toàn không quan tâm đến loại database sử dụng, SQL...

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

Hibernate là một trong những ORM Framework. Hibernate framework là một framework cho persistence layer. Như vậy, nhờ có Hibernate framework mà giờ đây khi bạn phát triển ứng dụng bạn chỉ còn chú tâm vào những layer khác mà không phải bận tâm nhiều



Figure 1.9: Hibernate

về persistence layer nữa. Hibernate giúp lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh. Hibernate cho phép bạn truy vấn dữ liệu bằng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc bằng SQL thuần.

- Lợi ích khi sử dụng Hibernate

Trước tiên cần kể tới những khó khăn khi triển khai ứng dụng business dùng ngôn ngữ SQL thuần túy:

Mã thì rải rác khắp nơi đến những nơi không quản lý nổi.

Nếu thiết kế bị thay đổi sau khi phát triển ứng dụng, phải trả giá rất đắt để nhận ra những nơi cần phải thay đổi.

Khó tìm và fix bug.

Quản lý các kết nối database là một nhiệm vụ cực kỳ khó khăn vì mã SQL nằm tùm lùm, kết nối database cũng vậy.

Quản lý transaction là một nhiệm vụ phức tạp.

- Lợi ích khi sử dụng Hibernate (với HQL – Hibernate Query Language):

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

- *Tìm kiếm và sắp xếp nhanh.*
- *Làm việc được với dữ liệu lớn.*
- *Làm việc trên nhóm dữ liệu.*
- *Joining, aggregating.*
- *Chia sẻ nhiều người dùng và nhiều vùng.*
- *Giải quyết tương tranh (Transaction)*
- *Hỗ trợ cho nhiều ứng dụng.*
- *Bảo đảm toàn vẹn.*
- *Ràng buộc nhiều cấp độ.*

- **Tách biệt giao tác.**

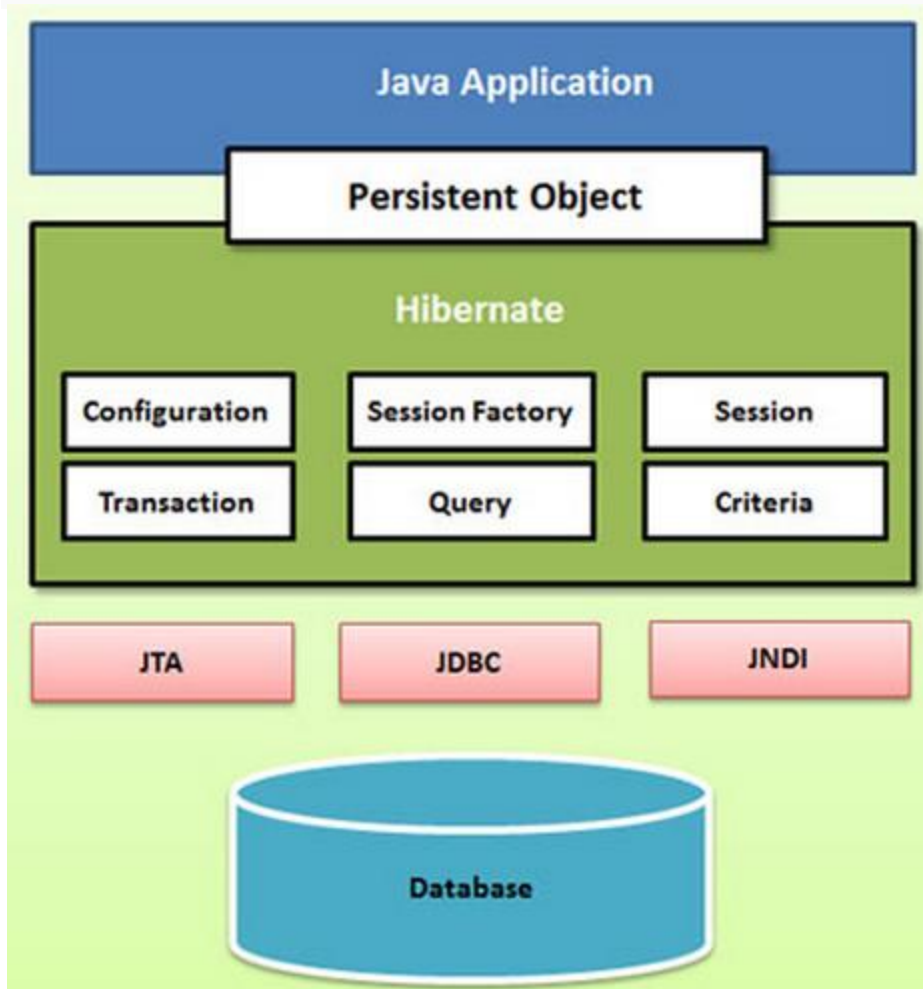


Figure 1.10: Cấu trúc Hibernate

Sơ đồ trên cho thấy rằng Hibernate là sử dụng cơ sở dữ liệu và file cấu hình để cung cấp các dịch vụ lâu dài cho ứng dụng.

Để sử dụng Hibernate, sự cần thiết phải tạo ra các lớp Java biểu diễn cho bảng tương ứng trong cơ sở dữ liệu và sau đó ánh xạ các thuộc tính của lớp với các cột của bảng trong cơ sở dữ liệu. Sau đó, Hibernate có thể được sử dụng để thực hiện các thao tác

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

trên cơ sở dữ liệu như chèn, chọn, cập nhật và xóa các bản ghi trong bảng. Hibernate tự động tạo ra các truy vấn để thực hiện các thao tác này.

Quản lý kết nối

Hibernate cung cấp dịch vụ quản lý kết nối cơ sở dữ liệu hiệu quả. Kết nối cơ sở dữ liệu là phần tốn kém tài nguyên và hiệu năng nhất trong việc tương tác với cơ sở dữ liệu vì nó đòi hỏi nhiều tài nguyên phải mở và đóng kết nối cơ sở dữ liệu.

Quản lý giao dịch

Quản lý dịch vụ giao dịch cung cấp khả năng người sử dụng để thực hiện nhiều hơn một câu lệnh tại một thời điểm.

Đối tượng bản đồ quan hệ

Đối tượng bản đồ quan hệ là kỹ thuật lập ánh xạ biểu diễn dữ liệu từ một mô hình đối tượng cho đến một mô hình dữ liệu quan hệ. Đối tượng này được sử dụng để thực hiện các thao tác như truy vấn, thêm mới, cập nhật và xóa các bản ghi dạng bảng bên dưới. Hibernate là rất tốt công cụ như xa như là đối tượng bản đồ quan hệ là mối quan tâm, nhưng về mặt quản lý kết nối và quản lý giao dịch, đó là thiếu hiệu năng và khả năng. Vì vậy, thường Hibernate được sử dụng để quản lý kết nối khác và các công cụ quản lý giao dịch. Ví dụ apache DBCP được sử dụng để kết nối tổng hợp với Hibernate.

1.1.8 Json (JavaScript Object Notation.)

Nó là một chuẩn để định dạng dữ liệu, về mặt này, có thể so sánh JSON với XML, YAML... Nhưng khi JSON đi với JavaScript hoặc ActionScript thì nó có tính ưu việt hơn hẳn

Lợi ích của JSON

Khi sử dụng JSON với JavaScript hay ActionScript, không cần phải có các bước phân tích phức tạp như đối với XML. Mà có thể truy vấn trực tiếp giá trị theo tên (khóa) được định nghĩa trong JSON.

Nên sử dụng JSON trong những tình huống nào

1. Lưu trữ dữ liệu đơn thuần. Đó là khi bạn muốn lưu trữ dữ liệu dưới dạng metadata ở phía server. Chuỗi JSON sẽ được lưu vào database và sau đó khi cần dữ liệu thì sẽ được giải mã. Ví dụ với PHP, cung cấp các hàm liên quan đến JSON để mã và giải mã là `json_encode` và `json_decode`.
 - Chú ý: phương pháp này cũng tương tự như sử dụng tính năng `serialize` và `unserialize` của PHP. Nhưng trong khi `serialize` và `unserialize` sử dụng với cả dữ liệu và biến, tức là phụ thuộc vào ngôn ngữ lập trình là PHP và dĩ nhiên không thể transfer sang ngôn ngữ lập trình khác để `unserialize` được. Vì vậy, nếu dữ liệu của bạn chỉ đơn thuần là dữ liệu cơ bản (chuỗi kí tự, số...) thì bạn hoàn toàn không nên sử dụng `serialize` mà nên sử dụng JSON.
2. Sử dụng JavaScript, ActionScript để xử lý thông tin trả về từ phía server. Rất nhanh và rất dễ dàng.

1.1.9 SOA (Service - Oriented Architecture)

SOA là một kiểu kiến trúc hỗ trợ cho hướng dịch vụ.

Service-oriented architecture (SOA) là một phương pháp được sử dụng để tạo ra một kiến trúc dựa trên việc sử dụng các dịch vụ. Dịch vụ (như dịch vụ web RESTful) thực hiện một số chức năng nhỏ, chẳng hạn như sản xuất dữ liệu, xác nhận khách hàng, hoặc cung cấp các dịch vụ phân tích đơn giản.

Lợi ích của Service-Oriented Architecture (SOA):

1. Platform Độc lập - Kể từ khi dịch vụ web có thể được công bố và tiêu thụ trên nền tảng phát triển và hoạt động, doanh nghiệp có thể tận dụng các ứng dụng di sản hiện có của nó mà cư trú trên các loại khác nhau của máy chủ và xây dựng thêm các chức năng mà không cần phải xây dựng lại toàn bộ điều. Nó cũng giúp một doanh nghiệp để tích hợp các ứng dụng của mình với những đối tác của mình.

2. Vai trò Developer Focused - Từ một dịch vụ là độc lập thực hiện rời rạc của các dịch vụ khác, các nhà phát triển phụ trách của một dịch vụ có thể tập trung hoàn toàn vào việc thực hiện và duy trì các dịch vụ mà không cần phải lo lắng về các dịch vụ khác miễn là hợp đồng xác định trước được vinh danh.

3. Địa điểm Transparency - dịch vụ Web thường được xuất bản vào một thư mục nơi mà người tiêu dùng có thể tìm họ. Ưu điểm của phương pháp này là các dịch vụ web có thể thay đổi vị trí của mình bất cứ lúc nào. Người tiêu dùng của dịch vụ sẽ có thể xác định vị trí các dịch vụ thông qua các thư mục.

4. Mã Tái sử dụng - Từ SOA phá vỡ một ứng dụng thành từng miếng nhỏ và độc lập của chức năng, các dịch vụ có thể được tái sử dụng trong nhiều ứng dụng, do đó việc giảm chi phí phát triển.

5. Greater Testability - , dịch vụ độc lập nhỏ là dễ dàng hơn để kiểm tra và sửa lỗi hơn các ứng dụng nguyên khối. Điều này dẫn đến phần mềm đáng tin cậy hơn.

6. Phát triển song song - Kể từ khi dịch vụ này là độc lập với nhau và hợp đồng giữa các dịch vụ được xác định trước, các dịch vụ có thể được phát triển song song - điều này sẽ rút ngắn vòng đời phát triển phần mềm đáng kể.

7. Khả năng mở rộng tốt hơn - Từ vị trí của một dịch vụ không quan trọng nữa, các dịch vụ có thể được minh bạch chuyển đến một máy chủ mạnh mẽ hơn để phục vụ người tiêu dùng nhiều hơn nếu cần thiết. Ngoài ra, có thể có nhiều trường hợp của các dịch vụ đang chạy trên các máy chủ khác nhau. Điều này làm tăng khả năng mở rộng.

8. Tính sẵn có cao hơn - Từ vị trí của một dịch vụ không quan trọng và bạn có thể có nhiều trường hợp của một dịch vụ, có thể để đảm bảo tính sẵn sàng cao.

1.1.10 Amazon S3 (Simple Store Service)

Amazon Simple Storage Service là một nơi lưu trữ dữ liệu trên internet. Nó được thiết kế để làm cho quy mô trang web dễ dàng hơn cho nhà phát triển.

Amazon S3 có một giao diện dịch vụ web đơn giản là bạn có thể sử dụng để lưu trữ và lấy lại một khối lượng lớn dữ liệu ở mọi lúc, mọi nơi trên web.

AWS Account Access Keys

Tài khoản access keys cung cấp quyền truy cập đầy đủ đến tài nguyên AWS thuộc sở hữu của người dùng. Sau đây là ví dụ của access keys:

- Access key ID (20-ký tự, chuỗi gồm chữ và số). VD:
AKIAIOSFODNN7EXAMPLE
- Mật khẩu access key (một chuỗi 40 ký tự và số). VD:
wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

The access key ID xác định duy nhất một tài khoản AWS. Bạn có thể sử dụng nhiều access keys để gửi xác nhận yêu cầu từ Amazon S3.

Buckets

Bucket là một container cho các đối tượng được lưu trữ trong Amazon S3. Ví dụ, nếu đối tượng tên: photos/puppy.jpg được lưu trữ trong johnsmith bucket, thì có URL: <http://johnsmith.s3.amazonaws.com/photos/puppy.jpg>

Amazon S3 Objects

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

Amazon S3 có key, value store được thiết kế để lưu trữ nhiều đối tượng như ý bạn. Bạn lưu trữ nhiều đối tượng trong 1 hoặc nhiều Buckets. Một đối tượng gồm:

- Key – Tên bạn đặt cho đối tượng. Bạn sử dụng Key để truy xuất đối tượng.
- Version ID – với một bucket, một key and version ID xác nhận duy nhất một object.

The version ID là một chuỗi do Amazon S3 tạo ra khi bạn thêm 1 đối tượng vào một bucket.

- Value – Nội dung bạn lưu trữ.

Một object value có thể là bất kỳ dãy bytes. Objects có thể khoảng trong từ 0 - 5 TB.

- Metadata – Một tập hợp các cặp name – value giá trị với cái bạn lưu trữ thông tin liên quan đến các objects.

Bạn có thể chỉ định metadata, được gọi như xác định user, objects của bạn trong Amazon S3. Amazon S3 cũng chỉ định hệ thống metadata đến những objects – nó được dùng để quản lý các đối tượng.

- Subresources – Amazon S3 uses the subresource mechanism to store object-specific additional information.

Because subresources are subordinates to objects, they are always associated with some other entity such as an object or a bucket.

- Access Control Information – Bạn có thể kiểm soát quyền truy cập vào các đối tượng bạn lưu trữ trong Amazon S3.

Object Metadata

Có 2 loại metadata: system metadata and user-defined metadata.

System-Defined Metadata

Cho mỗi object được lưu trữ trong một bucket, Amazon S3 duy trì một tập hợp các hệ thống metadata. Amazon S3 xử lý hệ thống này khi cần. Ví dụ, Amazon S3 duy trì cho object tạo ngày và kích thước của metadata và sử dụng thông tin đó như một phần của quản lý object.

Có 2 loại hệ thống metadata:

- Metadata như object tạo ngày.
- Hệ thống metadata khác như: the storage class cấu hình cho object và cho dù object mã hóa phía máy chủ kích hoạt là những ví dụ của hệ thống metadata mà giá trị của bạn kiểm.

User-Defined Metadata

Khi tải một object, bạn có thể chỉ định metadata đến object. Bạn cung cấp thông tin tùy chọn này như cặp name-value khi bạn gửi một yêu cầu PUT hoặc POST để tạo object.

Object Versioning

Versioning cho phép bạn giữ nhiều phiên bản của 1 object trong một bucket, ví dụ: my-image.jpg (version 111111) and my-image.jpg (version 222222).

1.1.11 Spring Social

Các dự án Spring Social cho phép các ứng dụng của bạn để thiết lập kết nối với Software-as-a-Service (SaaS) nhà cung cấp như Facebook và Twitter để gọi API thay mặt cho người sử dụng.

Với Spring Social, ứng dụng của bạn có thể chạy một phần của người tiêu dùng dịch vụ, tương tác với một nhà cung cấp dịch vụ thay mặt người sử dụng. Các tính năng chính của Spring Social là:

- "Connect Framework" để xử lý ủy quyền cốt lõi và dòng kết nối với các nhà cung cấp dịch vụ.
- "Connect Controller" mà xử lý việc trao đổi OAuth giữa một nhà cung cấp dịch vụ, người tiêu dùng, và người sử dụng trong một môi trường ứng dụng web.
- "Romantic Controller" cho phép người dùng xác thực với ứng dụng của bạn bằng cách đăng nhập với tài khoản nhà cung cấp của họ, chẳng hạn như Twitter hay Facebook tài khoản của họ.

