**BÁO CÁO**

**PHÁT TRIỂN HỆ THỐNG THÔNG MINH**

NHÓM: 05

**Đề tài 16: “Hệ thống hỗ trợ giao tiếp với người câm điếc”**

**Nhóm BTL: 11**

Thành viên :

* **Nguyễn Khánh Nam – B20DCCN454**
* Trần Thanh Tuấn – B20DCCN620
* Nguyễn Hoàng Việt – B20DCCN728
* Đàm Trọng Ngọc Hà – B20DCCN211

**MỤC LỤC**

**Phần A: Thiết kế hậu cần server phần quản lý mẫu cho bài toán số 2 3**

1. Mô tả cho bài toán số 2 3
2. Thiết kế CSDL cho bài toán số 2 3
3. Thiết kế lớp thực thể cho quản lý mẫu cho bài toán số 2 4
4. Thiết kế các giao diện chung cho quản lý mẫu bài toán số 2 4
5. Thiết kế biểu đồ lớp chi tiết 5

Chức năng 1: Tìm mẫu cho bài toán số 2 10

Chức năng 2: Thêm mẫu cho bài toán số 2 14

Chức năng 3: Sửa mẫu cho bài toán số 2 18

Chức năng 4: Xóa mẫu cho bài toán số 2 23

**Phần A: Thiết kế hậu cần server phần quản lý mẫu cho bài toán số 2**

1. Mô tả bài toán số 2

* Đầu vào bài toán là khung ảnh chứa hình ảnh bàn tay
* Đầu ra của bài toán là thông điệp của khung ảnh đó

1. Thiết kế CSDL cho quản lý mẫu cho bài toán số 2

A diagram of a computer code

Description automatically generated with medium confidence

Mỗi thực thể đề xuất một bảng tương ứng :

* User -> User
* Sample -> Sample
* Label -> Label

Đưa thuộc tính không phải đối tượng của lớp thực thể thành thuộc tính của bảng tương ứng:

* User: id, name, username, password
* Sample: id, name, valiDate, link
* Label: id, name, createDate, status

Chuyển quan hệ số lượng giữa các lớp thực thể thành quan hệ số lượng giữa các bảng :

* n Sample – 1 Label

Bổ sung các thuộc tính khóa. Khóa chính được thiết lập với thuộc tính id của các bảng tương ứng. Khóa ngoại được thiết lập cho các bảng :

* Sample: labelId

1. Thiết kế lớp thực thể cho quản lý mẫu cho bài toán số 2

A diagram of a username and password

Description automatically generated

Trích lớp thực thể :

* Trích các danh từ xuất hiện :

+ Sample : lớp

+ Label : lớp

+ User : lớp

+ Hệ thống: trừu tượng -> loại

+ Tên : thuộc tính

+ Mật khẩu : thuộc tính

* Xét quan hệ giữa các lớp

+ Một nhãn có thể có nhiều mẫu, nhiều mẫu sẽ được gán cho một nhãn -> mối quan hệ giữa Sample – Label là n – 1, Label quan hệ lỏng với Sample

1. Thiết kế các giao diện chung cho quản lý mẫu bài toán số 2

A login screen with blue and black text

Description automatically generated

*Giao diện đăng nhập*

A screenshot of a computer

Description automatically generated

*Giao diện trang chủ*

1. Thiết kế biểu đồ lớp chi tiết

A diagram of a computer program

Description automatically generated with medium confidence

Hàm login() : khởi tạo giao diện login.html

* Đầu vào : đối tượng dạng User, model dạng Model, request dạng HttpServletRequest
* Đầu ra : trỏ tới lớp view “login.html”
* Gán vào lớp UserController

Hàm logout() : xử lý đăng xuất

* Đầu vào : session dạng HttpSession
* Đầu ra : trỏ tới lớp view “login.html”
* Gán vào lớp UserController

Hàm getHome() : khởi tạo giao diện home.html

* Đầu vào : session dạng HttpSession
* Đầu ra : trỏ tới lớp view “home.html”
* Gán vào lớp UserController

Hàm getUserByUsernameAndPassword() : lấy ra đối tượng có username và password cung cấp

* Đầu vào : username và password dạng String
* Đầu ra : đối tượng dạng User
* Gán vào lớp UserService

Hàm findByUserNameAndPassword() : lấy ra đối tượng có username và password cung cấp

