Trường Đại học Công nghệ - ĐHQGHN
Khoa Công nghệ thông tin

.

BÀI TẬP LỚN: PHÂN TÍCH & THIẾT KẾ HƯỚNG ĐỐI TƯỢNG

Giảng viên: PGS.TS. Đặng Đức Hạnh

ThS. Trần Mạnh Cường

HANDBOOK
ỨNG DỤNG CHĂM SÓC SỨC KHỎE TRỰC TUYẾN

Ngày: 01/04/2024

Chuẩn bị bởi: Nhóm 5

Mục lục

Lịch sử sửa đổi

Họ tên	Thời gian	Lý do sửa đổi	Phiên bản
Lê Trọng Minh	01/04/2024	Khởi tạo mẫu tài liệu	1.0
Dương Nguyễn Việt Anh	19/05/2024	Hoàn thiện tài liệu	1.1

1. Tổng quan

1.1. Giới thiệu

Tài liệu này bổ sung cho tài liệu khóa học "Object-oriented Analysis and Design" (Phân tích và Thiết kế hướng đối tượng) của lớp nhóm 5, với Ứng dụng chăm sóc sức khỏe trực tuyến.
Báo cáo này được viết dựa trên định dạng báo cáo "IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998". Báo cáo dựa trên nội dung được chấp nhận và đáp ứng các yêu cầu khác của Ứng dụng chăm sóc sức khỏe trực tuyến.

1.2. Quy ước tài liệu

Không

1.3. Đối tượng dự kiến và đề xuất cách đọc

Các đối tượng đọc khác nhau dành cho tài liệu này là:

- Quản trị dự án: Người phụ trách quản lý và chịu trách nhiệm về chất lượng hệ thống. Quản trị dự án nên đọc toàn bộ tài liệu để phục vụ việc lên kế hoạch và phân công công việc.
- Nhà phát triển: Người thực hiện nhiệm vụ phát triển hệ thống từ đầu vào là bản thiết kế và tài liệu để tạo thành đầu ra là một phiên bản có thể chạy được.
- Người kiểm thử: Người kiểm thử đọc tài liệu này để viết các ca kiểm thử.
- Người viết tài liệu: Người sẽ viết tài liệu trong tương lai (các báo cáo, biên bản).

1.4. Phạm vi dự án

Phần mềm chăm sóc sức khỏe trực tuyến được thiết kế nhằm cung cấp các dịch vụ chăm sóc sức khỏe thông qua nền tảng trực tuyến. Phần mềm sẽ được phát triển dưới dạng ứng dụng web để có thể truy cập từ mọi thiết bị kết nối internet. Người dùng cuối bao gồm các cá nhân quan tâm đến việc duy trì và cải thiện sức khỏe cá nhân, bao gồm người bệnh, người tìm kiếm thông tin sức khỏe, cũng như nhà cung cấp dịch vụ y tế. Phần mềm cho phép người dùng có thể tạo và quản lý hồ sơ sức khỏe cá nhân, bao gồm thông tin về lịch sử bệnh lý, thuốc đã dùng, kết quả xét nghiệm, và các thông tin liên quan khác. Người bệnh cũng có thể đặt lịch khám, tương tác trực tiếp với các bác sĩ, chuyên gia y tế thông qua cuộc gọi video, tin

nhắn, hoặc hệ thống thảo luận trực tuyến. Các bác sĩ có thể quản lý hồ sơ bệnh án của bệnh nhân, theo dõi sức khỏe, tư vấn, giải đáp các câu hỏi của bệnh nhân, …

1.5. Tài liệu tham khảo

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.

[2] Slide môn học Phân tích và thiết kế hướng đối tượng do giảng viên cung cấp.

[3] Từ điển thuật ngữ của *Ứng dụng chăm sóc sức khỏe trực tuyến*.

2. Cơ chế phân tích

2.1. Persistence

Các lớp có vai trò lưu trữ dữ liệu, cần xác định:

- **Volume:** số lượng đối tượng cần lưu trữ là bao nhiêu?
- **Duration:** các đối tượng thường được lưu trong bao lâu?
- **Retrieval mechanism:** một đối tượng cụ thể được xác định và lấy về như thế nào?
- **Reliability:** các đối tượng cần sống sót nếu như một tiến trình bị crash, hay cả hệ thống crash?
- **Update frequency:** các đối tượng được tạo một lần và gần như không bị thay đổi, hay các đối tượng thường xuyên được cập nhật?

2.2. Communication

Đối với tất cả các thành phần mô hình cần giao tiếp với các thành phần hoặc dịch vụ đang chạy trong các tiến trình hoặc luồng khác, cần xác định:

- **Latency**: Các tiến trình phải giao tiếp với nhau nhanh đến mức nào?
- **Synchronicity**: Giao tiếp không đồng bộ
- **Size of message**: Một chuỗi rộng có thể phù hợp hơn là một số
- **Protocol**: Kiểm soát các luồng, đệm, v.v.

2.3. Security

Các lớp có yêu cầu bảo mật cao, như HealthRecord, MedicalImage, Appointment, ExaminationResult, User cần xác định:

- **User granularity:** Người dùng hệ thống có các role gì?
- **Security rules:** Đề ra các chuẩn mực về bảo mật để bảo vệ dữ liệu người dùng
- **User privilege:** Các role của người dùng hệ thống có thể làm được những gì trên hệ thống?

2.4. Các cơ chế khác

- **Error detection / handling / reporting**: Các lỗi nên được phát hiện, xử lý và báo cáo như thế nào?
- **Distribution**: Dữ liệu nên được lưu vào máy chủ nào?
- **Transaction management**: Một giao dịch nên được hoàn tất như thế nào?

2.5. Analysis-to-Design-to-Implementation Mechanisms Map

Cơ chế phân tích	Cơ chế thiết kế	Cơ chế cài đặt
Volume	RDBMS	JDBC + MySQL
Duration	RDBMS	JDBC + MySQL
Retrieval mechanism	RDBMS	JDBC + MySQL
Reliability	RDBMS	JDBC + MySQL
Latency	Communication	
Synchronicity	Communication	
Size of message	Communication	
Protocol	Communication	
Update frequency	Security	Secure User Set-up

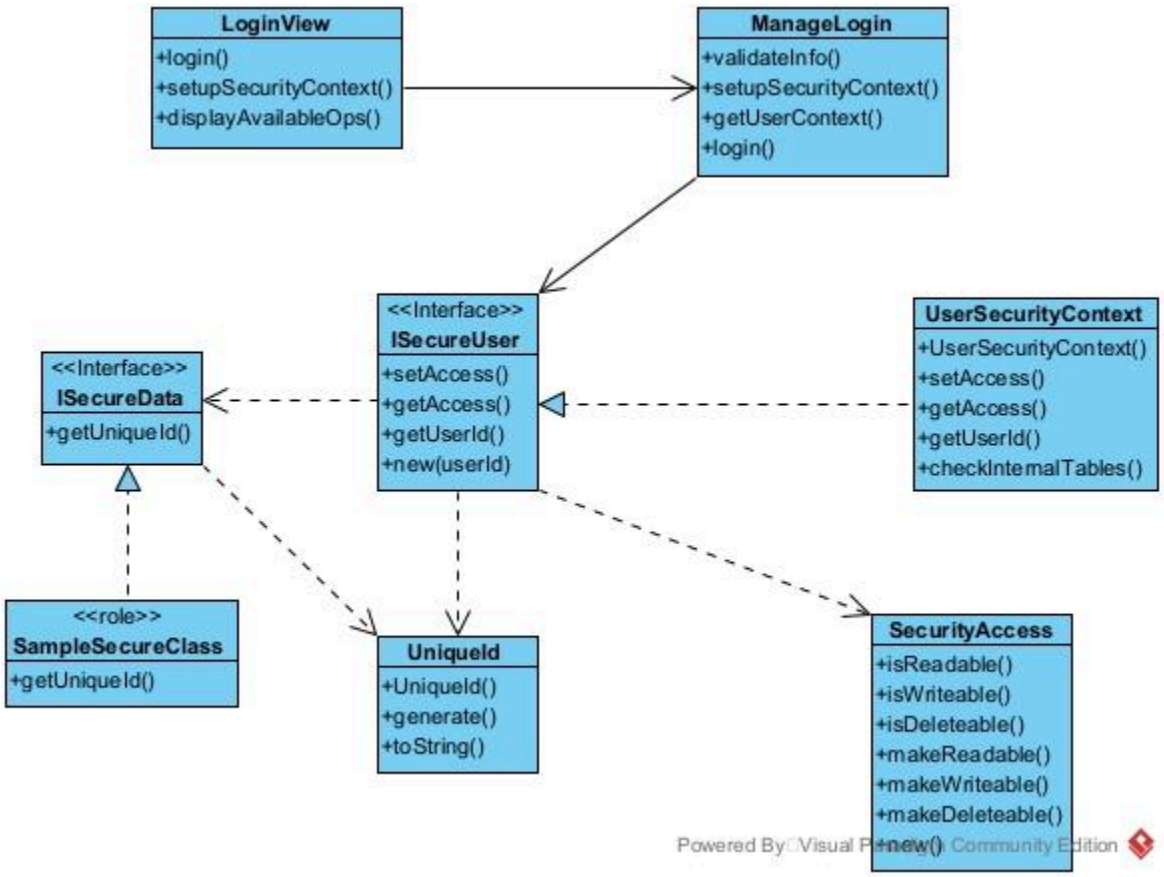
User granularity	Security	Secure User Setup
Security rules	Security	Secure Data Access
User privileges	Security	Secure Data Access
Privilege types	Security	Secure Data Access
Error detection / handling / report		
Distribution		RMI
Transaction management		