Provider Accounts

Để provider sign in tương thích với Spring application, thì cần cấu hình ProviderSignInController như sau:

```
@Bean public ProviderSignInController providerSignInController() {
    return new
    ProviderSignInController(connectionFactoryLocator(),usersConnectionRepository(),
    new SimpleSignInAdapter());
}
```

Về, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

1.2 Thư Viện

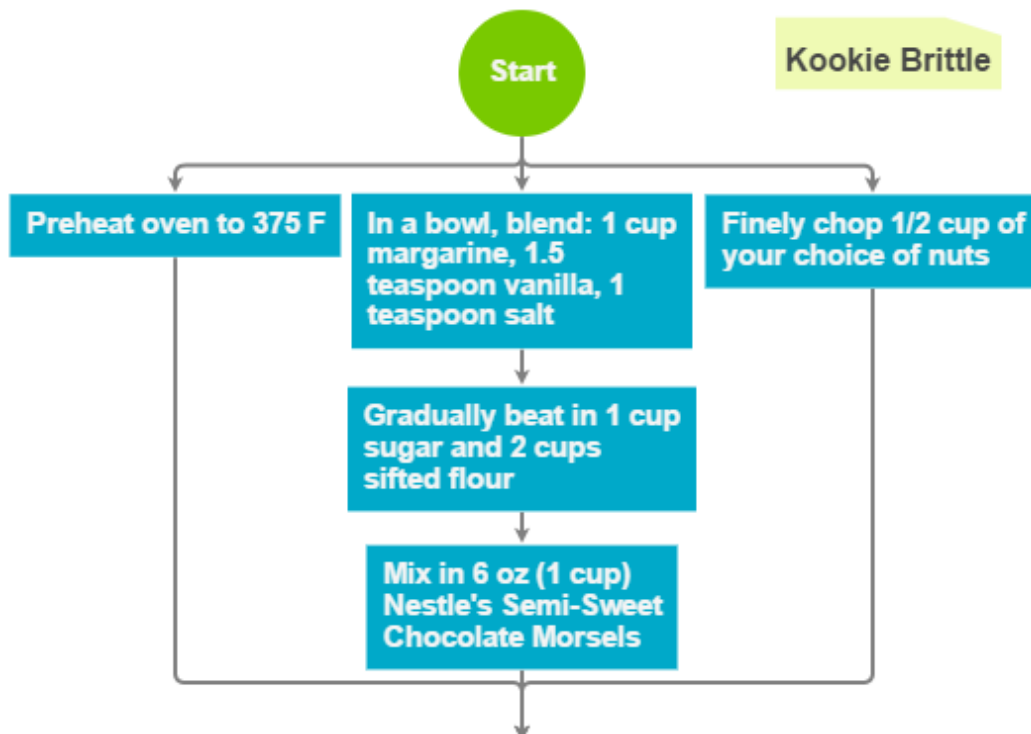
1.2.1 Thư viện vẽ *conceptmap* GoJS

GoJS là thư viện JavaScript hỗ trợ vẽ các sơ đồ tương tác như Conceptmap, State Chart, MindMap, Flowchart,...

GoJS sử dụng định dạng JSON để lưu trữ cũng như nạp lại Diagram nên rất dễ để sử dụng.

Ngoài ra GoJS còn hỗ trợ xuất Diagram thành dạng hình ảnh hoặc SVG (Scalable Vector Graphics – định nghĩa đồ họa trên định dạng XML).

Vì GoJS là thư viện JavaScript nên nó cũng phụ thuộc vào các tính năng của HTML5.



Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework



GoJS hỗ trợ tất cả các trình duyệt hiện đại (IE9+), bao gồm cả trình duyệt trên di động.



Chương 2. Thiết kế.

2.1 Kiến trúc hệ thống.

2.1.1 Chức năng chính.

- Vẽ Concpet Map.
- Lưu trữ Concept Map.
- Chia sẻ Concept Map.

2.1.2 Webservice.

- Amazon S3.
- Social facebook.

2.2 Thiết kế sơ đồ User case.

2.2.1 Xây dựng User Case Admin.

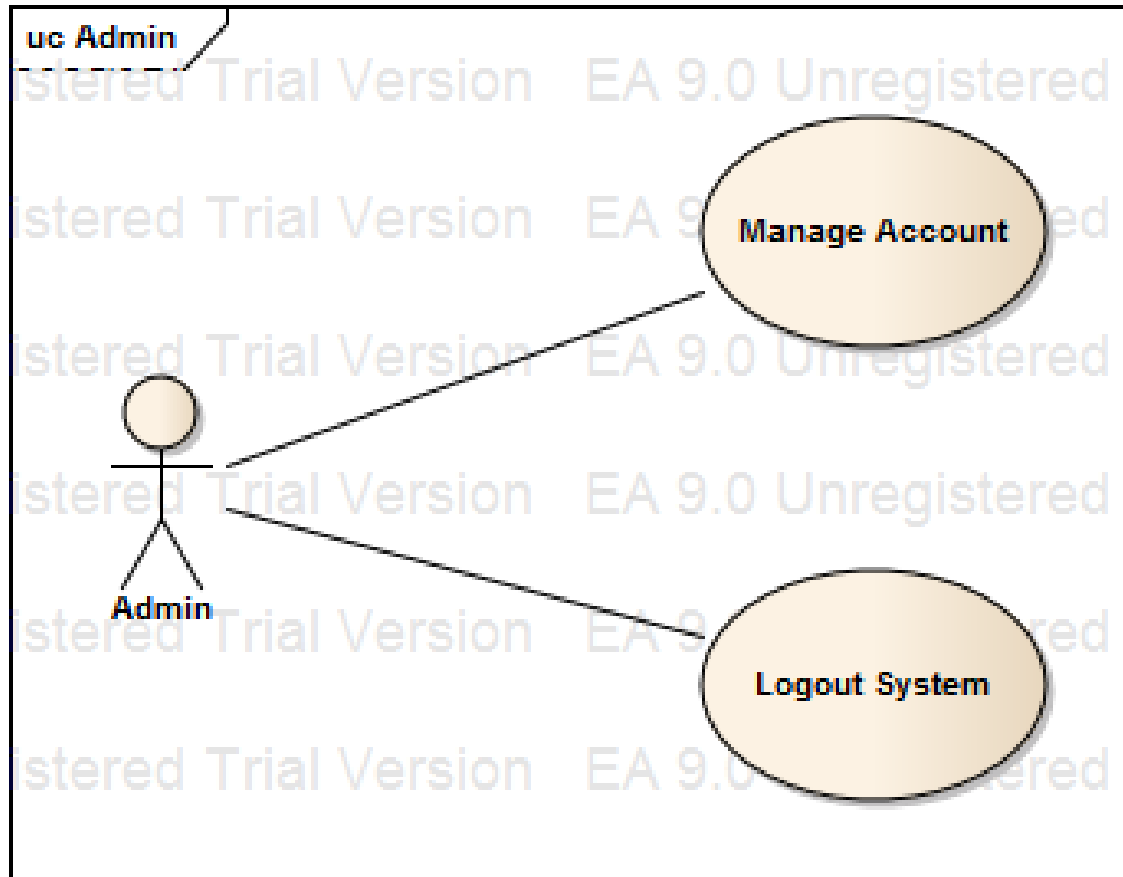


Figure 2.1: Hình User case admin.

2.2.2 Xây dựng User Case Khách.

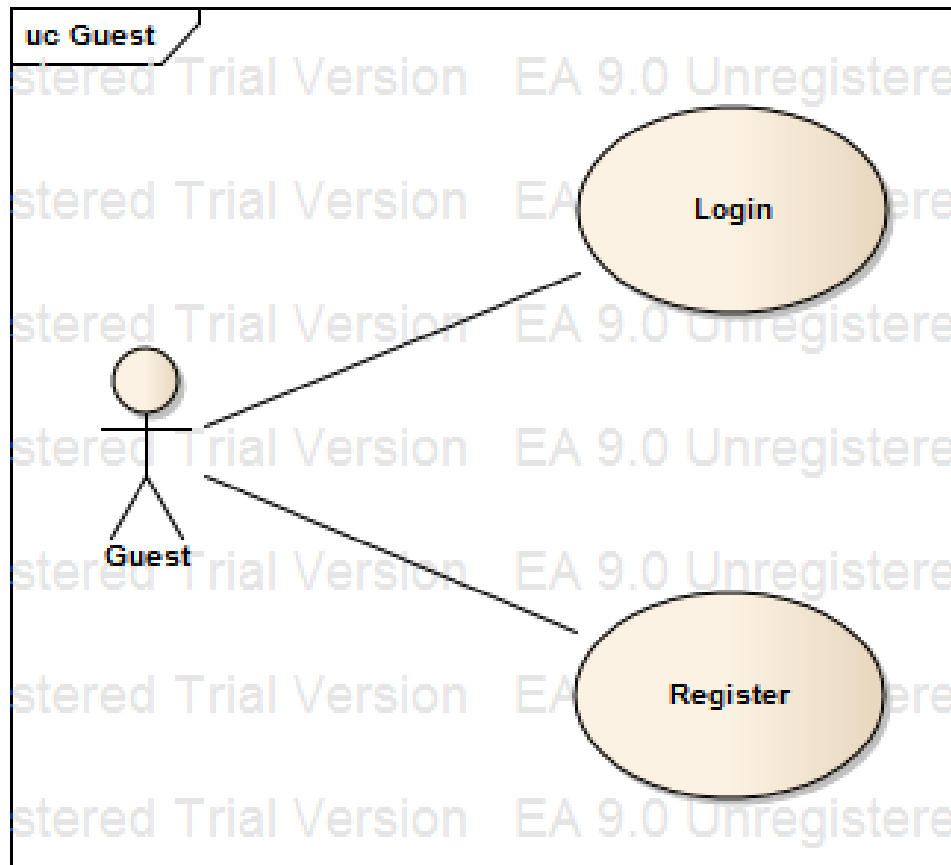


Figure 2.2: Hình User case Guset.

2.2.3 Xây Dựng User case Học sinh.

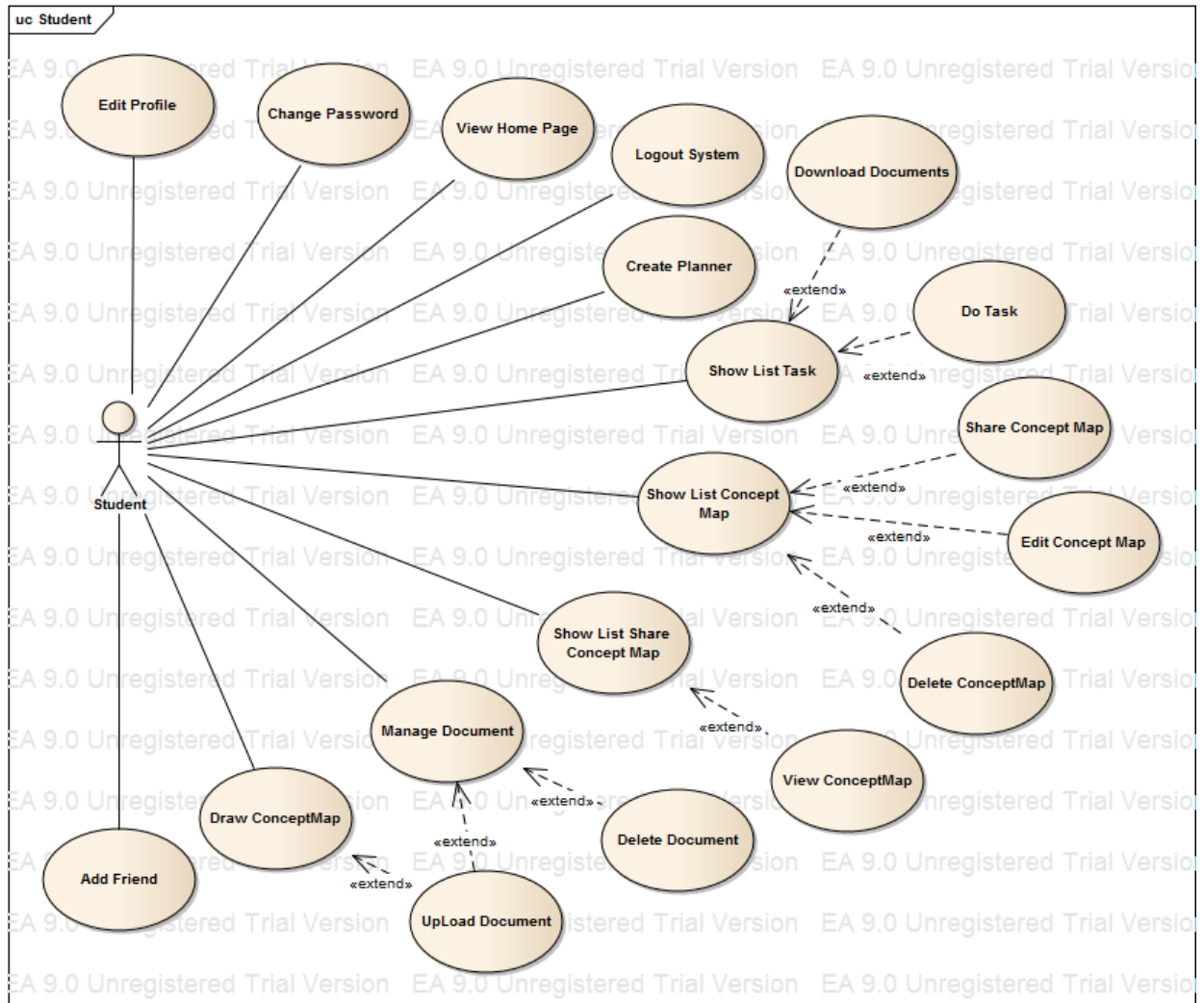


Figure 2.3: Hình User case học sinh.

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

2.2.4 Xây dựng User Giáo Viên.

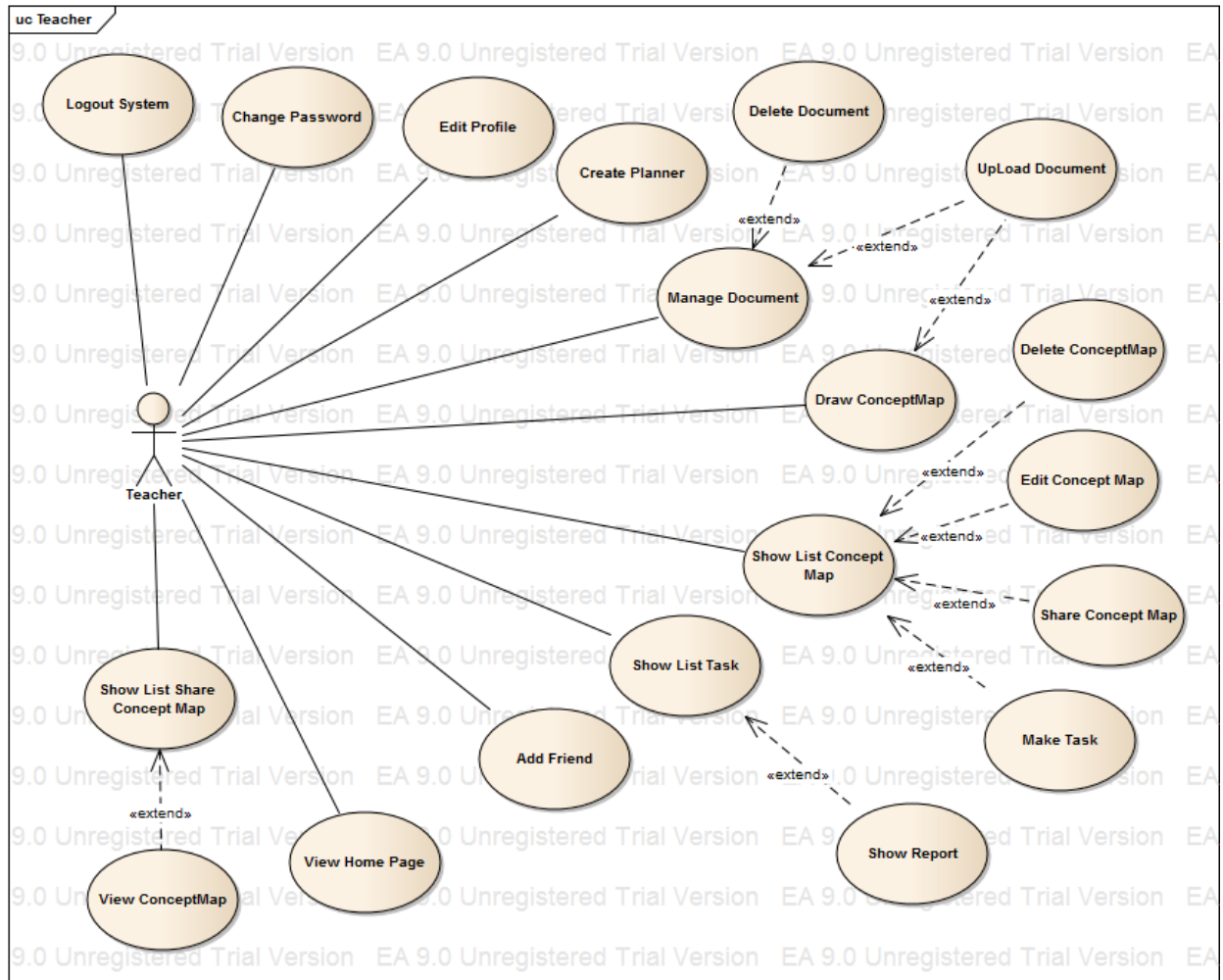


Figure 2.4: Hình Use case Teacher.