* Đầu vào : username và password dạng String
* Đầu ra : đối tượng dạng User
* Gán vào lớp UserRepo

Hàm getAllSample() : khởi tạo lớp view “SampleAdd.html”

* Đầu vào : model dạng Model, session dạng HttpSession
* Đầu ra : trỏ tới lớp view “SampleAdd.html”
* Gán vào lớp SampleController

Hàm addSample() : khởi tạo lớp view “SampleAll.html”

* Đầu vào : model dạng Model, session dạng HttpSession
* Đầu ra : trỏ tới lớp view “SampleAll.html”
* Gán vào lớp SampleController

Hàm save() : thực hiện lưu mẫu mới

* Đầu vào : labelId dạng int, model dạng Model, session dạng HttpSession
* Đầu ra : trỏ tới lớp view “SampleAll.html”
* Gán vào lớp SampleController

Hàm editSample() : khởi tạo lớp view “SampleEdit.html”

* Đầu vào : id dạng int, model dạng Model, session dạng HttpSession
* Đầu ra : trỏ tới lớp view “SampleEdit.html”
* Gán vào lớp SampleController

Hàm update() : thực hiện lưu mẫu sau khi cập nhật

* Đầu vào : sample dạng Sample, label dạng Label, session dạng HttpSession
* Đầu ra : trỏ tới lớp view “SampleAll.html”
* Gán vào lớp SampleController

Hàm deleteSample() : khởi tạo lớp view “SampleDelete.html”

* Đầu vào : session dạng HttpSession
* Đầu ra : trỏ tới lớp view “SampleDeleteConfirm.html”
* Gán vào lớp SampleController

Hàm deleteSampleForm() : thực hiện xóa theo yêu cầu của user

* Đầu vào : id dạng int, model dạng Model, session dạng HttpSession
* Đầu ra : trỏ tới lớp “SampleAll.html”
* Gán vào lớp SampleController

Hàm findSampleByName() : khởi tạo giao diện “SampleFind.html”

* Đầu vào : name dạng String, model dạng Model, session dạng HttpSession
* Đầu ra : trỏ tới lớp view “SampleFind.html”
* Gán vào lớp SampleController

Hàm getAll() : lấy ra tất cả mẫu

* Đầu vào : không có
* Đầu ra : danh sách đối tượng dạng Sample
* Gán vào lớp SampleService

Hàm getSampleById(): lấy ra mẫu theo id

* Đầu vào : id dạng int
* Đầu ra : đối tượng dạng Sample
* Gán vào lớp SampleService

Hàm saveSample() : lưu mẫu

* Đầu vào : đối tượng dạng Sample
* Đầu ra : boolean
* Gán vào lớp SampleService

Hàm getListSampleByName() : lấy ra danh sách mẫu theo tên

* Đầu vào : name dạng String
* Đầu ra : boolean
* Gán vào lớp SampleService

Hàm deleteSample() : thực hiện xóa mẫu

* Đầu vào : id dạng int
* Đầu ra : boolean
* Gán vào lớp SampleService

Hàm findById() : lấy mẫu theo id

* Đầu vào : id dạng int
* Đầu ra : đối tượng dạng Sample
* Gán cho lớp SampleRepo

Hàm findAll() : lấy ra tất cả mẫu

* Đầu vào : không có
* Đầu ra : danh sách đối tượng dạng Sample
* Gán cho lớp SampleRepo

Hàm getAll (): lấy ra danh sách đối tượng dạng Label

* Đầu vào : không có
* Đầu ra : danh sách đối tượng dạng Label
* Gán cho lớp LabelService

Hàm getLabelByNameId() : lấy ra tên nhãn theo id

* Đầu vào : id dạng int
* Đầu ra : tên của nhãn
* Gán cho lớp LabelService

Hàm getLabelByName() : lấy ra đối tượng dạng Label theo tên

* Đầu vào : name dạng String
* Đầu ra : đối tượng dạng Label
* Gán cho lớp LabelService

Hàm findById() : lấy nhãn theo id

* Đầu vào : id dạng int
* Đầu ra : đối tượng dạng Label
* Gán cho lớp LabelRepo

Hàm findAll() : lấy ra tất cả nhãn

* Đầu vào : không có
* Đầu ra : danh sách đối tượng dạng Label
* Gán cho lớp LabelRepo

Hàm findByName() : lấy ra nhãn theo tên

* Đầu vào : name dạng String
* Đầu ra : đối tượng dạng Label
* Gán cho lớp LabelRepo