3. Cơ chế cài đặt

3.1. Security

3.1.1. Static view

3.1.1.1. Biểu đồ lớp



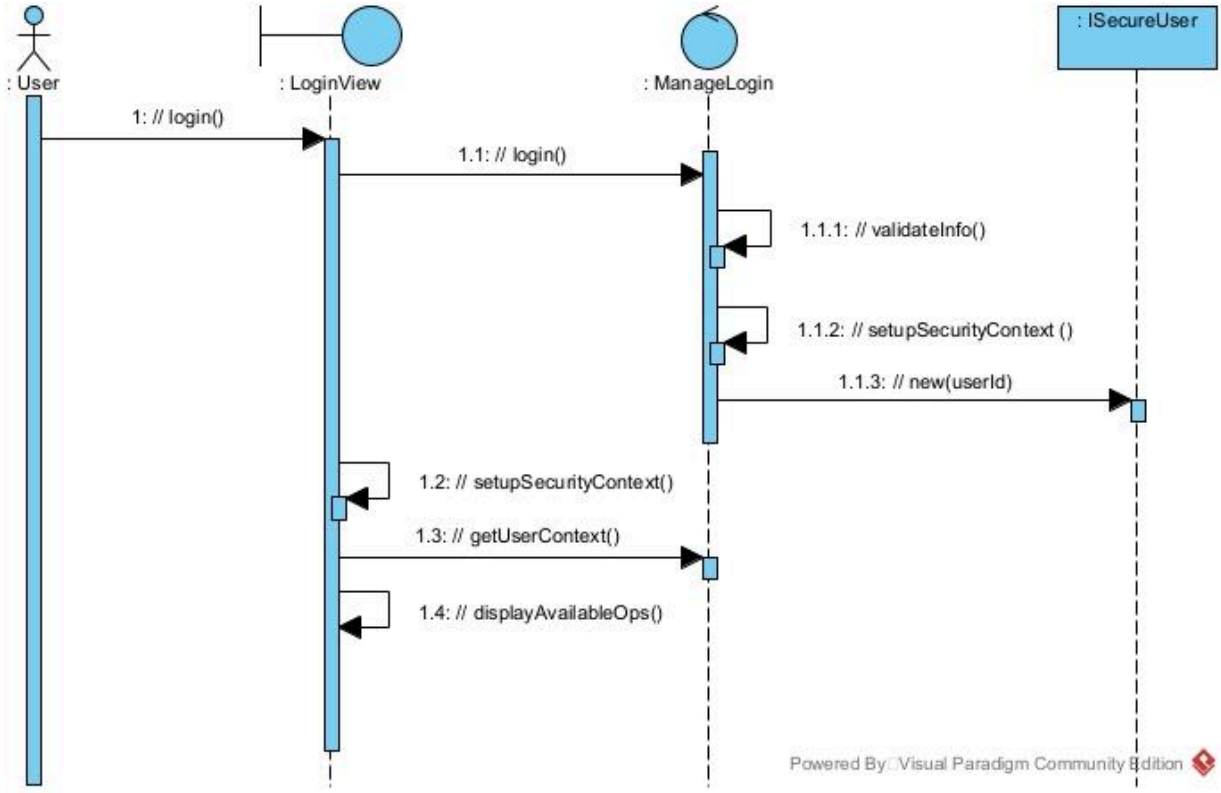
Ảnh 1.1. Biểu đồ lớp cơ chế security - static view.

3.1.1.2. Mô tả biểu đồ lớp

- **ISecureData**: Cơ chế phân tích: security
- **Security Access**: Cơ chế phân tích: security
- **UserSercurityContext**: Cơ chế phân tích: security
- **UniqueId**: Cơ chế phân tích: security

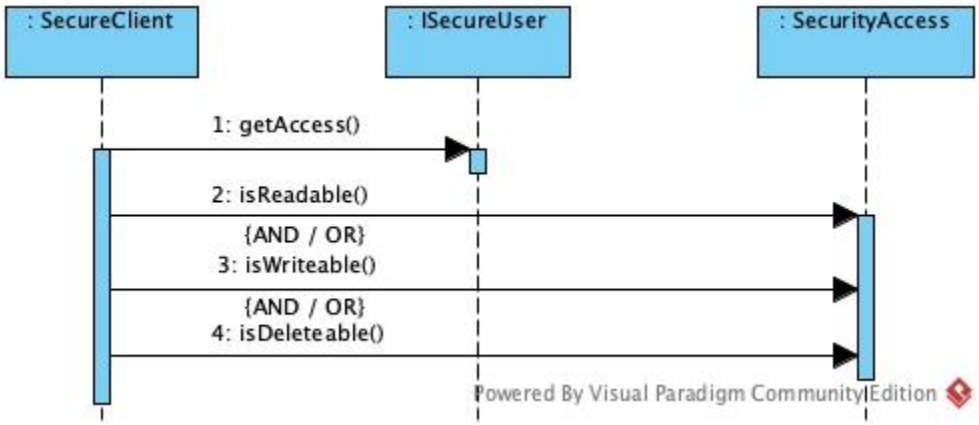
- **ISecureUser**: Cơ chế phân tích: security
- **ManageLogin**: Cơ chế phân tích: security

3.1.2. Dynamic view: Secure User Setup



Ảnh 1.2. Biểu đồ trình tự Secure User Setup.

3.1.3. Dynamic view: Secure Data Setup

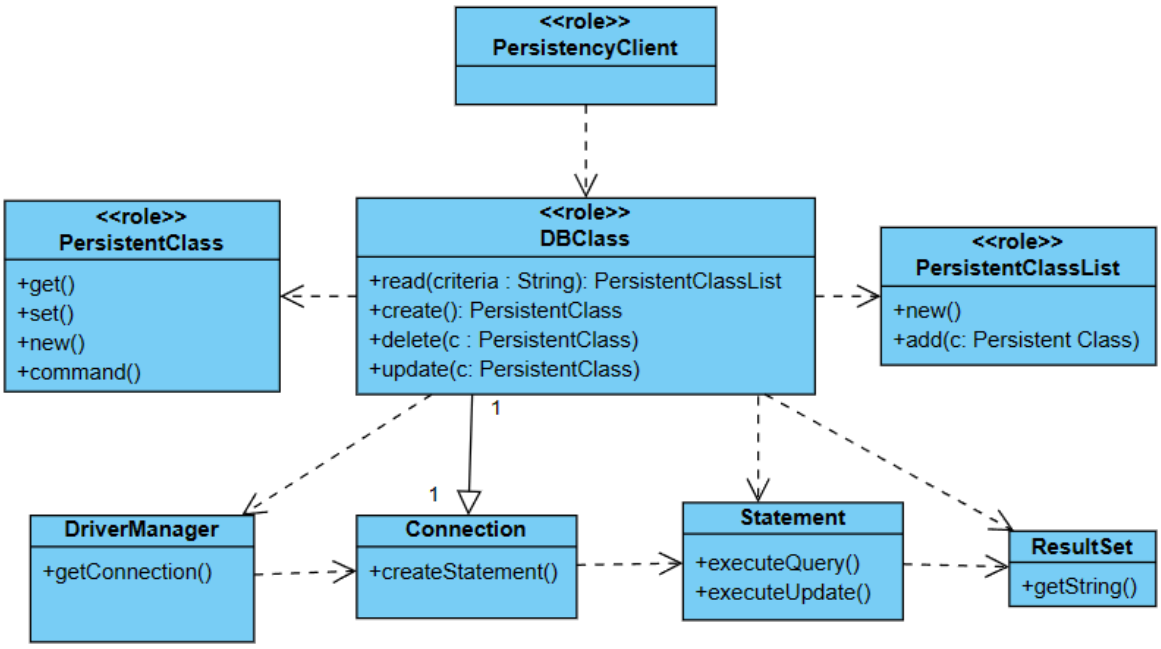


Ảnh 1.3. Biểu đồ trình tự Secure Data Access.