2.3 Thiết kế Sequence Diagram.

2.3.1 View Detail ConceptMap.

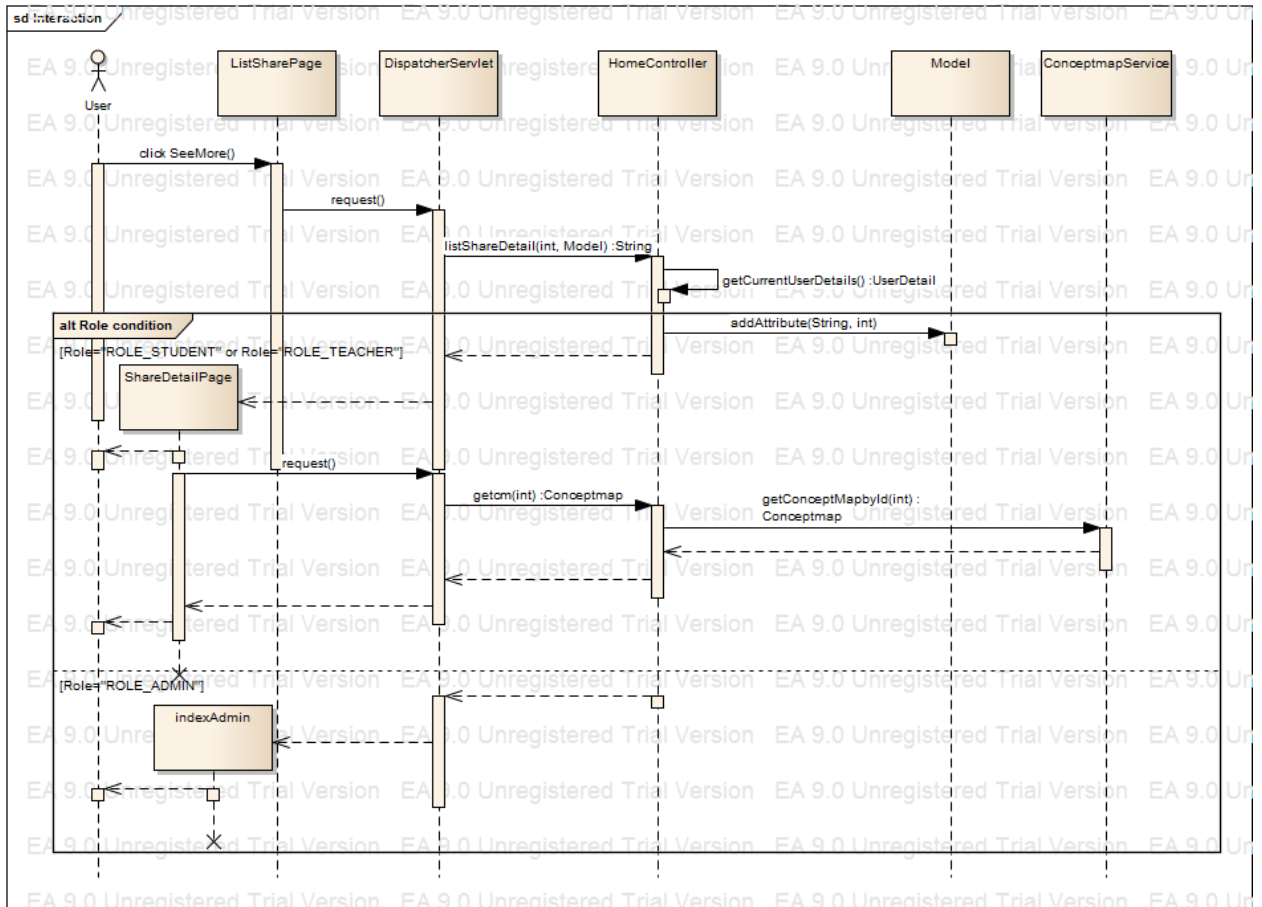


Figure 2.5: Sequence View Detail ConceptMap.

2.3.2 Upload Document.

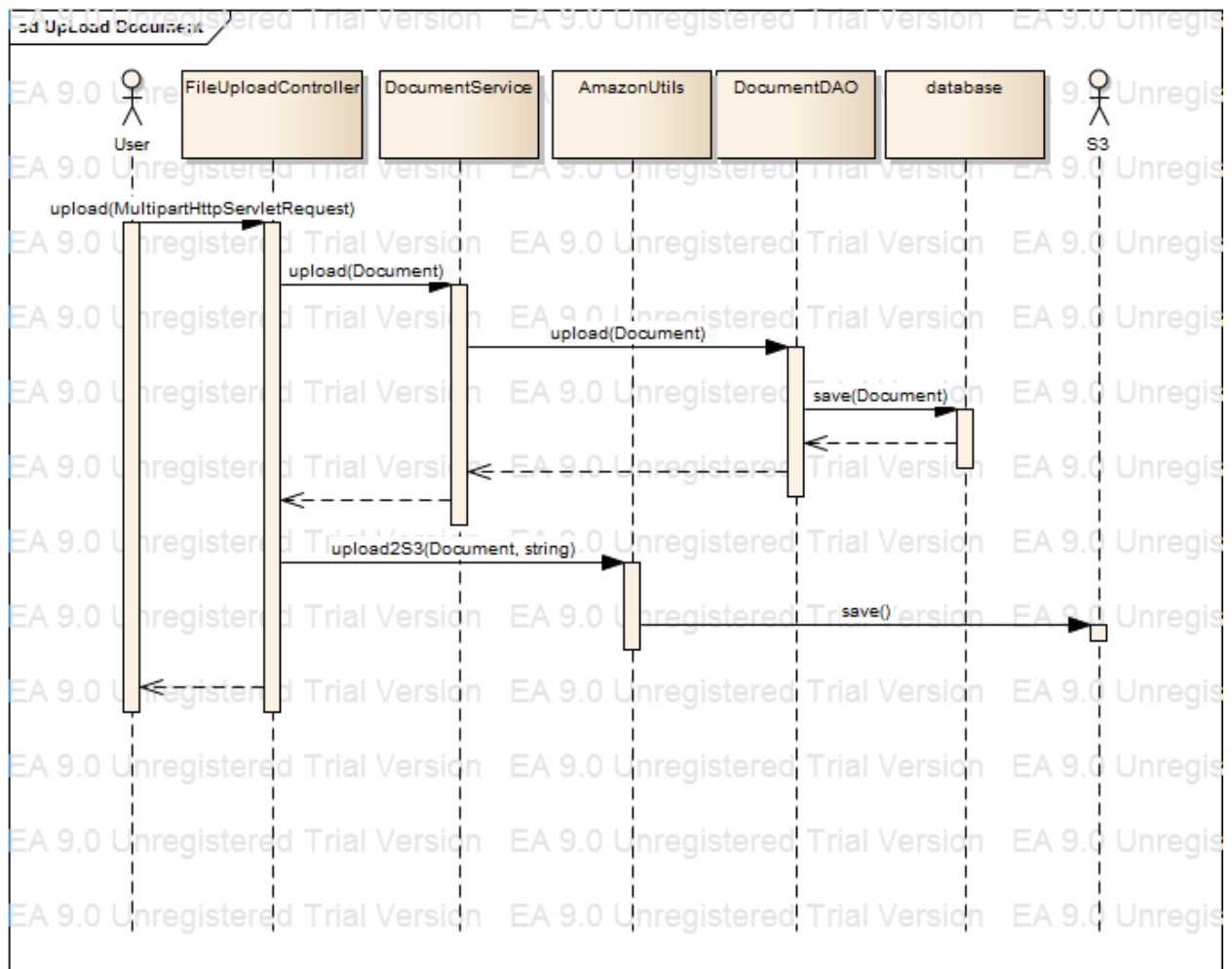


Figure 2.6: Sequence Upload Document.

2.3.3 Show List Task.

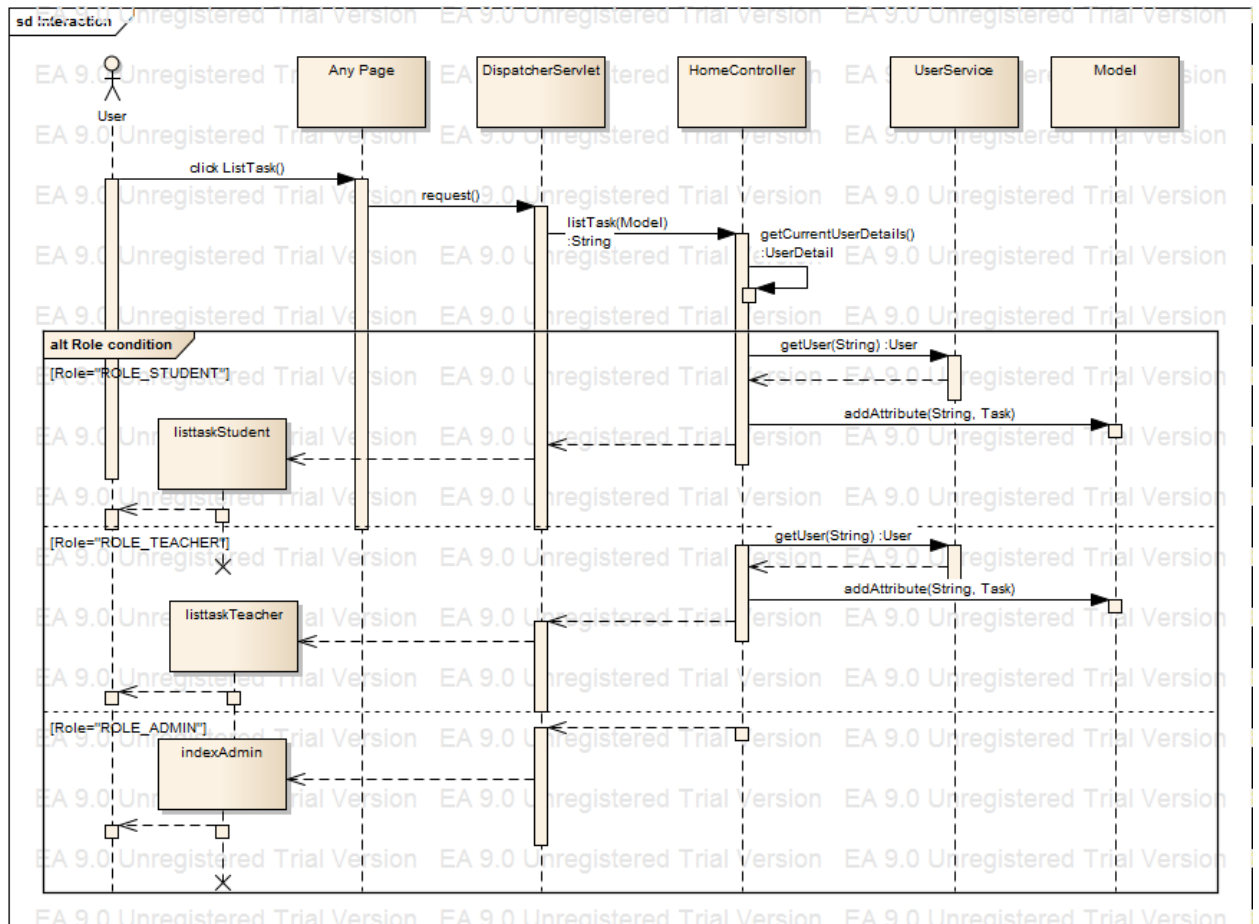


Figure 2.7: Sequence Show List Task.

2.3.4 Show List Share.

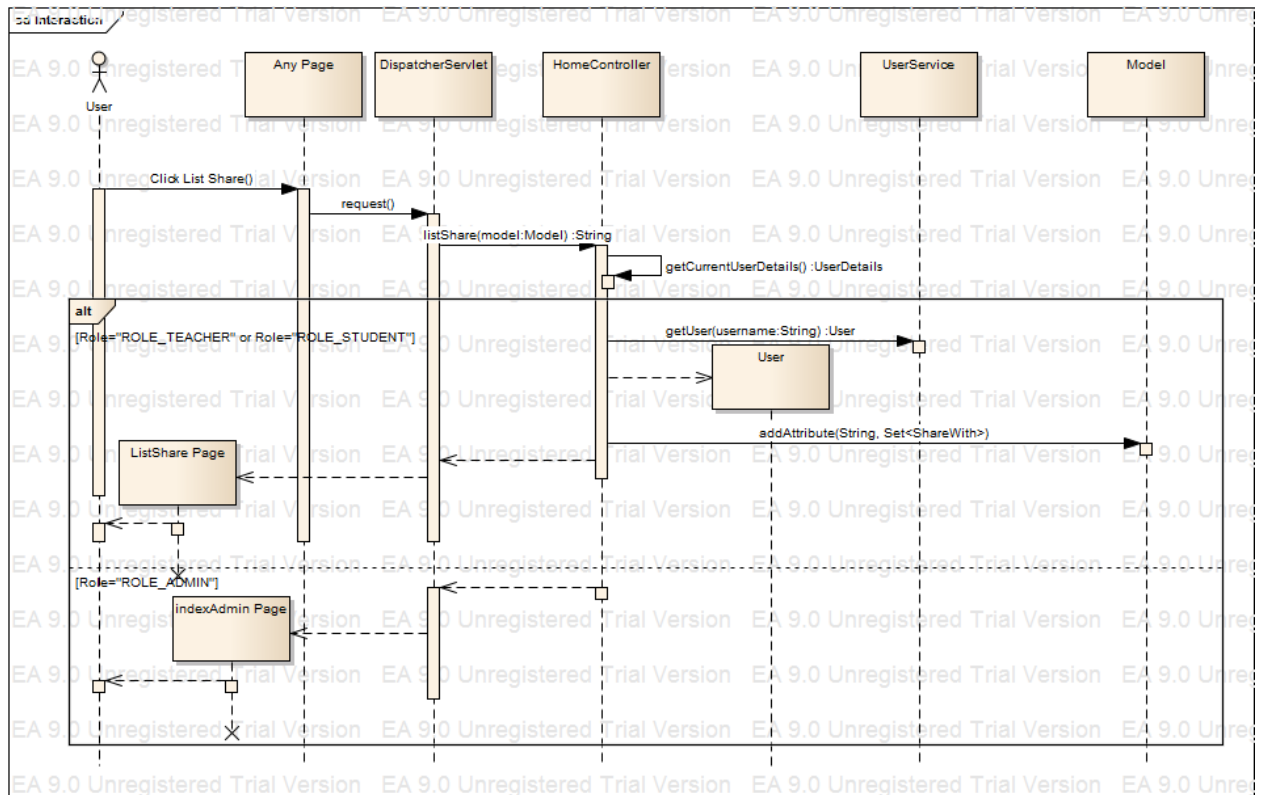


Figure 2.8: Sequence Show List Share.