Chức năng 1: Tìm mẫu cho bài toán số 2

1. Thiết kế biểu đồ hoạt động tuần tự

A diagram of a project

Description automatically generated

Kịch bản cho biểu đồ tuần tự trên như sau

1. Người dùng nhập thông tin đăng nhập trong trang Login.html và nhấn nút Login
2. Login.html gọi đến lớp UserController.
3. Lớp UserController gọi đến lớp User
4. Lớp User thực hiện hàm đóng gói User()/setter()
5. Lớp User trả về lớp UserController đối tượng User
6. Lớp UserController gọi đến lớp UserService
7. Lớp UserService thực hiện hàm getUserByUsernameAndPassword()
8. Lớp UserService gọi đến lớp UserRepo
9. Lớp UserRepo thực hiện hàm setId()/setName()
10. Lớp UserRepo trả về lớp UserService đối tượng User
11. Lớp UserService trả về lớp UserController đối tượng User
12. Lớp UserController gọi tới giao diện Home.html
13. Giao diện Home.html hiển thị cho người dùng
14. Người dùng nhấn nút Sample Home
15. Giao diện Home.html gọi đến lớp SampleController
16. Lớp SampleController thực hiện hàm getAllSample()
17. Lớp SampleController gọi đến lớp SampleService
18. Lớp SampleService thực hiện hàm getAll()
19. Lớp SampleService gọi đến lớp SampleRepo
20. Lớp SampleRepo thực hiện hàm findAll()
21. Lớp SampleRepo gọi tới lớp Sample
22. Lớp Sample thực hiện hàm getter()
23. Lớp Sample trả về cho lớp SampleRepo đối tượng Sample
24. Lớp SampleRepo trả về cho lớp SampleService danh sách đối tượng Sample
25. Lớp SampleService trả về cho lớp SampleController danh sách đối tượng dạng Sample
26. Lớp SampleController gọi tới giao diện SampleAll.html
27. Giao diện SampleAll.html hiển thị cho người dùng
28. Người dùng nhập tên mẫu cần tìm và nhấn nút Find
29. Giao diện SampleAll.html gọi tới lớp SampleController
30. Lớp SampleController thực hiện hàm findSampleByName()
31. Lớp SampleController gọi tới lớp SampleService
32. Lớp SampleService thực hiện hàm getListSampleByName()
33. Lớp SampleService gọi tới lớp SampleRepo
34. Lớp SampleRepo thực hiện hàm findByName()
35. Lớp SampleRepo gọi tới lớp Sample
36. Lớp Sample thực hiện hàm getter()
37. Lớp Sample trả về cho lớp SampleRepo đối tượng Sample
38. Lớp SampleRepo trả về cho lớp SampleService danh sách đối tượng Sample
39. Lớp SampleService trả về cho lớp SampleController danh sách đối tượng dạng Sample
40. Lớp SampleController gọi tới giao diện SampleFind.html
41. Giao diện SampleFind.html hiển thị cho người dùng
42. Kết quả

A screenshot of a computer screen

Description automatically generated

*Giao diện trang chủ với chức năng tìm mẫu*

Đầu tiên, người dùng nhập tên mẫu cần tìm vào ô trống. Sau đó nhấn nút Find. Nút Find được gán cho 1 đường link “@{/sample/find}”. SampleController sẽ lấy giá trị mà người dùng đã nhập ra, đổi nó sang dạng String và sử dụng SampleService để tìm kiếm. SampleService thực hiện hàm tìm kiếm getListSampleByName() và trả về 1 danh sách các mẫu có tên như yêu cầu tìm kiếm. Sau đó SampleController đưa các giá trị đã xử lý vào trong modelAttribute. SampleController gọi tới lớp view “SampleFind.html”

Sau đây là kết quả tìm kiếm mẫu với tên “a”. Tìm được 40 kết quả và hiển thị ra màn hình.