3.2. Persistency - RDBMS - JDBC

3.2.1. Static view

3.2.1.1. Biểu đồ lớp



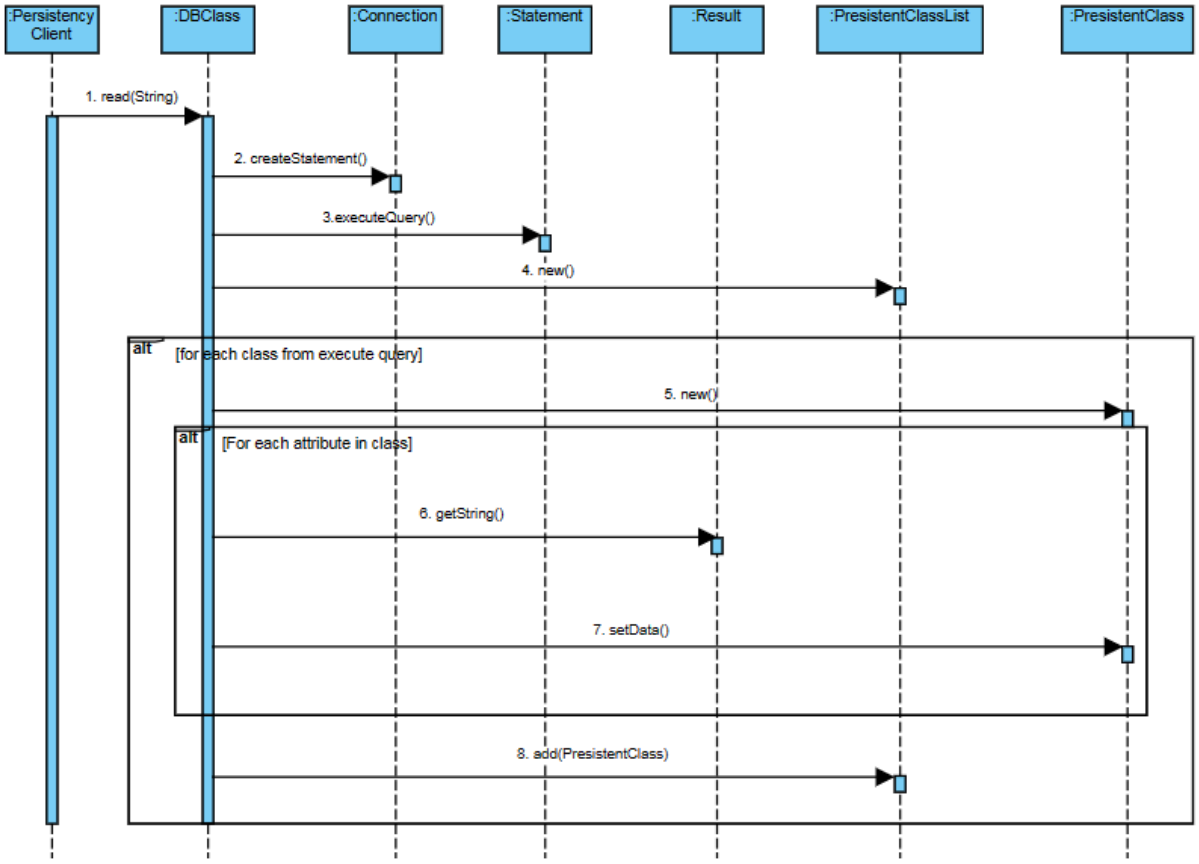
Ảnh 1.4. Biểu đồ lớp cơ chế persistency - static view.

3.2.1.2. Mô tả biểu đồ lớp

- **PersistencyClient**: Một ví dụ về một client của một lớp persistent.
- **PersistentClass**: Một ví dụ về một lớp có tồn tại trong ứng dụng.
- **PersistentClassList**: Một danh sách các đối tượng PersistentClass.
- **Statement**: Lớp được sử dụng để thực thi một câu lệnh SQL tĩnh và thu được kết quả do nó tạo ra. Các câu lệnh SQL không có tham số thường được thực thi bằng cách sử dụng các đối tượng Statement.
- **DBClass**: Một mẫu cho một lớp chịu trách nhiệm làm cho một lớp khác persistent. Mỗi lớp persistent sẽ có một DBClass tương ứng.

- **Connection**: Một kết nối (phiên) với cơ sở dữ liệu cụ thể. Trong ngữ cảnh là một kết nối, các câu lệnh SQL được thực thi và kết quả được trả về.
- **ResultSet**: Bộ kết quả cung cấp quyền truy cập vào một bảng dữ liệu. Đối tượng ResultSet thường được tạo bằng cách thực thi một Statement.
- **DriverManager**: Dịch vụ cơ bản để quản lý một bộ trình điều khiển JDBC.

3.2.2. Dynamic view: RDBMS Read



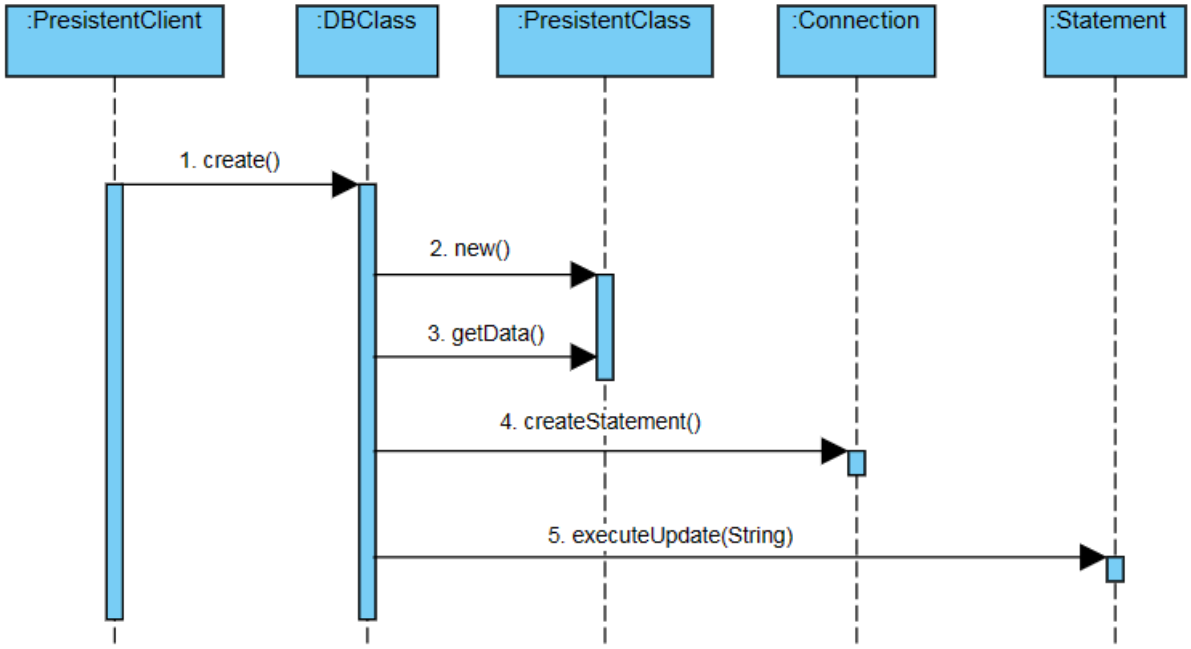
Ảnh 1.5. Biểu đồ trình tự JDBC RDBMS Read.

Biểu đồ này cho thấy việc đọc một đối tượng như thế nào.

Đầu tiên, SampleDBManager tạo một transaction chỉ đọc mới, sau đó tra cứu đối tượng bằng thao tác Map "get()". Khi đối tượng đã được tìm thấy, nó có thể được đọc bằng thao tác "getData()", transaction đã được thực hiện. RETAIN_HOLLOW được chỉ định, do đó, các tham chiếu đến đối tượng và dữ liệu được truy xuất có thể được sử dụng bên

ngoài transaction truy xuất. Sau khi transaction được thực hiện, đối tượng có thể được cập nhật.