2.3.5 Share ConceptMap.

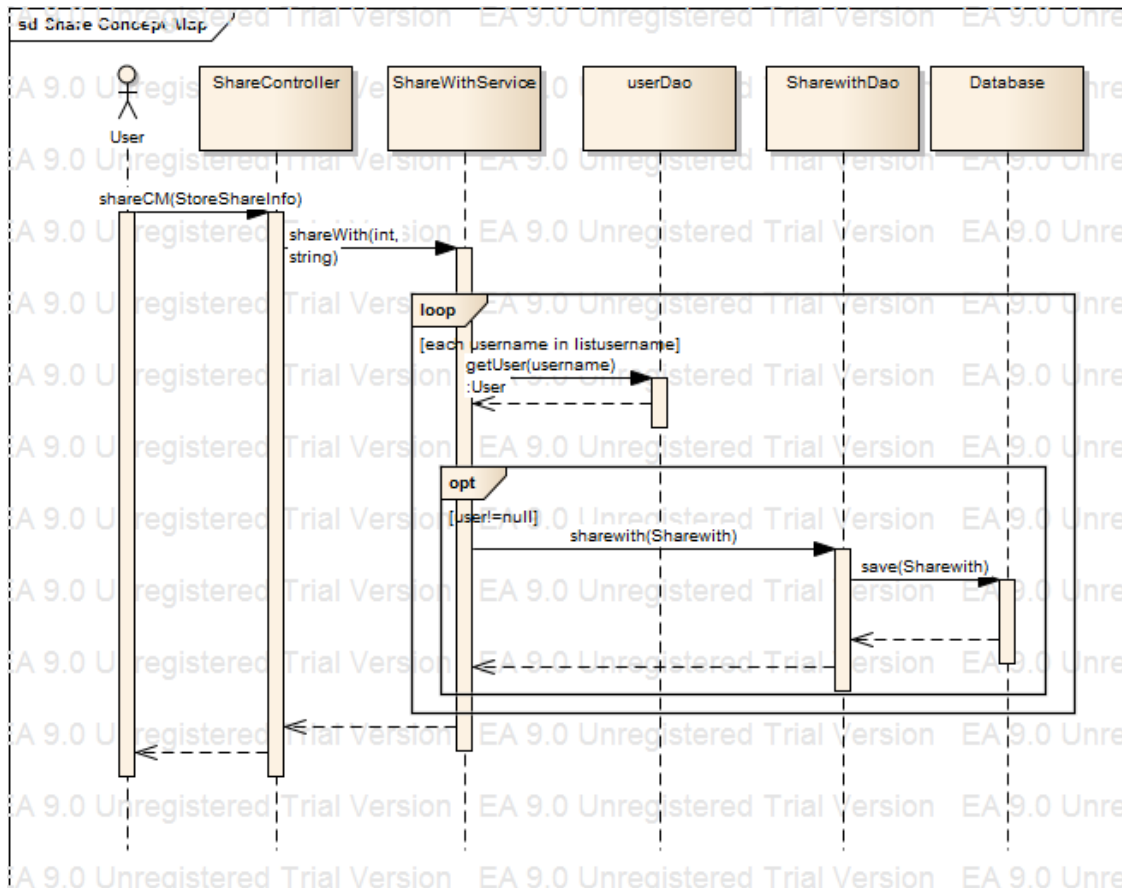


Figure 2.9: Sequence Share ConceptMap.

2.3.6 Save ConceptMap.

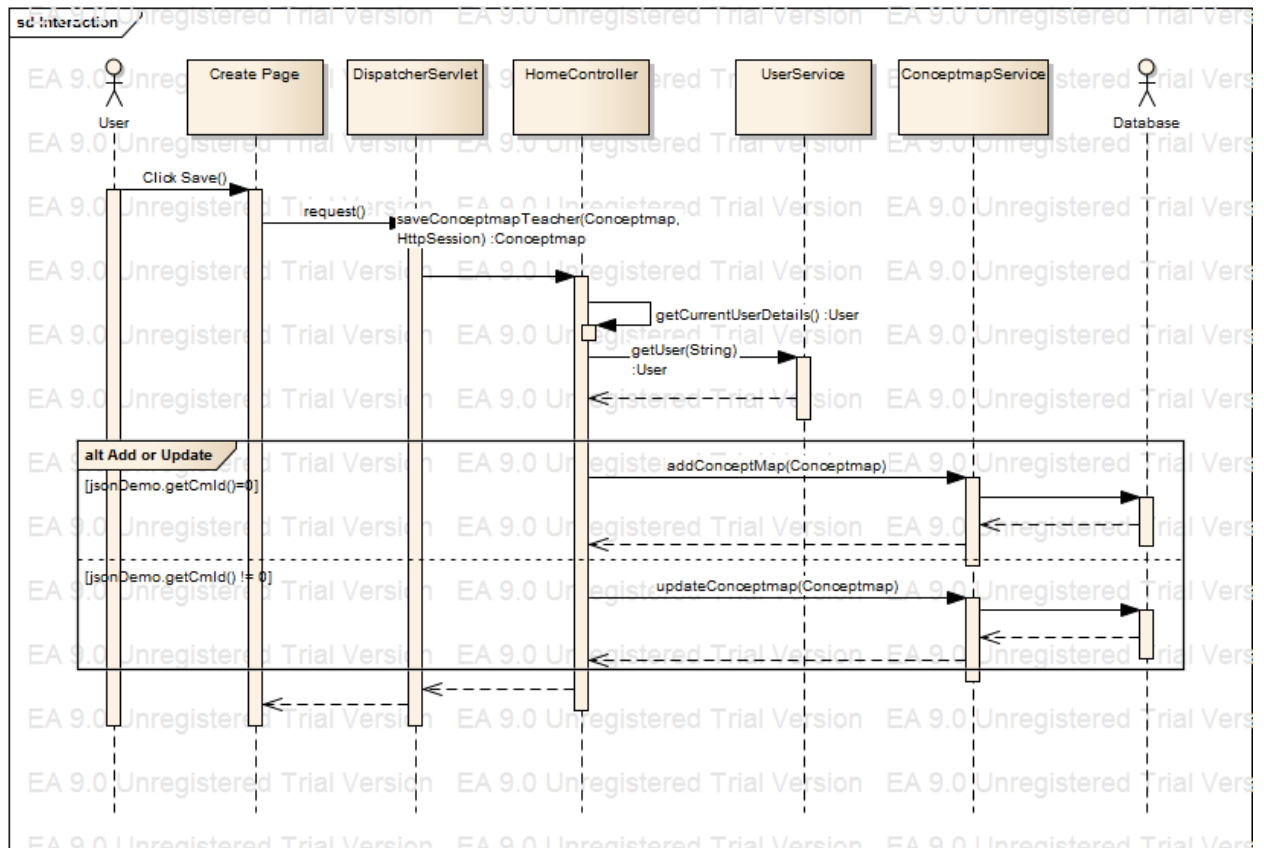


Figure 2.10: Sequence Save ConceptMap.

2.3.7 Edit ConceptMap.

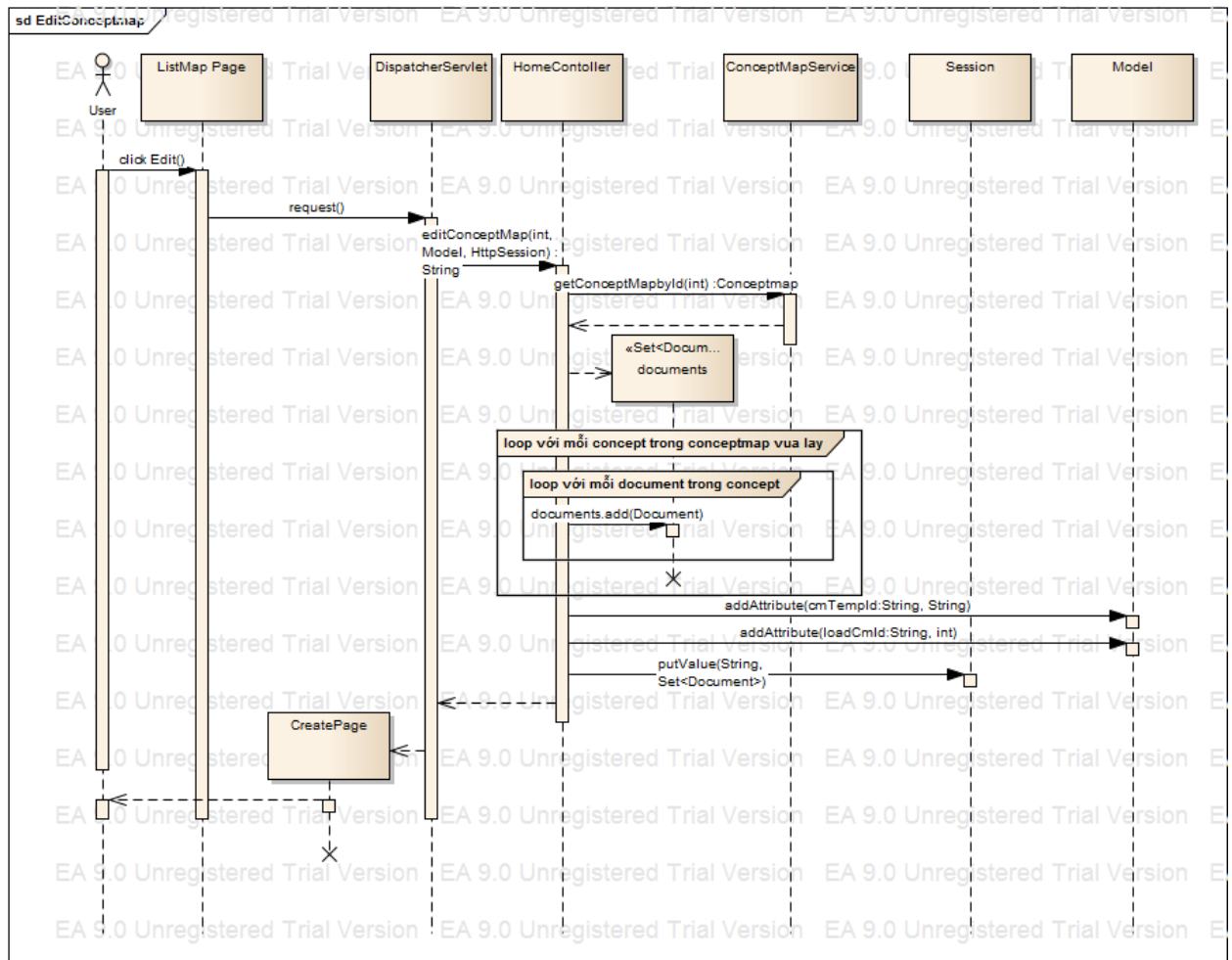


Figure 2.11: Sequence Edit ConceptMap.

2.3.8 Delete ConceptMap.

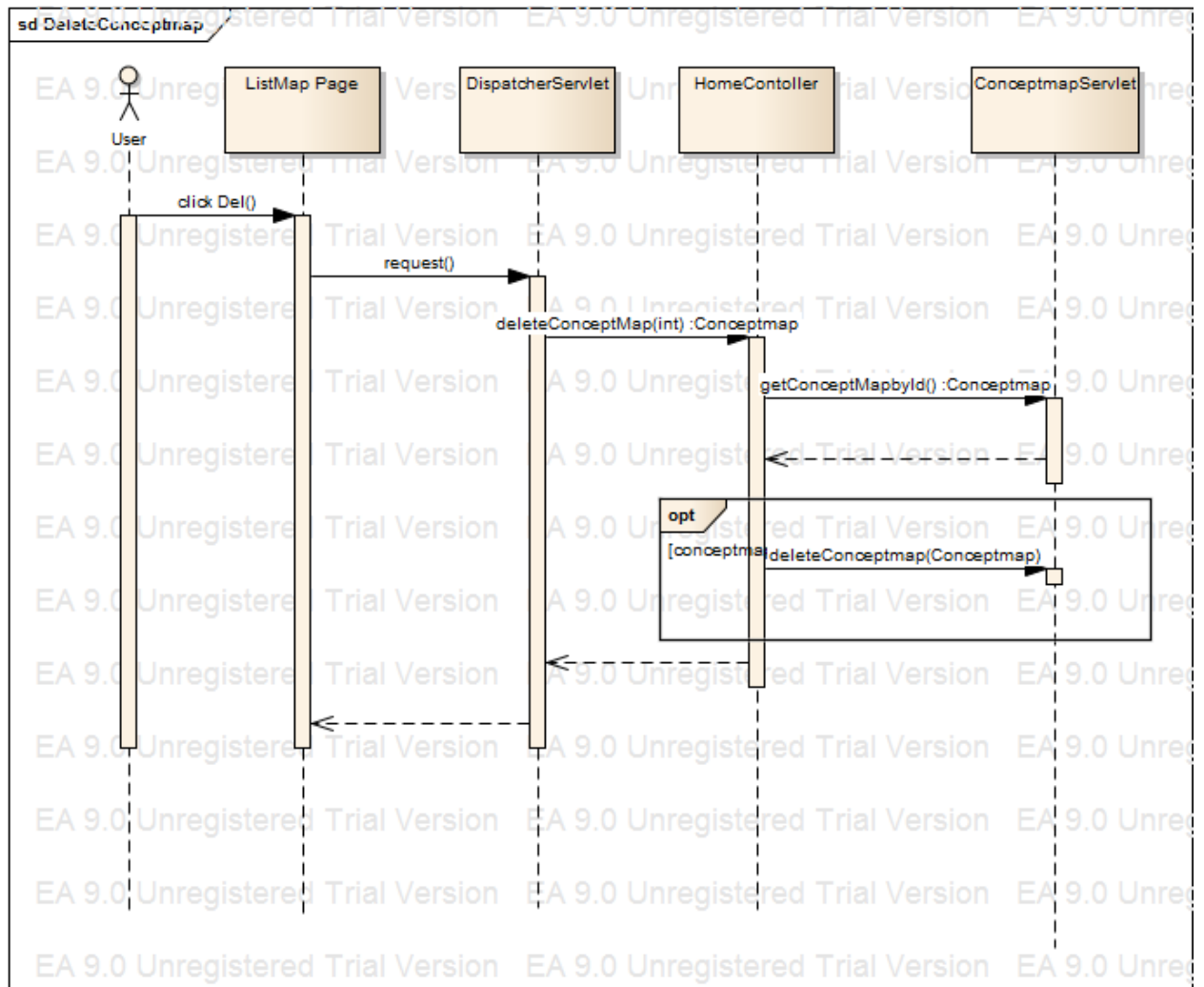


Figure 2.12: Sequence Delete ConceptMap.

2.3.9 Make Task.

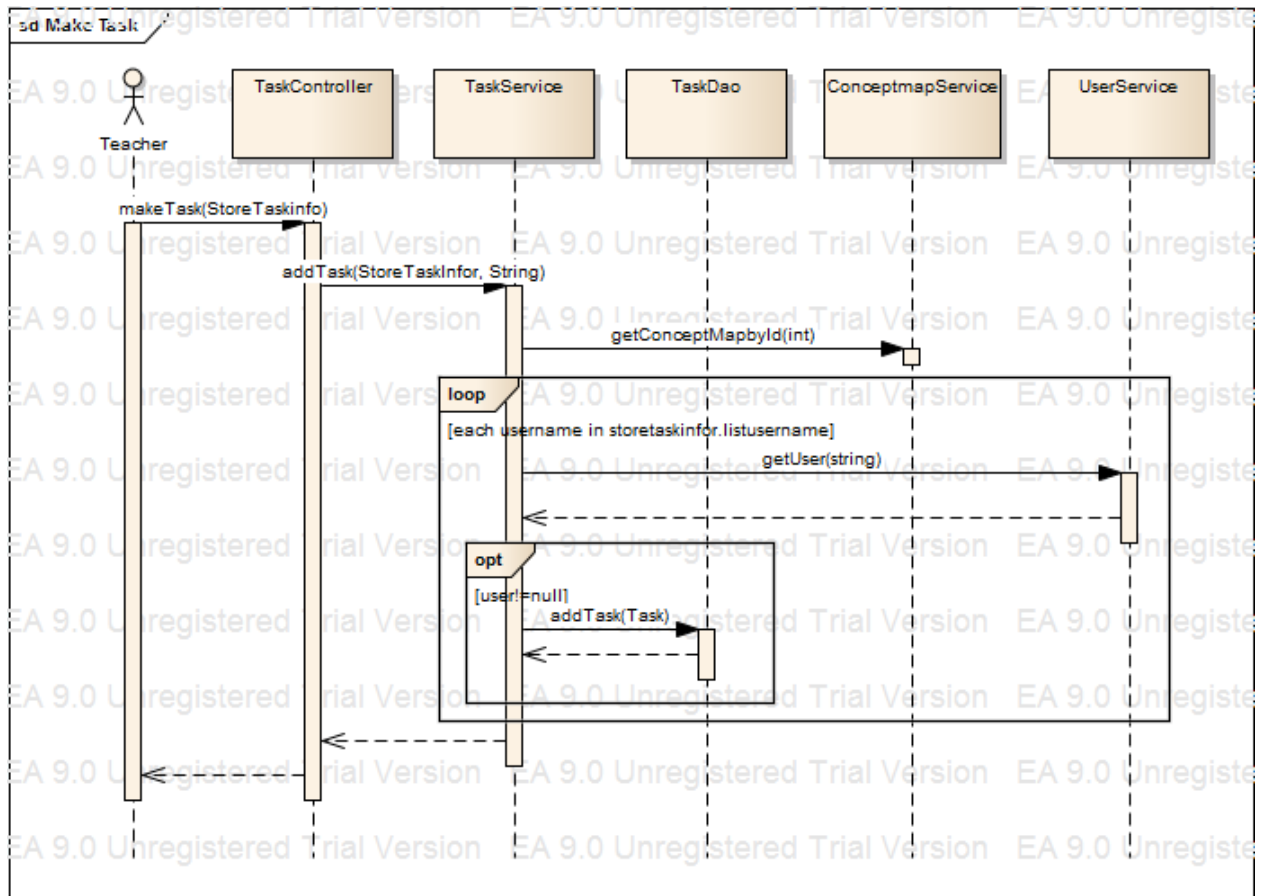


Figure 2.13: Sequence Make Task.