A screenshot of a computer

Description automatically generated

*Giao diện hiển thị kết quả tìm kiếm*

Chức năng 2: Thêm mẫu cho bài toán số 2

1. Thiết kế biểu đồ hoạt động tuần tự

A diagram of a graph

Description automatically generated

Kịch bản cho biểu đồ tuần tự trên là :

1. Người dùng đăng nhập thành công vào giao diện trang chủ Home.html và chọn Sample Home
2. Home.html gọi đến lớp SampleController
3. Lớp SampleController thực hiện hàm getAllSample()
4. Lớp SampleController gọi đến lớp SampleService
5. Lớp SampleService thực hiện hàm getAll()
6. Lớp SampleService gọi đến lớp SampleRepo
7. Lớp SampeRepo thực hiện hàm findAll()
8. Lớp SampleRepo gọi đến lớp Sample
9. Lớp Sample thực hiện hàm getter()
10. Lớp Sample trả về cho lớp SampleRepo danh sách đối tượng dạng Sample
11. Lớp SampleRepo trả về cho lớp SampleService danh sách đối tượng dạng Sample
12. Lớp SampleService trả về cho lớp SampleController danh sách đối tượng dạng Sample
13. Lớp SampleController gọi đến giao diện SampleAll.html
14. Giao diện SampleAll.html hiển thị giao diện cho người dùng
15. Người dùng nhấn vào nút Add Sample
16. Giao diện SampleAll.html gọi đến lớp SampleController
17. Lớp SampleController thực hiện hàm addSample()
18. Lớp SampleController gọi đến lớp Sample
19. Lớp Sample thực hiện hàm Sample()/setter()
20. Lớp Sample trả về cho lớp SampleController đối tượng dạng Sample
21. Lớp SampleController gọi tới lớp LabelService
22. Lớp LabelService thực hiện hàm getAll()
23. Lớp LabelService gọi tới lớp LabelRepo
24. Lớp LabelRepo thực hiện hàm findAll()
25. Lớp LabelRepo gọi tới lớp Label
26. Lớp Label thực hiện hàm getter()
27. Lớp Label trả về cho lớp LabelRepo danh sách đối tượng dạng Label
28. Lớp LabelRepo trả về cho lớp LabelService danh sách đối tượng dạng Label
29. Lớp LabelService trả về cho lớp SampleController danh sách đối tượng dạng Label
30. Lớp SampleController gọi tới giao diện SampleAdd.html
31. Giao diện SampleAdd.html hiển thị cho người dùng
32. Người dùng nhập ảnh và chọn nhãn, sau đó nhấn nút Save
33. Giao diện SampleAdd.html gọi tới lớp SampleController
34. Lớp SampleController gọi tới lớp Sample
35. Lớp Sample thực hiện hàm khởi tạo Sample()/setter()
36. Lớp Sample trả về cho lớp SampleController đối tượng dạng Sample
37. Lớp SampleController thực hiện hàm save ()
38. Lớp SampleController gọi tới lớp SampleService
39. Lớp SampleService thực hiện hàm saveSample()
40. Lớp SampleService gọi tới lớp SampleRepo
41. Lớp SampleRepo thực hiện hàm save()
42. Lớp SampleRepo trả về cho lớp SampleService biến true
43. Lớp SampleService trả về lớp SampleController
44. Lớp SampleController gọi tới giao diện SampleAll.html
45. Giao diện SampleAll.html hiển thị cho người dùng
46. Kết quả

A screenshot of a computer

Description automatically generated

*Giao diện trang chủ chứa chức năng thêm mẫu*

Người dùng nhấn vào nút Add Sample. Nút Add Sample đã được gán 1 đường link “@{/sample/add}”. Ở trong SampleController, khi được gọi tới đường link này, SampleController sẽ thực hiện hàm addSample(). Hàm addSample đưa 1 đối tượng Sample mới vào modelAttribute và đưa tất cả các nhãn mà bài toán đang có vào trong modelAttribute.

A screenshot of a computer

Description automatically generated

*Giao diện thêm nhãn hiển thị nút thêm ảnh và thanh chọn nhãn cho ảnh.*

Sau khi nhấn nút Save, SampleController sẽ lấy ra dối tượng Sample người dùng submit. SampleController sẽ khởi tạo 1 đối tượng Sample mới với các thuộc tính lấy từ đối tượng Sample người dùng đã submit, kèm theo đó là thời gian submit lên hệ thống. Sau đố SampleController sẽ lưu đối tượng Sample vào trong BD bằng cách sử dụng hàm saveSample() của SampleService. Sau khi lưu thành công, SampleController sẽ gọi tới lớp view của SampleAll.html

A screenshot of a computer screen

Description automatically generated

*Giao diện hiển thị mẫu 2 mới thêm thành công và có thời gian thêm*

Chức năng 3: Sửa mẫu cho bài toán số 2

1. Thiết kế biểu đồ hoạt động tuần tự

A diagram of a project

Description automatically generated

Kịch bản cho biểu đồ tuần tự trên :