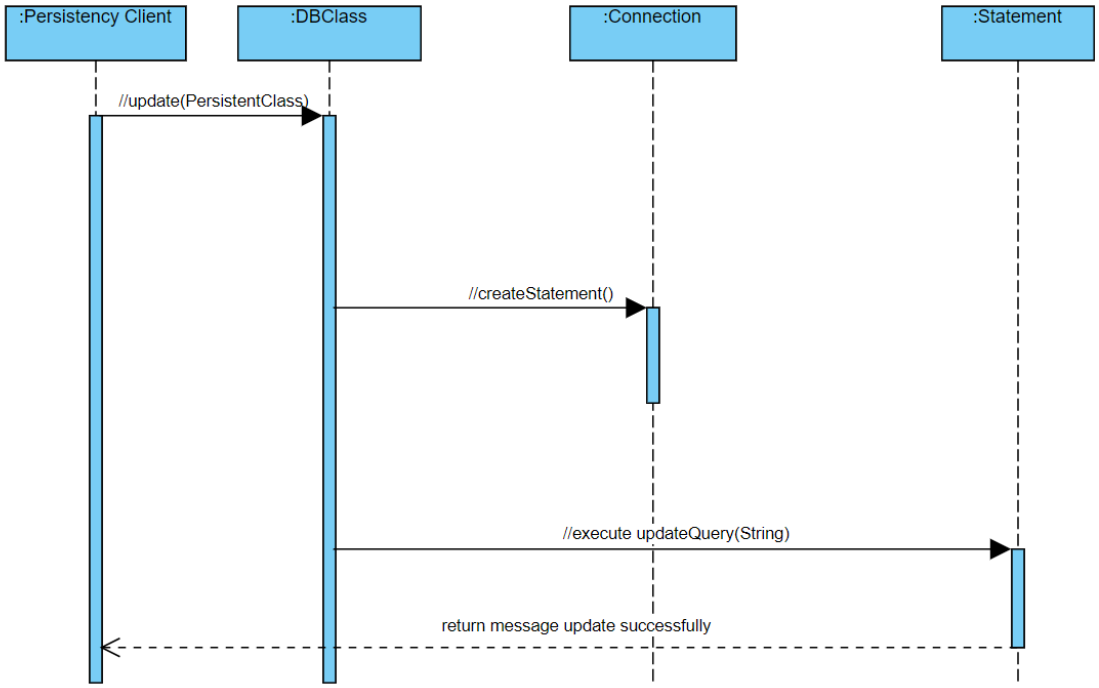
3.2.3. Dynamic view: RDBMS Create



Ảnh 1.6. Biểu đồ trình tự JDBC RDBMS Create

Biểu đồ này mô tả tạo một PersistentClass mới trong cơ sở dữ liệu như thế nào. Đầu tiên, SampleDBManager tạo một transaction và gọi constructor cho PersistentClass. Khi đã được tạo, lớp được thêm vào cơ sở dữ liệu thông qua "put()". Sau đó, transaction được thực hiện.

3.2.4. Dynamic view: RDBMS Update

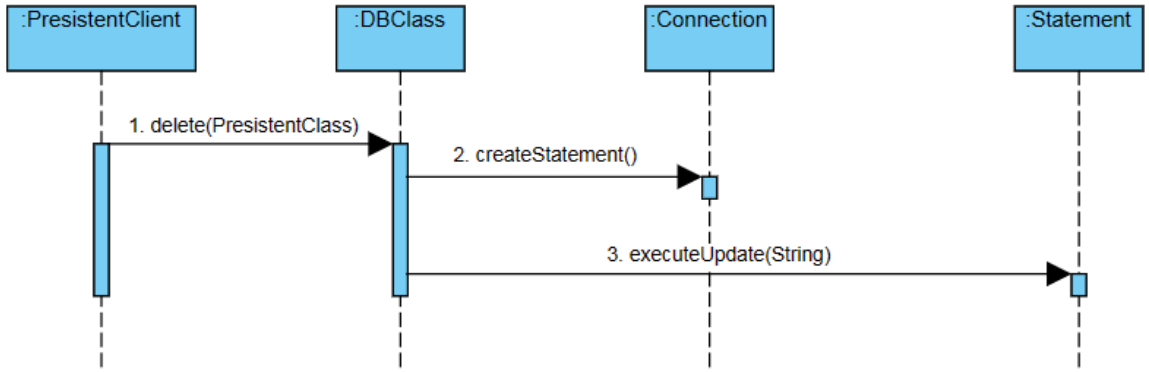


Ảnh 1.7. Biểu đồ trình tự JDBC RDBMS Update

Sơ đồ này cho thấy cách cập nhật một đối tượng.

Để cập nhật thông tin về một đối tượng cụ thể, client yêu cầu DBClass hỗ trợ. Sau đó, DBClass tạo một tập hợp các hướng dẫn bằng cách dùng createStatement() từ lớp Connection, chỉ định cách cập nhật thông tin trong cơ sở dữ liệu cho đối tượng đã cho. Sau đó, các hướng dẫn này được thực thi và dẫn đến việc cập nhật dữ liệu trong cơ sở dữ liệu.

3.2.5. Dynamic view: RDBMS Delete



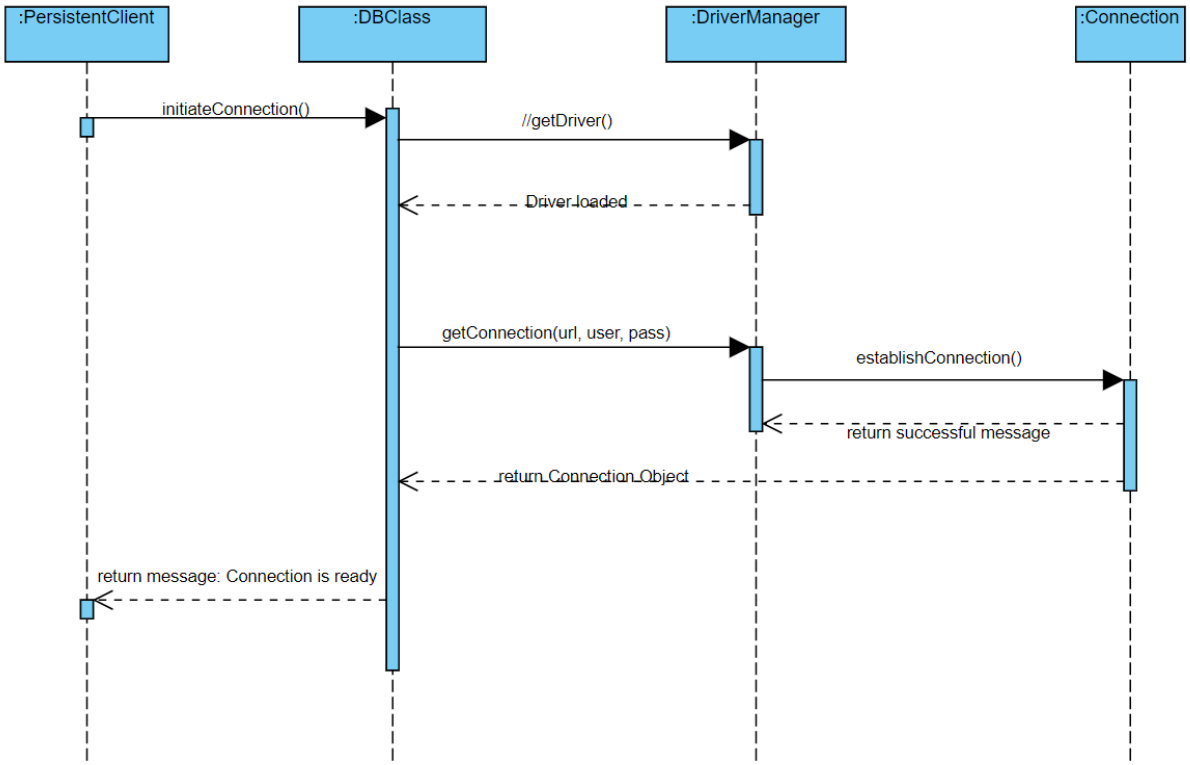
Ảnh 1.8. Biểu đồ trình tự JDBC RDBMS Delete

Biểu đồ này mô tả xóa một đối tượng khỏi cơ sở dữ liệu như thế nào.

Đầu tiên, SampleDBManager tạo một transaction mới, loại bỏ mọi phần cấu thành và sau đó loại bỏ đối tượng bằng cách sử dụng "remove()". Sau đó, đối tượng sẽ bị xóa hoàn toàn khỏi cơ sở dữ liệu ObjectStore ngay lập tức thông qua ObjectStore.destroy(). Khi đối tượng đã bị xóa, transaction được thực hiện.

Do đó, trong ObjectStore, việc xóa có hai bước - xóa khỏi lớp container là cơ sở dữ liệu trong bộ nhớ và xóa khỏi cơ sở dữ liệu vật lý. Đó là vì muốn việc xóa diễn ra ngay lập tức, trái ngược với bộ nhớ đệm.