2.3.10 Chấm Điểm.

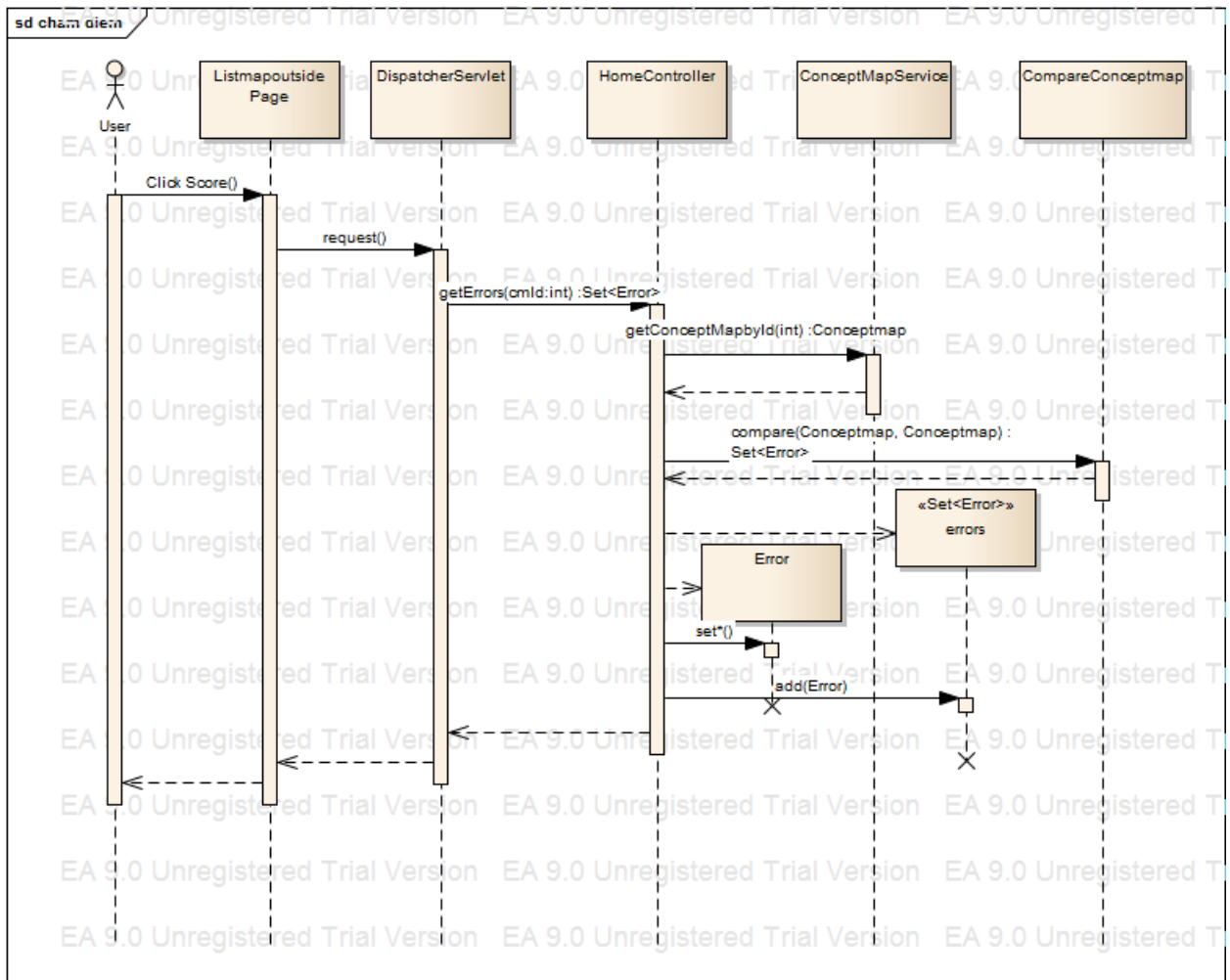


Figure 2.14: Sequence Chấm Điểm.