1. Người dùng đăng nhập thành công vào giao diện trang chủ Home.html và chọn Sample Home
2. Home.html gọi đến lớp SampleController
3. Lớp SampleController thực hiện hàm getAllSample()
4. Lớp SampleController gọi đến lớp SampleService
5. Lớp SampleService thực hiện hàm getAll()
6. Lớp SampleService gọi đến lớp SampleRepo
7. Lớp SampeRepo thực hiện hàm findAll()
8. Lớp SampleRepo gọi đến lớp Sample
9. Lớp Sample thực hiện hàm getter()
10. Lớp Sample trả về cho lớp SampleRepo danh sách đối tượng dạng Sample
11. Lớp SampleRepo trả về cho lớp SampleService danh sách đối tượng dạng Sample
12. Lớp SampleService trả về cho lớp SampleController danh sách đối tượng dạng Sample
13. Lớp SampleController gọi đến giao diện SampleAll.html
14. Giao diện SampleAll.html hiển thị giao diện cho người dùng
15. Người dùng nhấn vào nút Edit
16. Giao diện SampleAll.html gọi đến lớp SampleController
17. Lớp SampleController thực hiện hàm editSample()
18. Lớp SampleController gọi đến lớp SampleService
19. Lớp SampleService thực hiện hàm getSampleById()
20. Lớp SampleService gọi tới lớp SampleRepo
21. Lớp SampleRepo thực hiện hàm findById()
22. Lớp SampleRepo gọi tới lớp Sample
23. Lớp Sample thực hiện hàm getter()
24. Lớp Sample trả về cho lớp SampleRepo đối tượng dạng Sample
25. Lớp SampleRepo trả về cho lớp SampleService đối tượng dạng Sample
26. Lớp SampleService trả về cho lớp SampleController đối tượng dạng Sample
27. Lớp SampleController gọi tới lớp LabelService
28. Lớp LabelService thực hiện hàm getLabelByName()
29. Lớp LabelService gọi tới lớp LabelRepo
30. Lớp LabelRepo thực hiện hàm findByName()
31. Lớp LabelRepo gọi tới lớp Label
32. Lớp Label thực hiện hàm getter()
33. Lớp Label trả về cho lớp LabelRepo đối tượng dạng Label
34. Lớp LabelRepo trả về cho LabelService đối tượng dạng Label
35. Lớp LabelService trả về cho lớp SampleController đối tượng dạng Label
36. Lớp SampleController gọi tới giao diện SampleEdit.html
37. Giao diện SampleEdit.html hiển thị cho người dùng
38. Người dùng nhập ảnh mới, chọn nhãn và nhấn nút Save
39. Giao diện SampleEdit.html gọi tới lớp SampleController
40. Lớp SampleController gọi tới lớp Sample
41. Lớp Sample thực hiện hàm khởi tạo Sample()/setter()
42. Lớp Sample trả về cho lớp SampleController đối tượng dạng Sample
43. Lớp SampleController thực hiện hàm update()
44. Lớp SampleController gọi tới lớp SampleService
45. Lớp SampleService thực hiện hàm saveSample()
46. Lớp SampleService gọi tới lớp SampleRepo
47. Lớp SampleRepo thực hiện hàm save()
48. Lớp SampleRepo trả về cho lớp SampleService biến true
49. Lớp SampleService trả về cho lớp SampleController
50. Lớp SampleController gọi tới giao diện SampleAll.html
51. Giao diện SampleAll.html hiển thị cho người dùng
52. Kết quả

A screenshot of a computer

Description automatically generated

*Giao diện trang chủ chứa chức năng sửa mẫu*

Khi người dùng nhấn vào nút edit, SampleController sẽ lấy id của mẫu tương ứng và tìm mẫu đó trong DB bằng SampleService. Sau đó SampleController sẽ gọi tới lớp view SampleAdd.html

*A hand making a gesture

Description automatically generated*

*Giao diện sửa mẫu với nút chọn ảnh và chọn lại nhãn.*

Sau khi người dùng chọn lại ảnh, nhãn và nhấn nút Update, SampleController sẽ lấy được nhãn mới, ảnh mới và cập nhật lại thời gian thay đổi cho ảnh.