3.2.6. Dynamic view: RDBMS Initialize



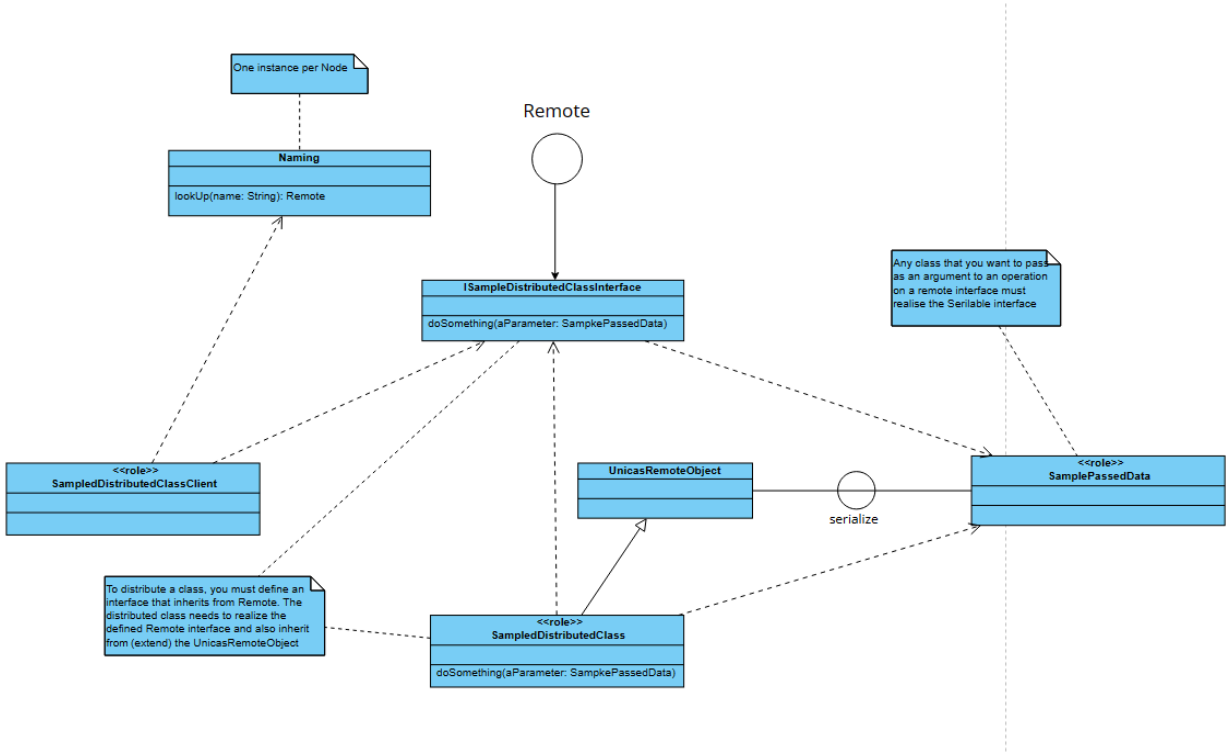
Ảnh 1.9. Biểu đồ trình tự JDBC RDBMS Initialize

Trình tự bắt đầu bằng việc PersistentClient yêu cầu DBClass thiết lập kết nối. DBClass tải JDBC driver cần thiết thông qua DriverManager.getDriver(). Tiếp theo, DBClass khởi tạo DriverManager để kết nối với cơ sở dữ liệu. Khi kết nối thành công, kích hoạt phần Connection. Kết nối đã tạo sau đó sẽ được đưa trở lại DBClass. Cuối cùng, DBClass thông báo cho PersistentClient rằng kết nối cơ sở dữ liệu đã sẵn sàng để sử dụng, hoàn thành trình tự và giải thích quy trình từng bước khởi tạo kết nối JDBC.

3.3. Distribution – RMI

3.3.1. Static view

3.3.1.1. Biểu đồ lớp



3.3.1.2. Mô tả biểu đồ lớp

Naming: Đây là cơ chế bootstrap để lấy tham chiếu tới các đối tượng từ xa dựa trên cú pháp Uniform Resource Locator (URL). URL cho một đối tượng từ xa được chỉ định bằng cách sử dụng tên máy chủ, cổng và tên thông thường:

- rmi://host:port/name
- host = tên host của registry (defaults to current host)
- port = port của registry (defaults to the registry port number)
- name = tên của đối tượng từ xa

SampleDistributedClass: Ví dụ về một lớp được phân tán.

Remote: Giao diện Remote dùng để xác định tất cả các đối tượng từ xa. Bất kỳ đối tượng nào là đối tượng từ xa đều phải trực tiếp hoặc gián tiếp

implements giao diện này. Chỉ những phương thức được chỉ định trong giao diện từ xa mới có sẵn từ xa.

Các lớp triển khai có thể implements bất kỳ số lượng giao diện từ xa nào và có thể mở rộng các lớp triển khai từ xa khác.

Đối với tất cả các lớp implements giao diện Remote, một stub từ xa và một skeleton từ xa được tạo. Các lớp này xử lý giao tiếp cần thiết để hỗ trợ phân tán.

SampleDistributedClassClient: Ví dụ về client của một lớp phân tán.

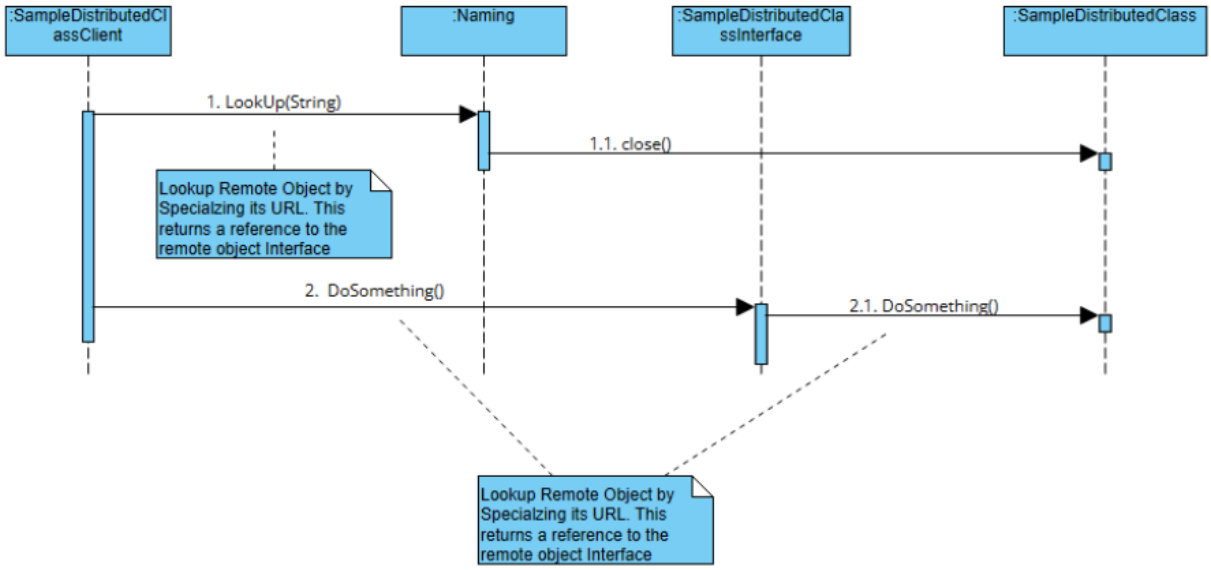
SamplePassedData: Ví dụ về dữ liệu được truyền đến / từ một lớp phân tán.

UnicastRemoteObject

ISampleDistributedClassInterface: Ví dụ về một giao diện được định nghĩa cho một lớp phân tán.

Serializable: Bất kỳ lớp nào bạn muốn truyền làm đối số cho một hoạt động trên giao diện từ xa đều phải implements giao diện Serializable.