2.3.11 Add Friend

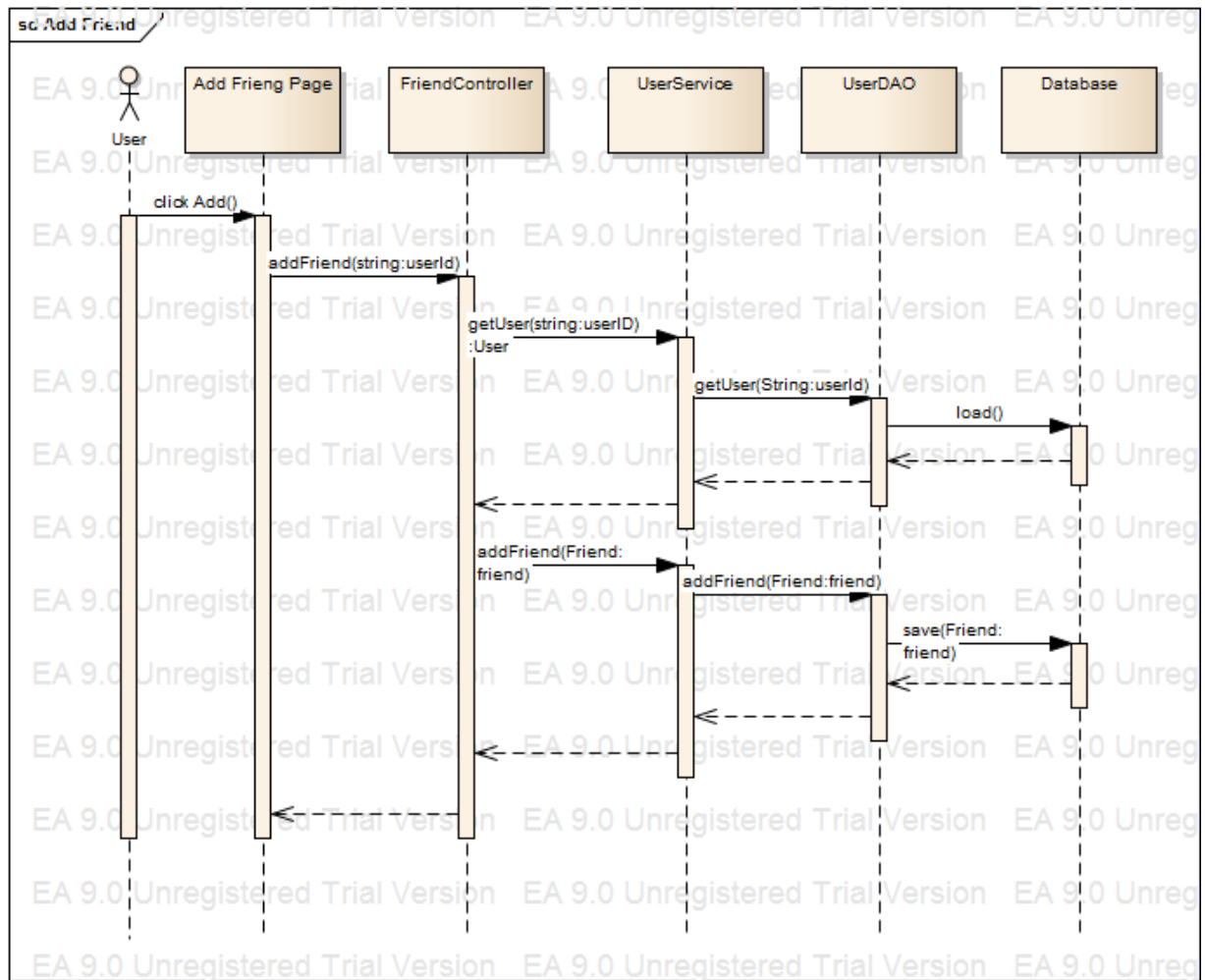


Figure 2.15: Sequene Add Friend

2.3.12 Download Document

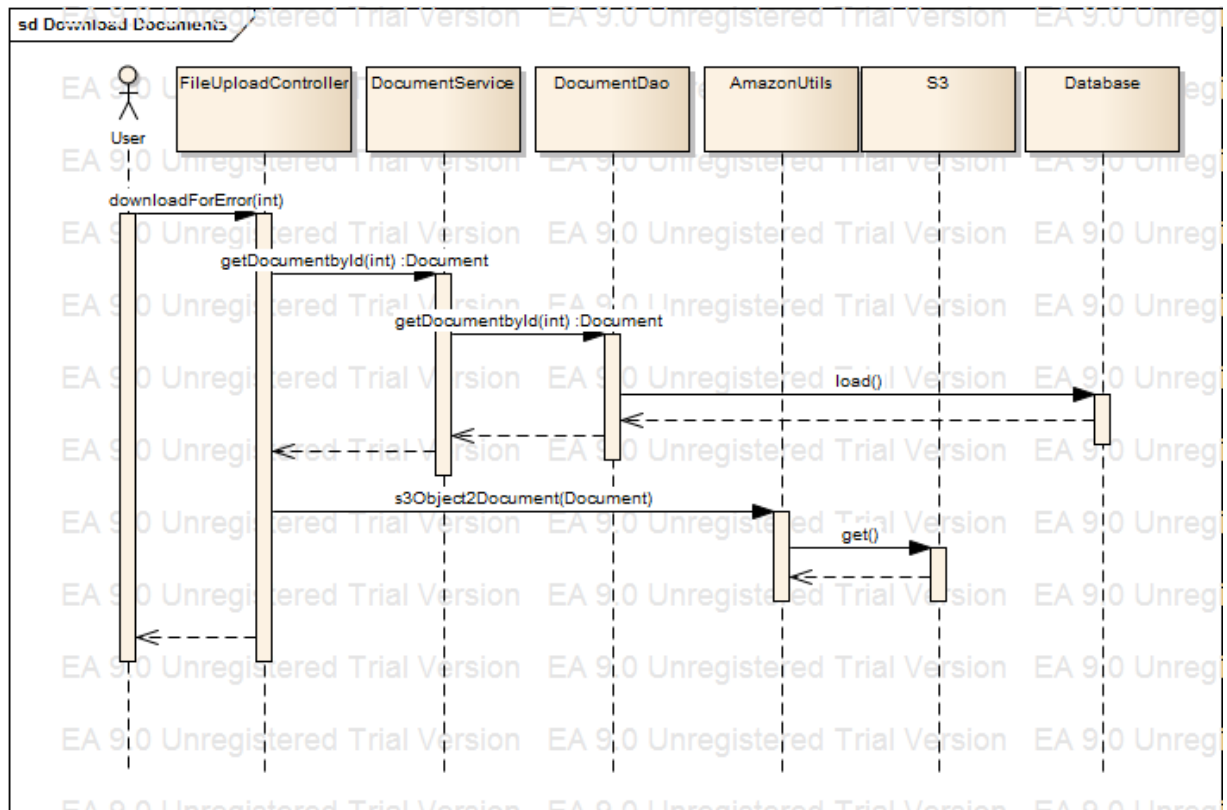


Figure 2.16: Sequence Download Document

2.3.13 Create Planner

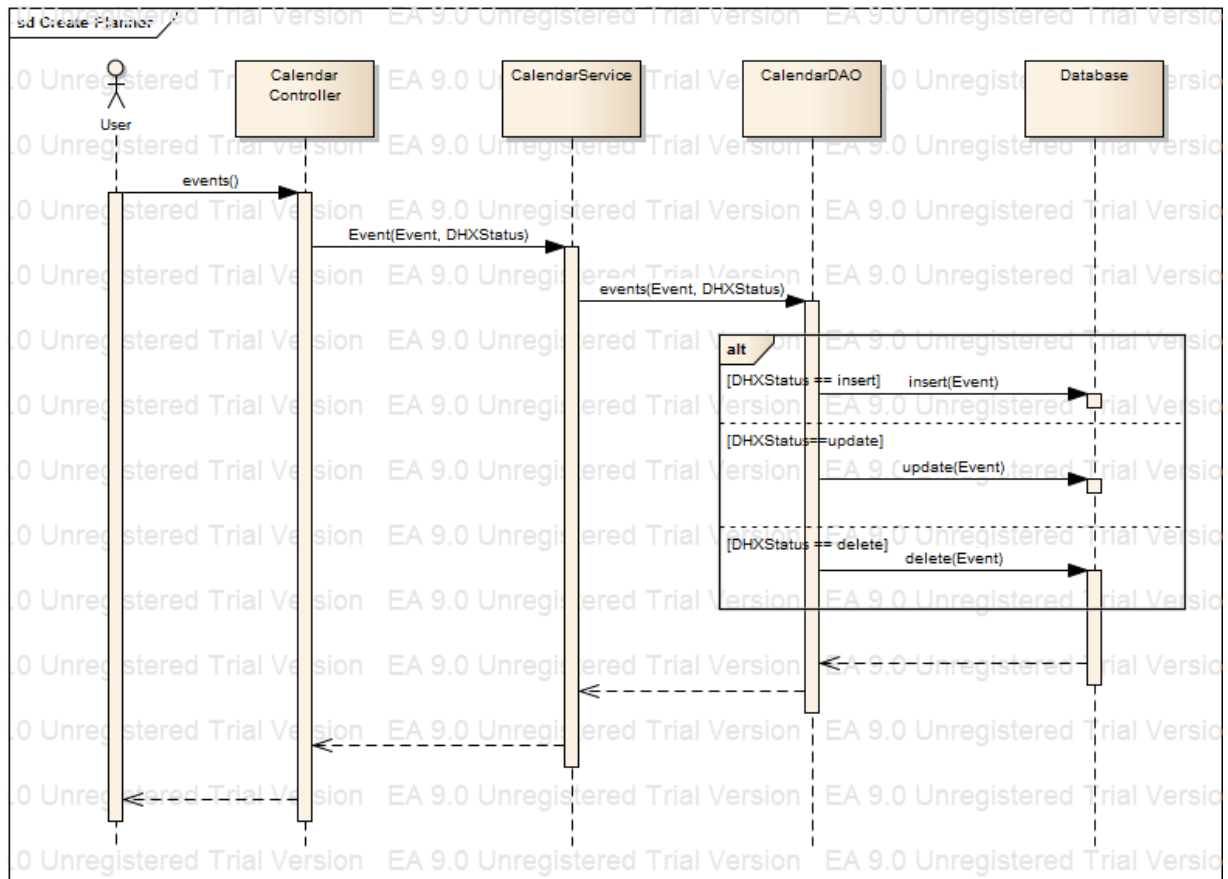


Figure 2.17: Sequence Create Planner

2.4 Thiết Kế Cơ Sở Dữ Liệu.

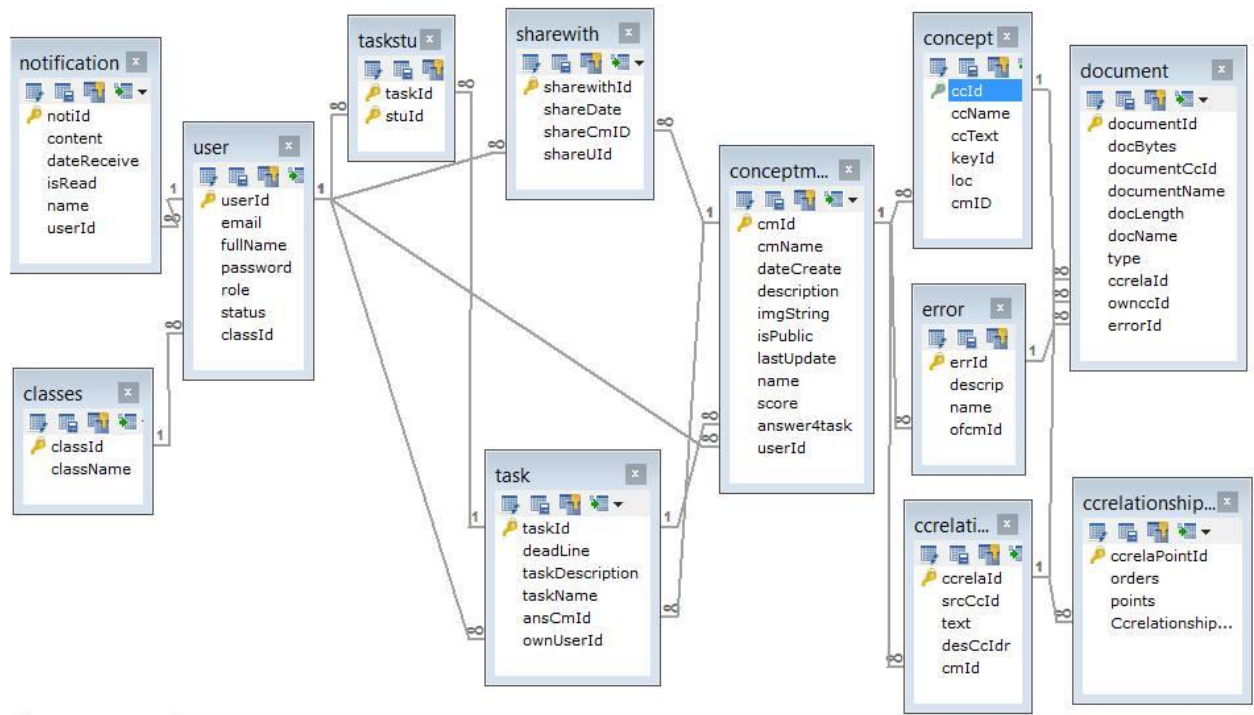


Figure 2.18: Database.

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

2.5 Thiết Kế Class Diagram

2.5.1 Class Model

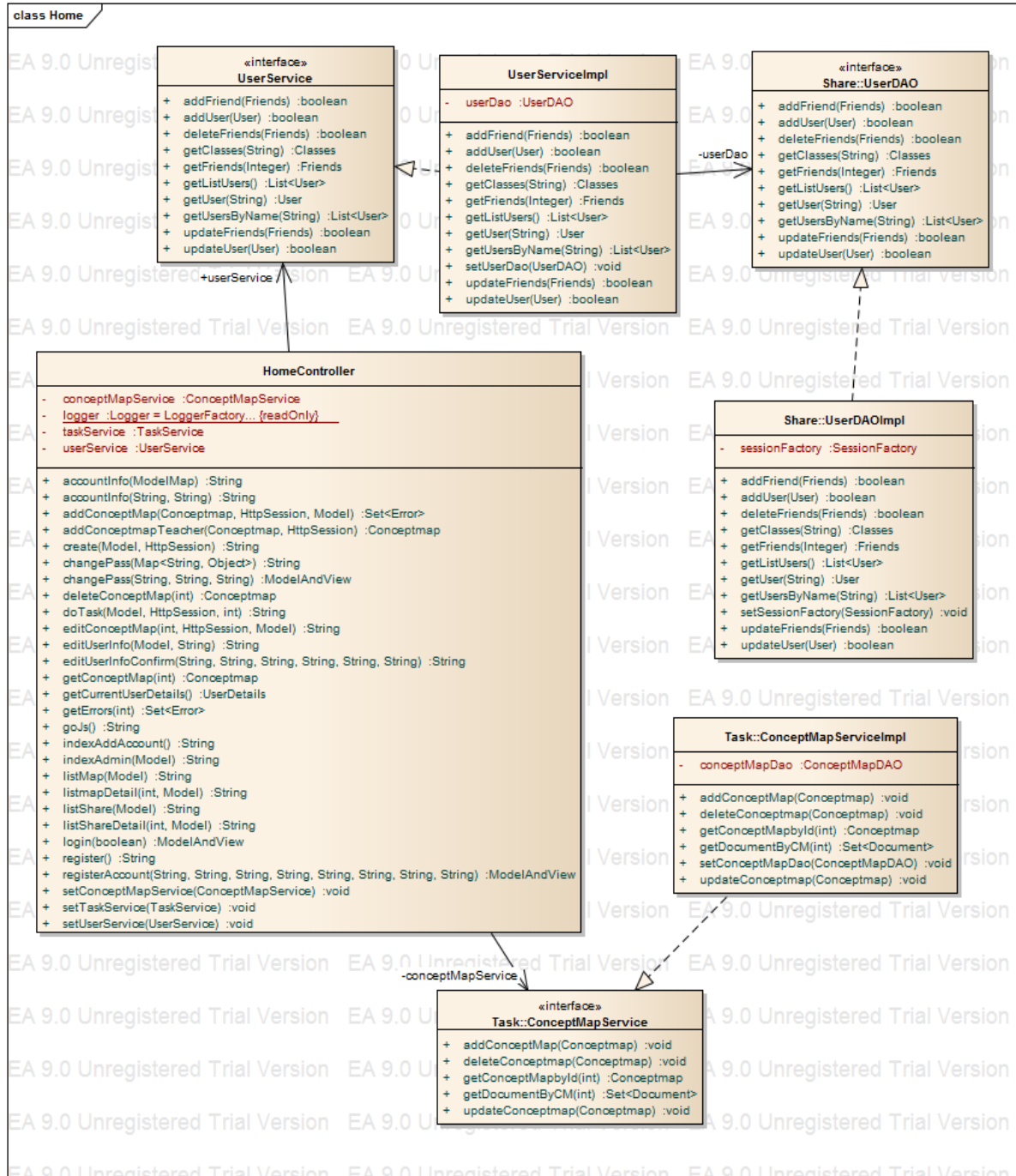


Figure 2.19: Class Diagram Model

Về, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

2.5.2 Class Task

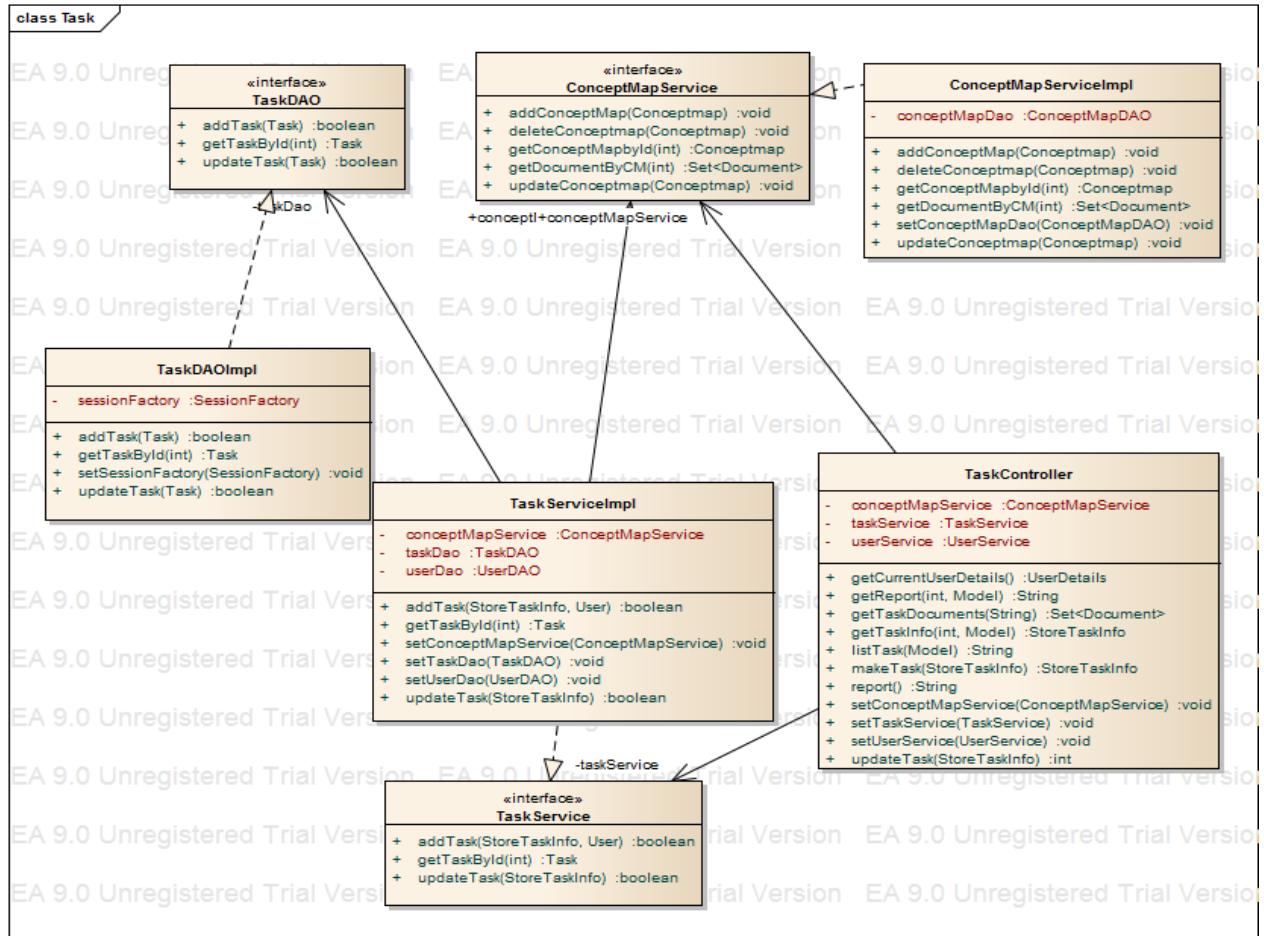


Figure 2.20: Class Diagram Task

2.5.3 Class FileUpload

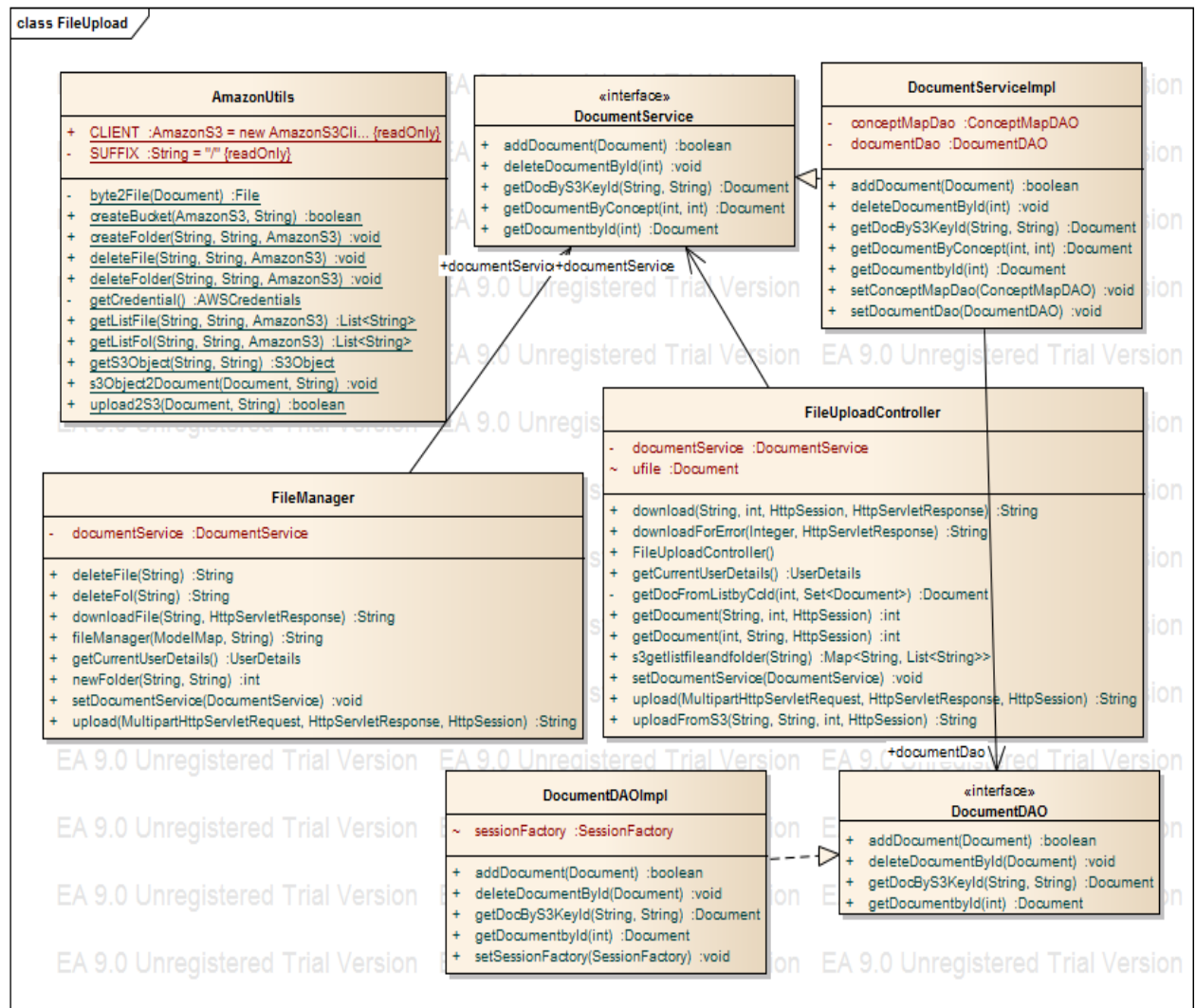


Figure 2.21: Class Diagram File Upload

2.5.4 Class Share

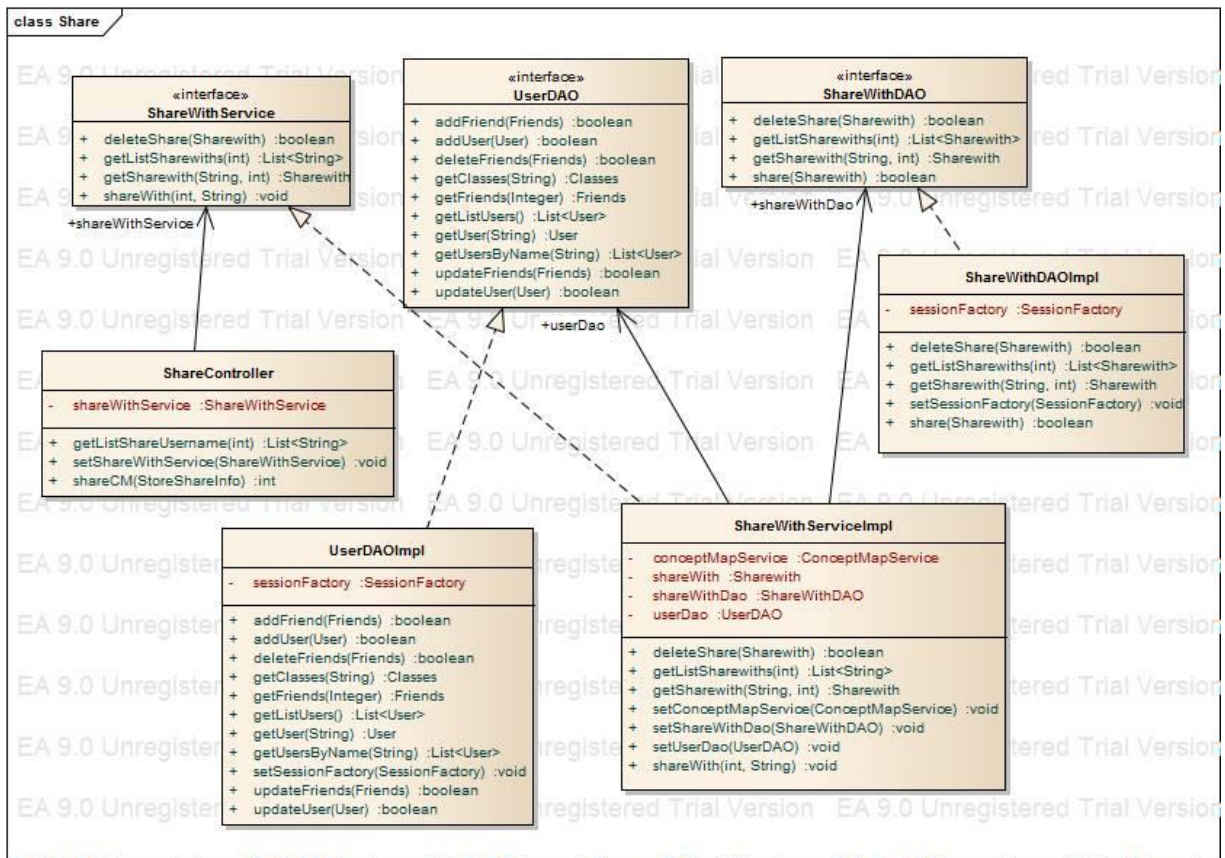
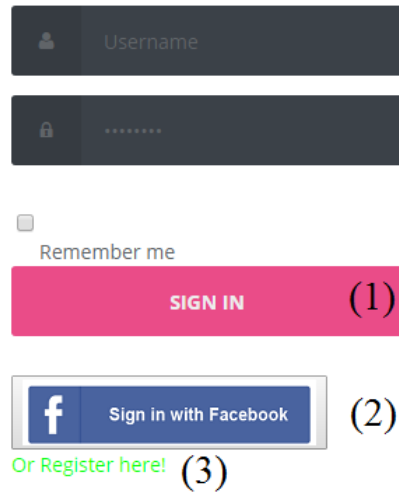


Figure 2.22: ClassDiagram Share

2.6 Thiết kế giao diện và xử lý

2.6.1 Thiết kế giao diện xử lý nghiệp vụ Login:

Giao diện Login



The image shows a login form with the following elements:

- A text input field labeled "Username" with a user icon.
- A password input field with a lock icon and masked characters "*****".
- A checkbox labeled "Remember me".
- A pink button labeled "SIGN IN" (annotated with (1)).
- A blue button labeled "Sign in with Facebook" with the Facebook logo (annotated with (2)).
- A green link labeled "Or Register here!" (annotated with (3)).