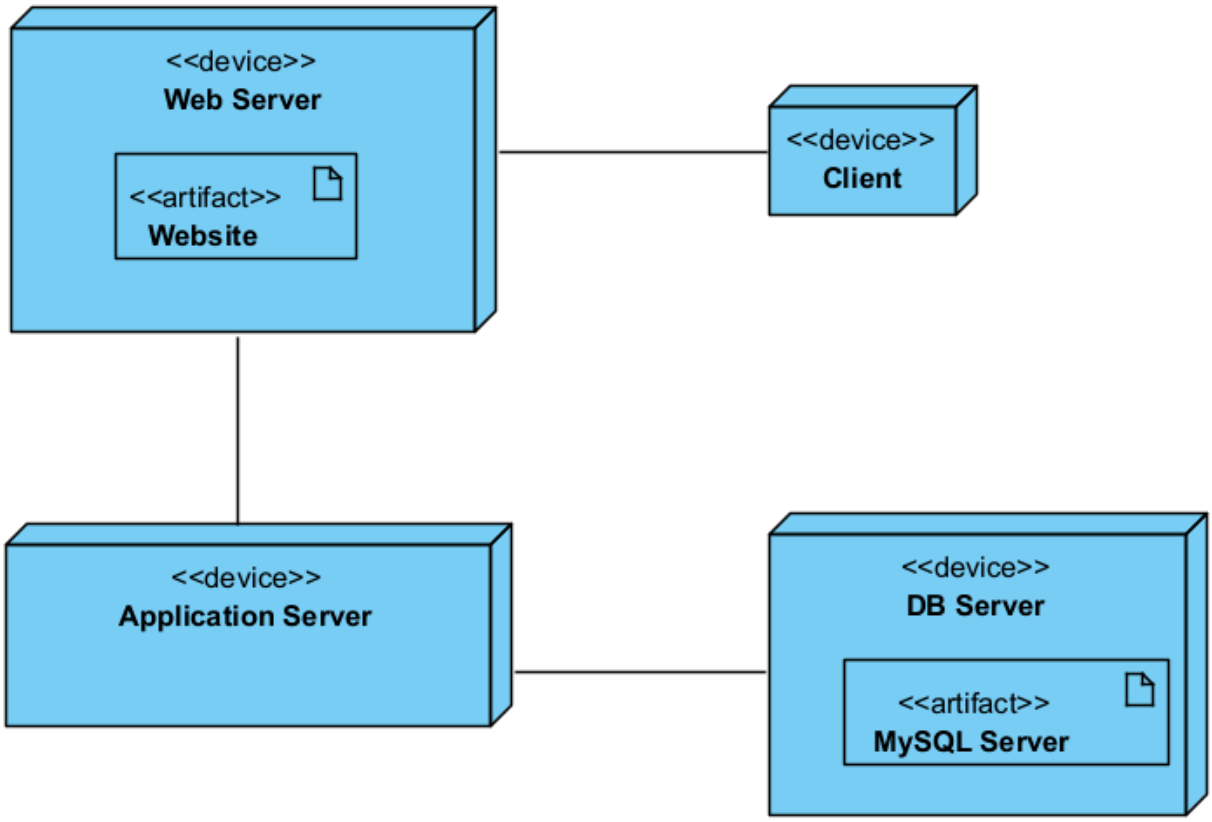
3.3.2. Dynamic view



4. Khung nhìn triển khai

Hệ thống được triển khai trên 3 Server khác nhau gồm Web Server, Application Server, DB Server và thiết bị đầu cuối.

- Web Server: Nơi lưu trữ Website dịch vụ.
- Application Server: Nơi xử lý các yêu cầu từ người sử dụng.
- DB Server: Nơi lưu trữ dữ liệu.
- Client: Thiết bị có thể chạy trình duyệt Web như PC, điện thoại và kết nối với Web Server thông qua Internet.



Cấu trúc phần cứng của hệ thống bao gồm máy chủ để lưu trữ và xử lý dữ liệu và các thiết bị người dùng như điện thoại thông minh, máy tính bảng hoặc máy tính để bàn để cung cấp giao diện người dùng. Máy chủ sẽ được đặt tại một trung tâm dữ liệu (data center) để đảm bảo tính an toàn, tin cậy và mở rộng của hệ thống.

Phần mềm của hệ thống được phát triển, khai thác trên nền tảng điện toán đám mây để đảm bảo tính linh hoạt và khả năng mở rộng.

Phần mềm của hệ thống được viết bằng ngôn ngữ lập trình Java và các Framework phổ biến như Spring MVC, Hibernate.

Để đảm bảo tính toàn vẹn và bảo mật của dữ liệu, hệ thống sử dụng các phương thức bảo mật như mã hóa SSL (Secure Sockets Layer) để mã hóa dữ liệu giao tiếp và JWT (JSON Web Token) để xác thực người dùng.

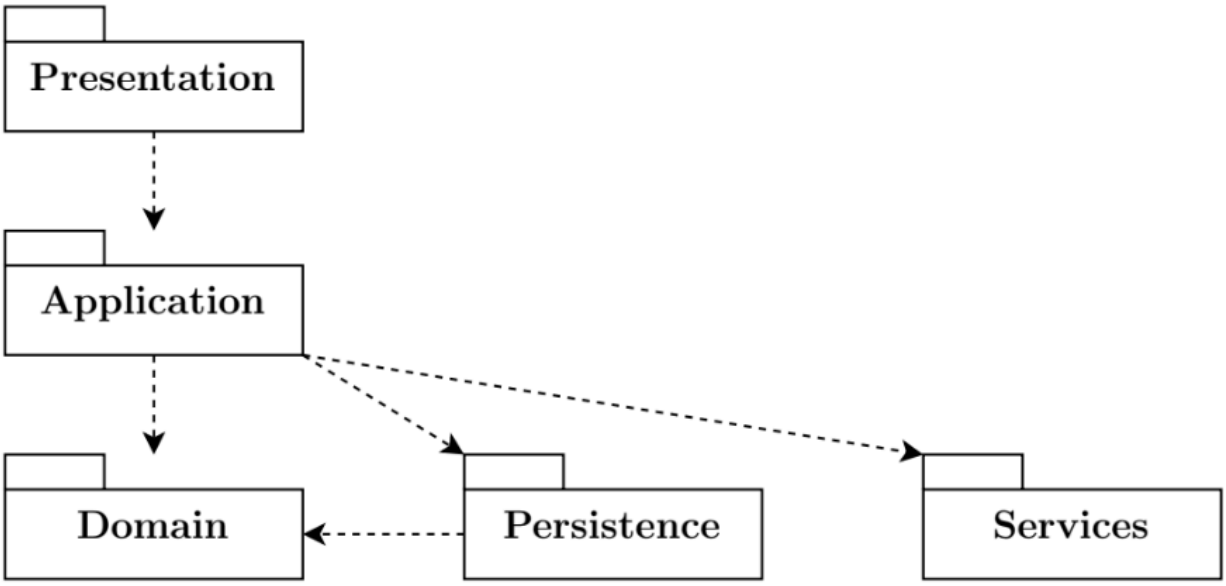
5. Góc nhìn logic

5.1. Tổng quan

Đây là một mô tả logic về kiến trúc của hệ thống. Mô tả những lớp quan trọng và cách tổ chức những gói dịch vụ và hệ thống con, và cách tổ chức những hệ thống con này vào các lớp. Trong phần này cũng mô tả những ca sử dụng quan trọng nhất, ví dụ những khía cạnh linh hoạt của kiến trúc hệ thống. Biểu đồ lớp có thể được đưa vào để minh họa các mối quan hệ giữa kiến trúc quan trọng các lớp, hệ thống con, gói và lớp.

Khung nhìn logic của hệ thống bao gồm 5 gói:

- **Giao diện** (Presentation): chứa các lớp cho mỗi biểu mẫu mà các tác nhân sử dụng để giao tiếp với Hệ thống.
- **Ứng dụng** (Application): chứa các lớp xử lý chính cho hệ thống.
- **Miền** (Domain): chứa các gói hỗ trợ các thực thể chính của hệ thống.
- **Nhất quán** (Persistence): chứa các lớp để đảm bảo tính nhất quán của dữ liệu.
- **Dịch vụ** (Services): chứa các lớp để cung cấp các lớp hệ thống cho mục đích bảo trì.

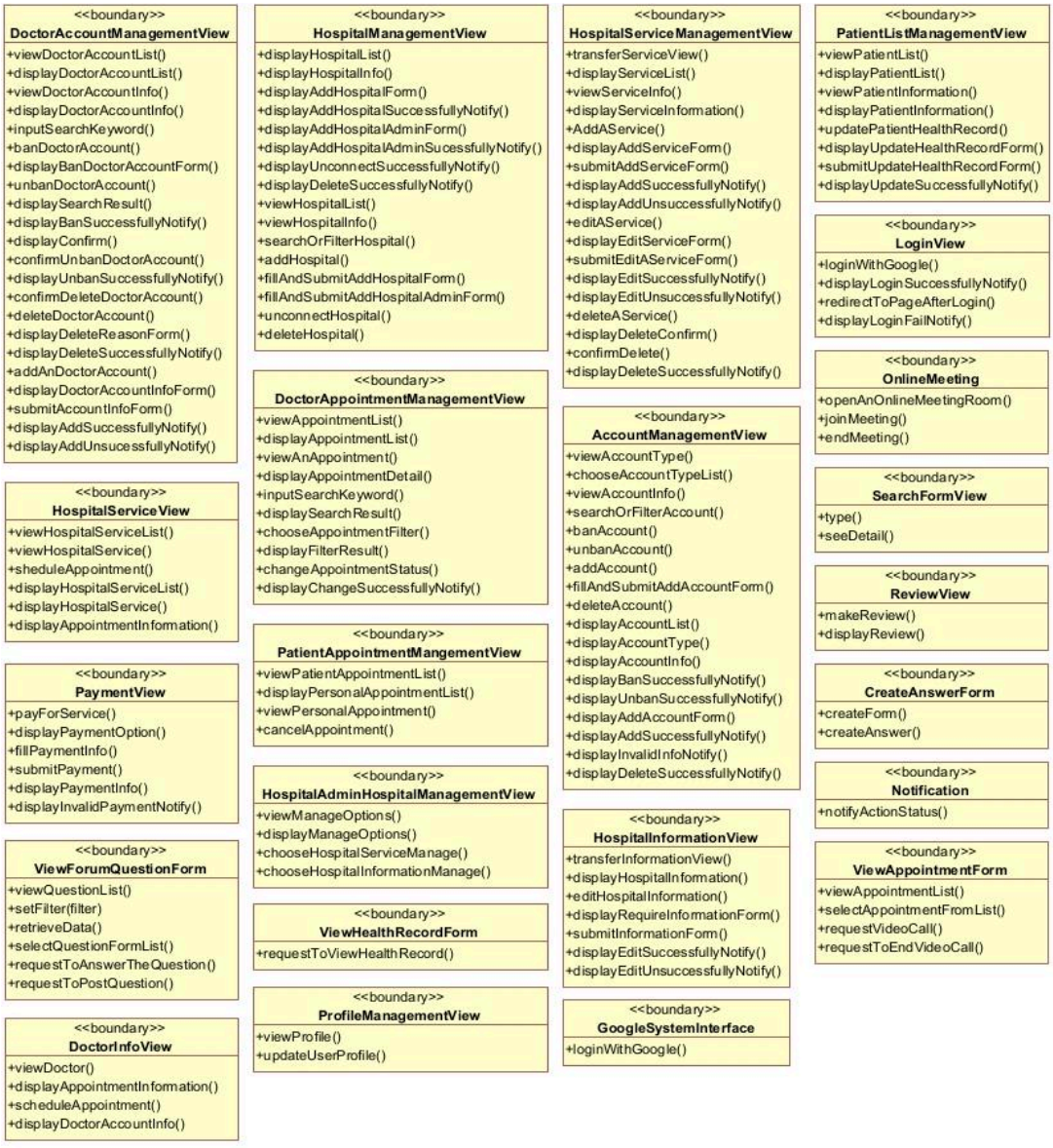


5.2. Các gói thiết kế kiến trúc

5.2.1. Gói giao diện

Mô tả ngắn gọn: Trong gói này chứa các lớp cho mỗi mẫu mà các tác nhân sử dụng để giao tiếp với hệ thống. Các lớp ranh giới tồn tại để hỗ trợ duy trì xác thực, quản lý các appointment, forum hỏi đáp, tìm kiếm bác sĩ, dịch vụ, triển khai việc khám trực tuyến.

Biểu đồ:



5.2.2. Gói ứng dụng

Mô tả ngắn gọn: Gói này chứa các lớp cho các logic nghiệp vụ trong hệ thống. Các lớp điều khiển tồn tại để hỗ trợ duy trì xác thực, quản lý quản lý các appointment, forum hỏi đáp, tìm kiếm bác sĩ, dịch vụ, triển khai việc khám trực tuyến.

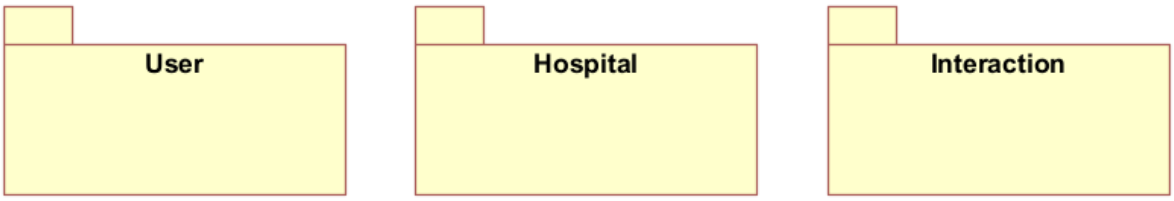
Biểu đồ:

5.2.3. Gói miền

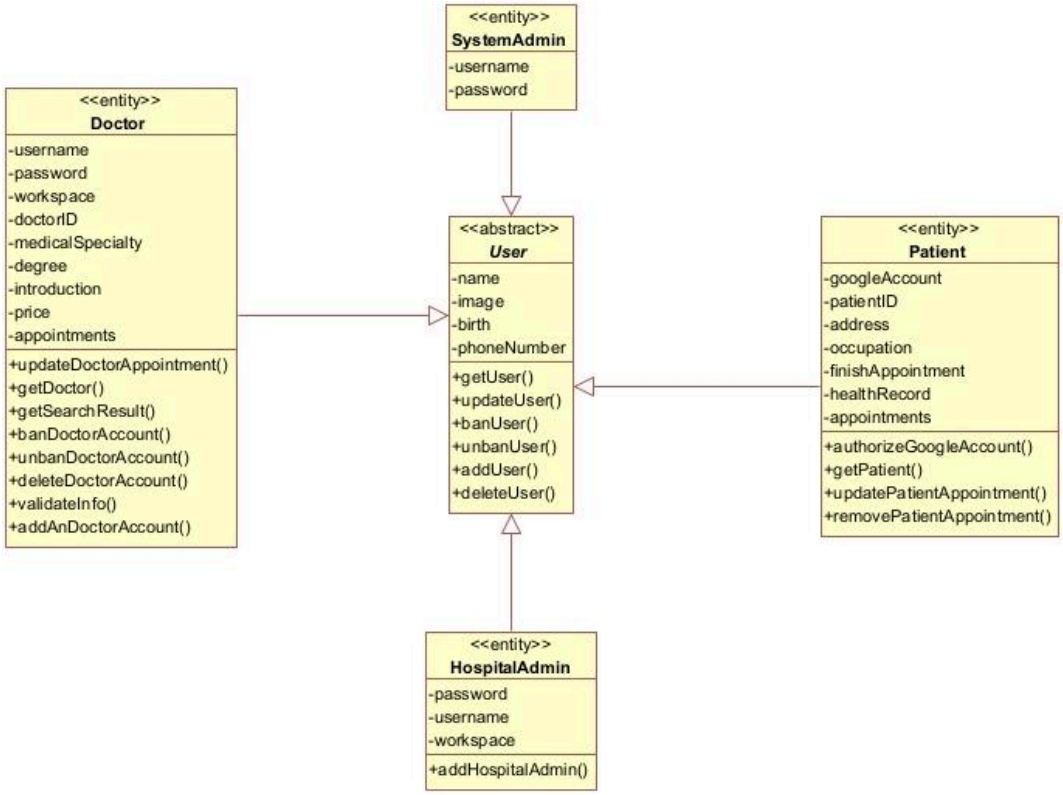
Mô tả ngắn gọn: Gói này chứa các gói chứa các lớp để hỗ trợ các thực thể người dùng, appointment, forum hỏi đáp. Gói miền bao gồm 3 gói con sau đây:

- **Gói người dùng**: chứa tất cả các lớp phục vụ quản lý các người dùng.
- **Gói hospital**: chứa các yêu cầu, dịch vụ lớp để hỗ trợ những nhiệm vụ liên quan đến bác sĩ, bệnh viện.

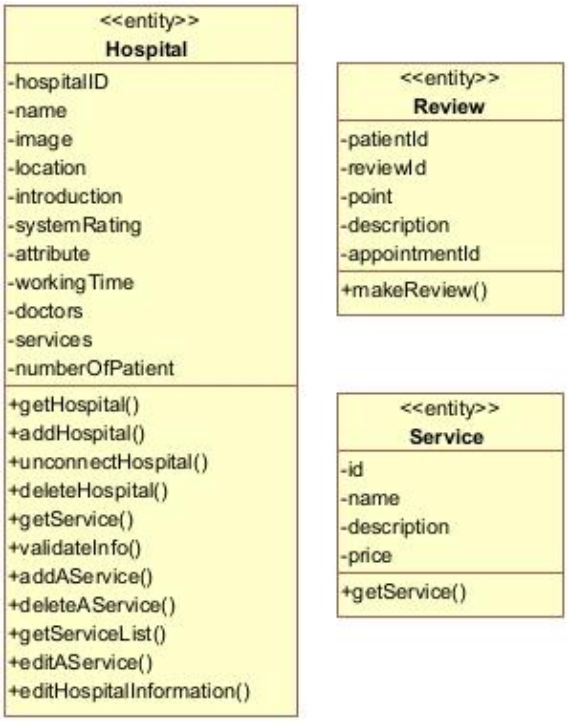
- **Gói interaction**: chứa tất cả lớp phục vụ việc tương tác giữa các người dùng.