Figure 2.23: Giao diện đăng nhập

Các xử lý trên giao diện Login:

Table 2.1: Các xử lý trên giao diện Login

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	Đăng nhập	Click button “Login”	Thực hiện đăng nhập
2	Liên kết tài khoản với Facebook	Click button “Sign in with Facebook”	
3	Đăng Ký	Click link “Register here”	

2.6.2 Thiết kế giao diện xử lý nghiệp vụ Đăng Ký

Giao diện Register

Figure 2.24 shows the Register form interface. It includes a 'Welcome' banner, gender selection buttons (Male/Female), and input fields for Username, Student Id, Class, Khoa, Email address, Password, and Full Name. A 'Create Account' button and a link 'Or Login here!' are also present. Annotations (1), (2), and (3) highlight specific elements: (1) points to the gender buttons, (2) points to the 'Create Account' button, and (3) points to the 'Or Login here!' link.

Figure 2.24: Giao diện thêm tài khoản

Các xử lý trên giao diện Add Account

Table 2.2: các xử lý trên giao diện Add account

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	Chọn giới tính	Click button “Male” or “Female”	
2	Tạo tài khoản mới	Điền đầy đủ thông tin form và click button “Create Account”	
3	Đăng nhập	Click link “Login”	

2.6.3 Thiết kế giao diện xử lý nghiệp vụ “Share ConceptMap”

Giao diện Share ConceptMap

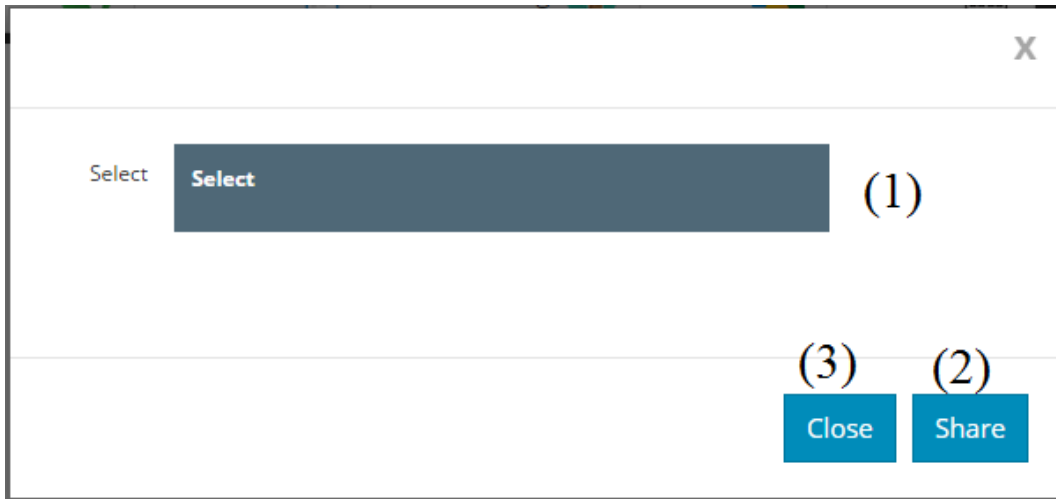


Figure 2.25: Giao diện Share ConceptMap

Các xử lý trên giao diện Share ConceptMapTable 2.3: Các xử lý trên giao diện Share Concept Map

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	Chọn user cần share	Chọn User name trong combobox.	
2	Đóng share	Click vào button Close	Đóng khung share concept map-> trở về trang xem danh sách concept map
3	Share concept map	Click share	Thực hiện share concept map

2.6.4 Thiết kế giao diện xử lý nghiệp vụ “Show List Task”

Giao diện ShowListTask

TaskId	Name	Description	Deadline	Get documents	Submit Answer
2	Task		1979-09-29 05:30:00.0	Get documents (1)	Expired
5	What is inheritance	inheritance	1979-09-22 16:30:00.0	Get documents	Expired
4	What is N	n	1979-09-21 15:50:00.0	Get documents	Expired
6	d	d	2015-06-13 11:16:03.0	Get documents	Submit (2)

Figure 2.26: Giao diện Show List Task

Danh sách các xử lý trên giao diện Show List Task

Table 2.4: Danh sách các xử lý trên giao diện Show List Task

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	Lấy Document	Click button “Get Document”	
2	Nộp bài	Click button “Submit”	

2.6.5 Thiết kế giao diện xử lý nghiệp vụ “Show List Share”

Giao diện Show List Share



Figure 2.27: Giao diện Show List Share

Danh sách xử lý trên giao diện Show List Share

Table 2.5: Danh sách xử lý trên giao diện Show List Share

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	ReadMore	Click Button “ReadMore”	

2.6.6 Thiết kế giao diện xử lý nghiệp vụ “Draw ConceptMap”

Giao diện Draw ConceptMap

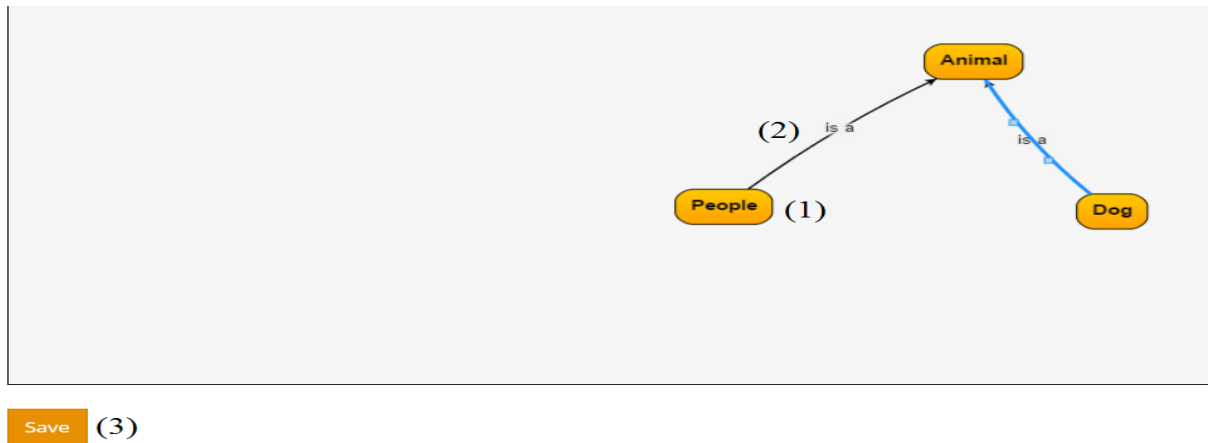


Figure 2.28: Giao diện Draw ConceptMap

Danh sách xử lý trên giao diện Draw ConceptMap

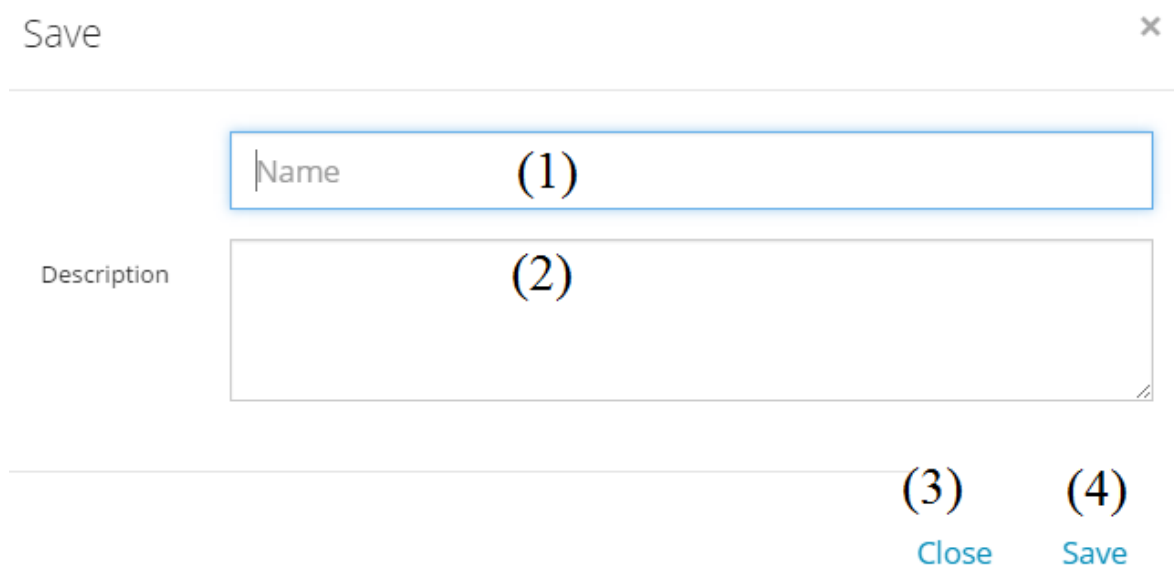
Table 2.6: Danh sách xử lý trên giao diện Draw ConceptMap

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	Vẽ Concept	Click vào khung	
2	Vẽ Realtionship	Nối 2 Concept lại với nhau	
3	Save ConceptMap	Nhấp vào Button “Save”	

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

2.6.7 Thiết kế giao diện xử lý nghiệp vụ “Lưu ConceptMap”

Giao diện Lưu ConceptMap



The image shows a 'Save' dialog box. The title bar says 'Save' and has a close button 'x'. Inside the dialog, there is a text input field labeled 'Name' (1) and a larger text area labeled 'Description' (2). At the bottom right, there are two buttons: 'Close' (3) and 'Save' (4).

Figure 2.29: Giao diện Lưu ConceptMap

Danh sách xử lý trên giao diện Save

Table 2.7: Danh sách xử lý trên giao diện Save

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	Nhập ConceptMap Name	Khi nhập xong textbox “name” và nhấp button “Save”	
2	Nhập ConceptMap Description	Khi nhập xong textbox “description” và nhấp button “Save”	

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

3	Close	Nhấp vào Button “Close”	
4	Save	Nhấp vào Button “Save”	

2.6.8 Thiết kế giao diện xử lý nghiệp vụ “Make task”

Giao diện Make Task

Make Task (5) x

Task Name (1)

Description (2)

DeadLine (3)

Task for (4)

(6)

Save

Figure 2.30: Giao diện Make Task

Danh sách xử lý trên giao diện MakeTask*Table 2.8: Danh sách xử lý trên giao diện MakeTask*

STT	Tên xử lý	Điều kiện gọi thực hiện	Ghi chú
1	Nhập ConceptMap Name	Nhập textbox “name”	
2	Nhập ConceptMap Description	Nhập textbox “description”	
3	Nhập hạn chót nộp bài	Chọn “Dead Line”	
4	Nhập lớp cần làm bài	Chọn Selector “Class”	
5	Close	Nhấp vào Button “X”	
6	Save	Nhấp vào Button “Save”	

Chương 3. Cài đặt và kiểm thử

3.1 Cài Đặt

OpenShift

OpenShift là Red Hat's Platform-as-a-Service (PaaS) hỗ trợ người lập trình có thể phát triển, host, and thay đổi kích thước của ứng dụng trên môi trường đám mây. Và đặc biệt là hoàn toàn miễn phí, tất nhiên cũng sẽ có những gói trả phí với những tính năng nâng cao.



Figure 3.1: Open Shift

Openshift hỗ trợ hầu hết các ngôn ngữ cũng như service để phát triển web hiện nay: Java, JBoss AS, JBoss EAP, Tomcat, WildFly, PHP, Node.js, Python, Ruby, Perl, Zend Server, Vert.x, DIY/Custom.

Và 4 database server: MySQL, MongoDB, PostgreSQL, Amazon RDS

JBoss OpenShift Eclipse Plugin

Hỗ trợ đầy đủ môi trường để người lập trình có thể tạo, deploy cũng như thay đổi source code trên mây một cách dễ dàng.

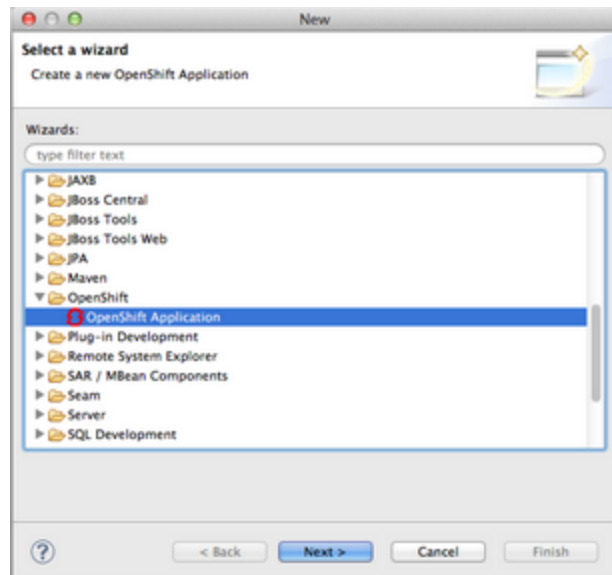


Figure 3.2: Jboss

Người lập trình có thể tạo trực tiếp một ứng dụng trên Opeshift thông qua Jboss Openshift tool mà không cần phải qua nhiều bước phức tạp

Một Openshift Project là một Git project, như vậy khi có sự thay đổi source code thì người lập trình chỉ cần đẩy những thay đổi đó lên server, server sẽ tự động được xây dựng lại theo code mới được gửi lên.