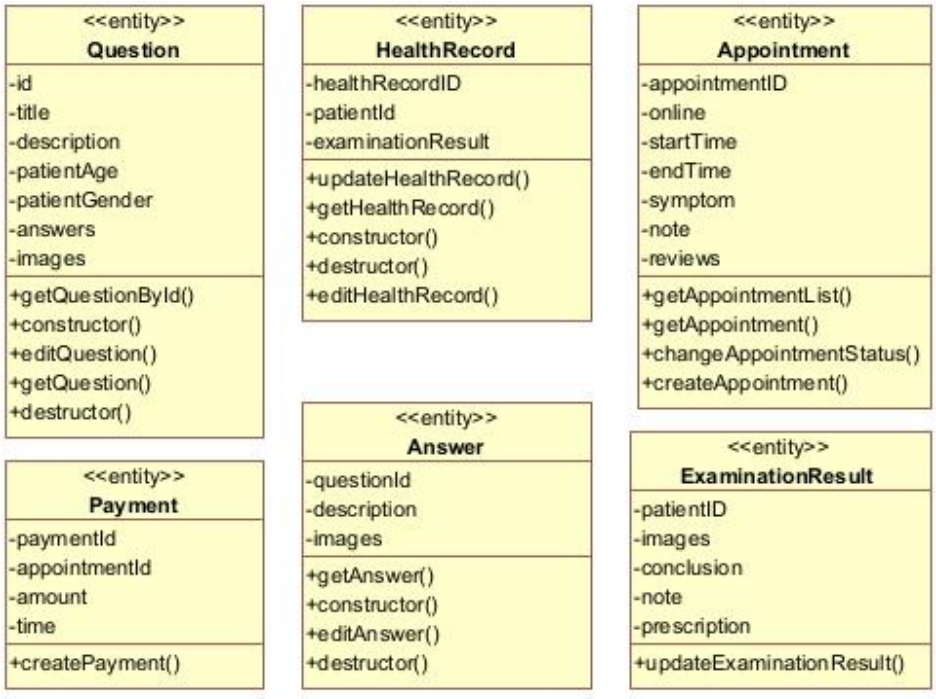
Gói người dùng



Gói Hospital



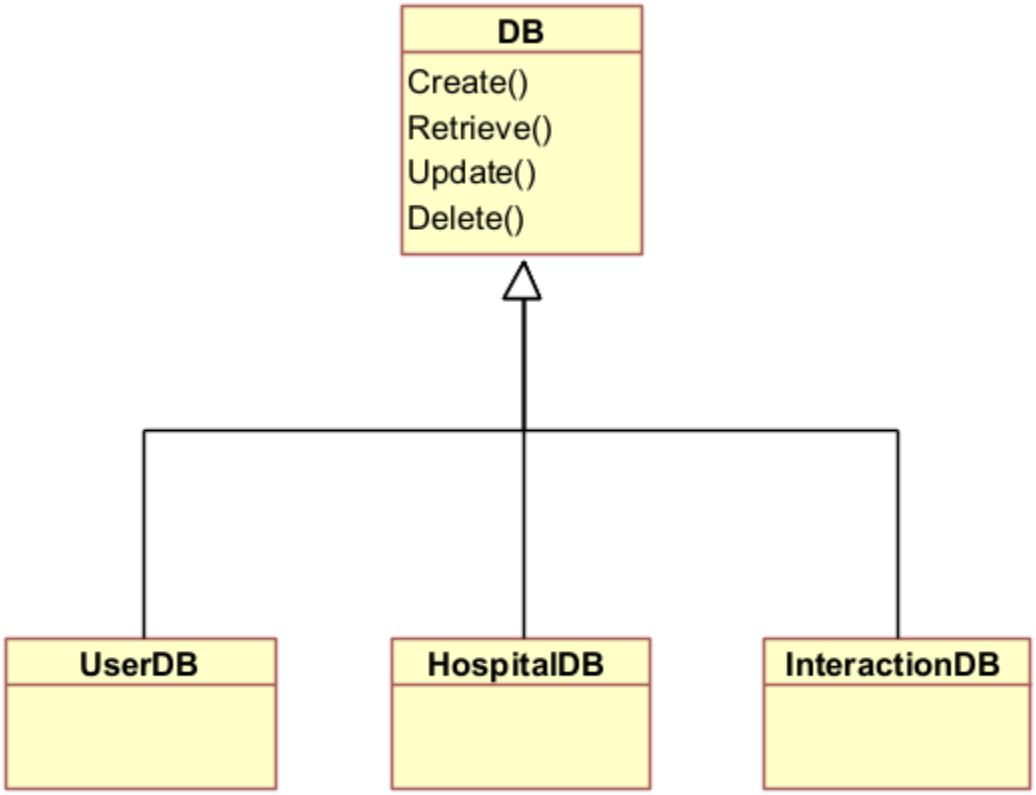
Gói Interaction



5.2.4. Gói nhất quán

Mô tả ngắn gọn: Gói này chứa các gói dữ liệu để đảm bảo tính nhất quán của dữ liệu. Bốn toán tử: thêm, sửa, xóa, cập nhật là bốn chức năng chính được thực hiện trong các ứng dụng cơ sở dữ liệu.

Biểu đồ:



6. Góc nhìn tiến trình

6.1. Mô hình tiến trình

Phần này sẽ mô tả sự phân hóa của hệ thống thành các quá trình nhẹ (các luồng điều khiển đơn) và các quy trình nặng (nhóm của các quy trình nhẹ), tổ chức các phần này theo nhóm các tiến trình giao giao tiếp hoặc tương tác. Mô tả các chế độ giao tiếp chính giữa các quá trình, chẳng hạn như chuyển tiếp tin nhắn, gián đoạn.

6.2. Mô tả các phần tử tiến trình

Tiến trình Quản lý người dùng: Đây là tiến trình quản lý thông tin người dùng và các hoạt động của họ trên ứng dụng.

Tiến trình này bao gồm các bước sau:

- Nhận yêu cầu đăng ký, đăng nhập hoặc cập nhật thông tin người dùng từ giao diện người dùng.
- Xử lý yêu cầu và truy vấn cơ sở dữ liệu để lưu trữ hoặc cập nhật thông tin người dùng.
- Kiểm tra tính hợp lệ của thông tin người dùng và xử lý các lỗi khi cần thiết.

Tiến trình thanh toán

Tiến trình này bao gồm các bước sau:

- Bệnh nhân chọn bác sĩ mà mình muốn được khám, sau đó điền/chọn các thông tin cần thiết và bấm "Đặt khám".
- Bệnh nhân thanh toán thông qua các phương thức thanh toán qua ví điện tử.
- Hệ thống xử lý thanh toán, kiểm tra tính hợp lệ của thông tin thanh toán và cập nhật trạng thái thanh toán.
- Hệ thống gửi thông tin thanh toán thành công hoặc không thành công cho người dùng.

Tiến trình đánh giá bệnh viện/dịch vụ khám/bác sĩ cho phép bệnh nhân đánh giá các bệnh viện/dịch vụ khám/bác sĩ mà họ đã trải nghiệm, đồng thời cung cấp thông tin đánh giá cho người dùng khác tham khảo.

Tiến trình này bao gồm các bước sau:

- Khởi động tiến trình: Bệnh nhân truy cập vào ứng dụng Chăm sóc sức khỏe trực tuyến, sau đó truy cập trang đánh giá sau khám. Bệnh nhân chọn một lịch khám chưa được đánh giá và truy cập chức năng đánh giá.

- Hiển thị giao diện đánh giá: Hệ thống hiển thị giao diện đánh giá cho bệnh nhân nhập thông tin về đánh giá, bao gồm điểm đánh giá (thang 5), và phản hồi chi tiết.
- Nhập thông tin đánh giá: Bệnh nhân nhập thông tin đánh giá vào các trường tương ứng trên giao diện đánh giá.
- Lưu thông tin đánh giá: Hệ thống lưu thông tin đánh giá vào cơ sở dữ liệu của ứng dụng.
- Xử lý thông tin đánh giá: Hệ thống sử dụng các thuật toán đánh giá để tính toán điểm đánh giá trung bình cho bệnh viện/dịch vụ khám/bác sĩ dựa trên đánh giá của bệnh nhân.
- Cập nhật thông tin đánh giá: Hệ thống cập nhật điểm đánh giá trung bình và số lượng đánh giá cho bệnh viện/dịch vụ khám/bác sĩ vào cơ sở dữ liệu của ứng dụng.
- Hiển thị thông tin đánh giá: Hệ thống hiển thị thông tin đánh giá của bệnh viện/dịch vụ khám/bác sĩ cho người dùng khác tham khảo trên trang chi tiết bệnh viện/dịch vụ khám/bác sĩ.

Kết thúc quá trình: Tiến trình đánh giá bệnh viện/dịch vụ khám/bác sĩ kết thúc và quay trở lại giao diện hồ sơ của bệnh viện/dịch vụ khám/bác sĩ vừa đánh giá.