Ngoài ra còn có thể xem log trực tiếp trên eclipse

3.2 Test

Table 3.1: Test Login

Module Code		Website Login		
Tester		Võ Thái Ngọc		
ID	Test Case Description	Test Case Procedure	Expected Output	Browser type
LOGI N_00 1	Kiểm tra nút check box remember me	1.Check vào remember me 2.Check vào remember me	1.Checkbox được chọn 2.Checkbox bỏ chọn	Chorme
LOGI N _002	Kiểm tra đăng nhập thành công với 1 tài khoản đã có sẵn trong database với quyền admin và chưa click vào checkbox Remember me	1.Nhập vào textbox Username kí tự admin 2.Nhập vào textbox Password kí tự 123 3.Click vào button SIGN IN	2.Kí tự trong textbox chuyển thành *** 3.Chuyển đến trang AccountMng	Chorme
LOGI N_00 3	Kiểm tra đăng nhập thành công với 1 tài khoản đã có	1.Nhập vào textbox Username kí tự admin 2.Nhập vào textbox Password kí tự 123	2.Kí tự trong textbox chuyển thành *** 3. Check box remember me được chọn	Chorme

	sẵn trong database với quyền admin và click vào checkbox Remember me	3. Click checkbox remember me 4. Click vào button SIGN IN. 5. Tắt trình duyệt 6. Mở trình duyệt truy cập vào lại trang AccountMng	4. Chuyển đến trang AccountMng 6. Hiển thị trang AccountMng	
LOGI N_00 4	Kiểm tra đăng nhập thành công với 1 tài khoản đã có sẵn trong database với quyền student và chưa click vào checkbox Remember me	1. Nhập vào textbox Username kí tự student1 2. Nhập vào textbox Password kí tự 123 3. Click vào button SIGN IN	2. Kí tự trong textbox chuyển thành *** 3. Chuyển đến trang listmap của student	Chorme
LOGI N-005	Kiểm tra đăng nhập thành công với 1 tài khoản đã có sẵn trong database với quyền student	1. Nhập vào textbox Username kí tự student1 2. Nhập vào textbox Password kí tự 123 3. Click checkbox remember me	2. Kí tự trong textbox chuyển thành *** 3. Check box remember me được chọn 4. Chuyển đến trang listmap student	Chorme

	và click vào checkbox Remember me.	4.Click vào button SIGN IN. 5. Tắt trình duyệt 6. Mở trình duyệt truy cập vào lại trang listmap student	6. Hiện thị trang listmap student	
LOGI N_00 6	Kiểm tra đăng nhập thành công với 1 tài khoản đã có sẵn trong database với quyền teacher và chưa click vào checkbox Remember me	1.Nhập vào textbox Username kí tự teacher 2.Nhập vào textbox Password kí tự 123 3.Click vào button SIGN IN	2.Kí tự trong textbox chuyển thành *** 3.Chuyển đến trang listmap của teacher	Chorme
LOGI N-007	Kiểm tra đăng nhập thành công với 1 tài khoản đã có sẵn trong database với quyền teacher và click vào checkbox	1.Nhập vào textbox Username kí tự teacher 2.Nhập vào textbox Password kí tự 123 3. Click checkbox remember me 4.Click vào button SIGN IN.	2.Kí tự trong textbox chuyển thành *** 3. Check box remember me được chọn 4.Chuyển đến trang listmap của teacher 6. Hiện thị trang lismap của teacher	Chorme

	Remember me.	5. Tắt trình duyệt 6. Mở trình duyệt truy cập vào lại trang listmap teacher		
LOGI N-008	Kiểm tra đăng nhập không thành công khi nhập vào username và password không có trong database	1.Nhập vào textbox Username kí tự 123 2.Nhập vào textbox Password kí tự 123 3.Click vào button SIGN IN	2. giá trị trong password trở thành **** 3. Dòng màu đỏ hiện ra “wrong user or password”	Chorme
LOGI N-009	Kiểm tra đăng nhập không thành công khi nhập vào username có trong database và password sai không có trong database	1.Nhập vào textbox Username kí tự admin 2.Nhập vào textbox Password kí tự 2121 3.Click vào button SIGN IN	2. giá trị trong password trở thành **** 3. Dòng màu đỏ hiện ra “wrong username or password”	Chorme
LOGI N-010	Kiểm tra khi đăng nhập lần đầu không thành công và	1.Nhập vào textbox Username kí tự 123 2.Nhập vào textbox Password kí tự 123	2.Kí tự trong textbox chuyển thành *** 3.Phía trên textbox Username hiển thị dòng	Chorme

	lần 2 thành công	3.Click vào button SIGN IN 4.Nhập vào textbox Username kí tự admin 5.Nhập vào textbox Password kí tự 123 6.Click vào button SIGN IN	chữ màu đỏ “wrong username or password” 5.Kí tự trong textbox chuyển thành *** 6.Chuyển đến trang tương ungws	
LOGI N-011	Kiểm tra các trường bắt buộc	1.Không nhập vào username và password	1. textbox Username hiển thị dòng chữ The User name field is required. Và textbox Password hiển thị dòng chữ The Password field is required	Chorme

Table 3.2: Test Trang List map student

Module Code		Website List Map Student		
Tester		Nguyễn Duy Tấn Phát		
ID	Test Case Description	Test Case Procedure	Expected Output	Browser type
LMST UDEN T_01	Kiểm tra hiển thị list map student	1 .Nhấn vào tab list map trên thanh menu	1. Hiện thị danh sách các concept map của student cùng với các task đã làm	Chorme
LMST UDEN T-02	Kiểm tra edit concept map	1.nhấn vào tab list map 2. chọn một concept map tên A, nhấn edit 3. Thêm một concept với nội dung NewConcept 4. Nhấn Save 5. Nhấn Save của popup 6. nhấn tab list map 7. Chọn concept map A	1. Hiện thị danh sách concep map cùng các task đã làm 2. Hiện thị nội dung concept map A 3. Thêm một concept NewConcept vào concept map 4. Popup hiển thị lên 5. Concept map được lưu lại 6. Hiện thị danh sách conceptmap cùng các task đã làm	Chorme

			7. concept map đã được update	
LMST UDEN T -03	Kiểm tra share concept map cho một tài khoản có trong database	1.Nhấn vào tab list map 2. chọn một concept map A, nhấn share 3. Điền tên username student2 vào ô textbox 4. Nhấn Share 5. Logout đăng nhập vào tài khoản student2 6. Vào listshare 7. Nhấn Edit concept map đã được share	1.Hiển thị list concept map 2. Một popup hiển thị lên 6. Hiển thị được concept map A đã share 7. Hiển thị nội dung concept map A	Chorme
LMST UDEN T_04	Kiểm tra chức năng delete concept map	1.Nhấn vào tab list map 2. chọn một concept map A, nhấn del 3. Nhấn Yes	1.Hiển thị danh sách các concept map 2. Một popup hiển thị lên 3. concept map A bị xóa	Chorme
LMST UDEN T_05	Kiểm tra danh sách concept map	1.Nhấn vào list map trên menu	1. hiển thị các task và concept map	Chorme

	trong task đã làm	2.Chọn read more của một task tên TASK đã làm	2. Hiện thị ra các concept map đã làm cho TASK đã được làm	
LMST UDEN T_06	Xem score của một concept map đã làm của một task	1.Chọn một concept map LAM, chọn score	1.Hiển thị điểm của concept map LAM là 100	Chorme
LMST UDEN T_07	edit concept map đã làm của một task	1.Chọn một concept map LAM, chọn edit 2. thêm một concept vào concept map NewConcept 3. Nhấn save 4. Chọn list map 5. Chọn task TASK 6. Chọn concept map LAM	1. Hiện thị nội dung concept map LAM 2. Hiện thị thêm một concept tên NewConcept 6. Hiện thị nội dung concept map LAM đã có thêm một concept NewConcept	Chorme
LMST UDEN T_08	Hiện thị nội dung một task chưa làm	1.Chọn list map 2.Chọn task NOTDO	1.Hiển thị các concept và task 2.Hiển thị một vùng không có gì	Chorme

LMST UDEN T -09	Kiểm tra share concept map cho một tài khoản không có trong database	1.Nhấn vào tab list map 2. chọn một concept map A, nhấn share 3. Điền tên username student9 vào ô textbox 4. Nhấn Share	4. không có lỗi xảy ra	Chorme
-----------------------	--	--	------------------------	--------

Table 3.3. Test Trang Create

Module Code		Website Create concept Map student		
Tester		Võ Thái Ngọc		
ID	Test Case Description	Test Case Procedure	Expected Output	Browser type
CCS TU DE NT_ 001	Kiểm tra chức năng Create concept map của student	1.Nhấn vào tab create trên thanh menu 2. Thêm một concept tên AABB 3. Nhấn save 4. Điền Name 123 5. Điền Description aabb 6. Nhấn Save 7. Nhấn về tab list map	1.Hiển thị trang create concept map 2.Hiển thị thêm một concept trong concept map 3. Một popup hiển thị lên 7. Hiển thị thêm một conceptmap với tên 123	Chrome
CCS TU DE NT_ 002	Create một concept map là một bài làm	1.Chọn List task trên thanh menu 2.Chọn task TASK, chọn DO 3. Thêm 1 Concept với tên NewConcept 4. Nhấn save	1. Hiển thị các task của student 2. Hiển thị trang create concept map 3. Thêm một concept thành công với tên NewConcept 4. Hiển thị một popup	Chrome

		5. Điền tên 321 6. Điền Description 54321 7. Nhấn save 8. Trở về ListMap, chọn TASK 9. Chọn concept map tên 321	7. Hiển thị kết quả 50 điểm 9. Hiển thị nội dung 321	
CCS TU DE NT_ 003	Kiểm tra list share	1. Nhấp vào tab list share 2. Chọn read more của concept map LAM	1. Hiển thị các concept map được share 2. Hiển thị nội dung concept map LAM	Chrome
CCS TU DE NT_ 004	Kiểm tra List task	1. Nhấp vào list task 2. Chọn Task TASK, chọn read more	1. Hiển thị các task được giao của student đó 2. Hiển thị tên và description của task đó	Chrome

Table 3.4: Test trang admin

Module Code		Website admin		
Tester		Võ Thái Ngọc		
ID	Test Case Description	Test Case Procedure	Expected Output	Browser type
ADMIN_001	Kiểm tra hiển thị danh sách user	Bấm vào Account management	Hiển thị các user có trong hệ thống	Chrome

3.3 Kết quả khảo sát từ người dùng.

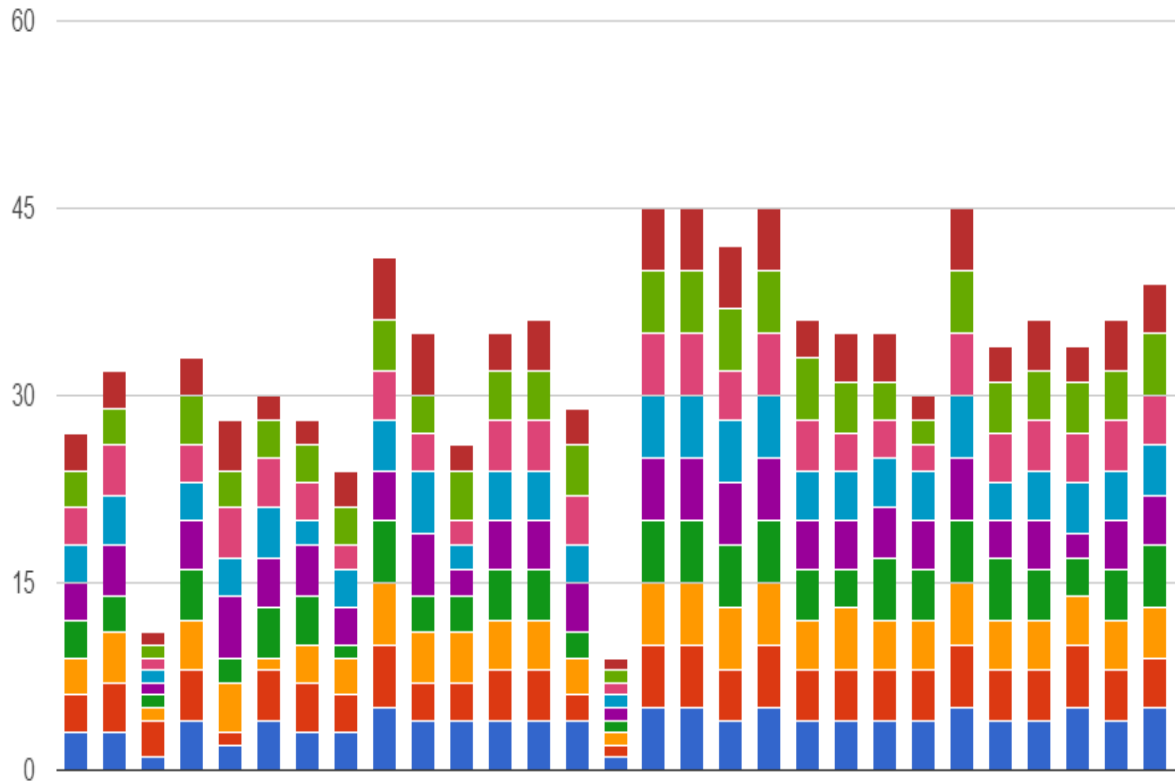












Figure 3.3: Khảo sát từ người dùng



3.4 So sánh với các sản phẩm tương đương

Table 3.7: So sánh sản Phẩm tương đương

Stt	Tính năng\Sản phẩm	Sản phẩm hiện tại	text2mindmap.com	mindmup.com	bubbl.us	examtime.com
1	Hỗ trợ thêm tên cho các relationship	✓	✗	✗	✗	✗
2	Hỗ trợ đính kèm tài liệu cho các concept	✓	✗	✓	✗	✓
3	Hỗ trợ tạo bài tập và giao bài tập cho sinh viên	✓	✗	✗	✗	✗
4	Hỗ trợ chấm điểm tự động khi sinh viên giao bài tập	✓	✗	✗	✗	✗
5	Hỗ trợ lưu trữ theo dạng gridview kèm hình ảnh đại diện cho conceptmap	✓	✗	✗	✗	✓
6	Hỗ trợ thêm các tên thay thế cho concept trong khi vẽ conceptmap	✓	✗	✗	✗	✗
7	Hỗ trợ xuất conceptmap ra nhiều định dạng khác nhau (ảnh,svg,pdf,..)	✗	✓	✓	✓	✓

8	Hỗ trợ undo/redo trong lúc vẽ conceptmap					
9	Hỗ trợ chia sẻ conceptmap					
10	Giao diện thân thiện, dễ sử dụng	5/10	8/10	8/10	5/10	10/10
11	Màu sắc sinh động tạo	3/10	8/10	8/10	3/10	10/10

Phần C: KẾT LUẬN

Tìm hiểu dịch vụ cloud storage của amazon (Simple storage service - S3)

- Cho phép upload một file bất kỳ từ web application lên một bucket tùy ý trên amazon S3.
- Khi người dùng chọn tham chiếu file trên, web app sẽ gửi request đến amazon s3 thông qua amazon web interface để nhận lại file này và hiển thị trên giao diện web cho người dùng.

Những chức năng đã hoàn thành

- Vẽ Concept Map
- Save Concept Map
- Share Concept Map cho 1 hay nhiều User khác
- Load 1 Concept Map đã vẽ trước đó
- Xem danh sách Concept Map
- Xem danh sách task
- Xem danh sách Share
- Login hệ thống
- Logout hệ thống
- Upload document
- Tạo Task
- Chấm điểm Concept Map
- Login thông qua Facebook
- Hỗ trợ lưu trữ ở amazon s3

Ưu điểm

- Có 2 chức năng mới so với các sản phẩm khác:
 - o Chấm điểm
 - o Tạo Task

Vẽ, Lưu trữ và Chia sẻ ConceptMap dùng Spring Framework

Nhược điểm

- Giao diện chưa được đẹp

Giải pháp cho hướng phát triển cho tương lai

- Phát triển thêm một số chức năng:

Tài liệu tham khảo

- [1]. <http://www.tutorialspoint.com/spring/>
- [2]. <http://msdn.microsoft.com/en-us/library/ff921152.aspx>
- [3]. <http://nthoai.blogspot.com/2008/09/su-dung-aop-voi-spring-net.html>
- [4]. <https://www.youtube.com/channel/UCvMawIqYD-1UE8Jn1GdaBcQ>
- [5]. <http://mergetag.com/spring-aop-advice-on-annotated-controllers-2/>
- [6]. <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>
- [7]. http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm
- [8]. <http://sammy-woo.blogspot.com/2011/07/t3.html>
- [9]. <http://www.tutorialspoint.com/hibernate/>
- [10]. <http://gojs.net/latest/index.html>
- [11]. <http://www.journaldev.com/2888/spring-framework-tutorial-with-example-projects>
- [12]. <http://www.mkylong.com/tutorials/spring-mvc-tutorials/>
- [13].
https://www.openshift.com/products?gclid=CjwKEAiAw56lBRCs29jB9uOvkygSJADnD3-6Ato7rUh7hPWozH69WFqSSGTtsBFWHAcsR5R1DzAliRoCv6zw_wcB
- [14]. <https://blog.openshift.com/day-28-openshift-eclipse-integration-for-java-developers/>