A screenshot of a computer screen

Description automatically generated

*Kết quả thay đổi mẫu*

Chức năng 4: Xóa mẫu cho bài toán số 2

1. Thiết kế giao diện

A screenshot of a computer

Description automatically generated

*Giao diện trang chủ với chức năng xóa mẫu*

*A screenshot of a computer

Description automatically generated*

*Giao diện xóa mẫu*

1. Thiết kế biểu đồ hoạt động tuần tự

A diagram of a computer

Description automatically generated

Kịch bản cho biểu đồ tuần tự trên

1. Người dùng đăng nhập thành công vào giao diện trang chủ Home.html và chọn Sample Home
2. Home.html gọi đến lớp SampleController
3. Lớp SampleController thực hiện hàm getAllSample()
4. Lớp SampleController gọi đến lớp SampleService
5. Lớp SampleService thực hiện hàm getAll()
6. Lớp SampleService gọi đến lớp SampleRepo
7. Lớp SampeRepo thực hiện hàm findAll()
8. Lớp SampleRepo gọi đến lớp Sample
9. Lớp Sample thực hiện hàm getter()
10. Lớp Sample trả về cho lớp SampleRepo danh sách đối tượng dạng Sample
11. Lớp SampleRepo trả về cho lớp SampleService danh sách đối tượng dạng Sample
12. Lớp SampleService trả về cho lớp SampleController danh sách đối tượng dạng Sample
13. Lớp SampleController gọi đến giao diện SampleAll.html
14. Giao diện SampleAll.html hiển thị giao diện cho người dùng
15. Người dùng nhấn vào nút Delete
16. Giao diện SampleAll.html gọi đến lớp SampleController
17. Lớp SampleController thực hiện hàm deleteSample()
18. Lớp SampleController gọi tới lớp SampleService
19. Lớp SampleService thực hiện hàm getSampleById()
20. Lớp SampleService gọi tới lớp SampleRepo
21. Lớp SampleRepo thực hiện hàm findById()
22. Lớp SampleRepo gọi tới lớp Sample
23. Lớp Sample thực hiện hàm getter()
24. Lớp Sample trả về cho lớp SampleRepo đối tượng dạng Sample
25. Lớp SampleRepo trả về cho lớp SampleService đối tượng dạng Sample
26. Lớp SampleService trả về cho lớp SampleController đối tượng dạng Sample
27. Lớp SampleController gọi tới giao diện SampleDelete.html
28. Giao diện SampleDelete.html hiển thị cho người dùng
29. Người dùng nhấn vào nút Delete
30. Giao diện SampleDelete.html gọi tới lớp SampleController
31. Lớp SampleController thực hiện hàm deleteSampleConfirm()
32. Lớp SampleController gọi tới giao diện SampleDeleteConfirm.html
33. Giao diện SampleDeleteConfirm.html hiển thị cho người dùng
34. Người dùng nhấn nút Yes
35. Giao diện SampleDeleteConfirm.html gọi tới lớp SampleController
36. Lớp SampleController thực hiện hàm deleteSample()
37. Lớp SampleController gọi tới lớp SampleService
38. Lớp SampleService thực hiện hàm deleteSample()
39. Lớp SampleService gọi tới lớp Sample
40. Lớp Sample thực hiện hàm getter()
41. Lớp Sample trả về cho lớp SampleService đối tượng dạng Sample
42. Lớp SampleService gọi tới lớp SampleRepo
43. Lớp SampleRepo thực hiện hàm delete()
44. Lớp SampleRepo trả về cho lớp SampleService biến true
45. Lớp SampleService trả về cho lớp SampleController
46. Lớp SampleController gọi tới giao diện SampleAll.html
47. Giao diện SampleAll.html hiển thị cho người dùng
48. Kết quả

A screenshot of a computer

Description automatically generated

*Giao diện trang chủ với chức năng xóa mẫu*

Tại giao diện trang chủ, khi người dùng nhấn nút Delete, SampleController sẽ lấy được id của mẫu cần xóa và tìm nó trong DB bằng SampleService. SampleService thực hiện hàm deleteSampleForm(), đưa các thông tin của mẫu cần xóa vào trong modelAttribute và hiển thị ra bằng lớp view SampleDeleteConfirm.html

A hand making a sign

Description automatically generated

*Giao diện trang xóa mẫu*

Sau khi người dùng nhấn nút delete, SampleController sẽ thực hiện hàm deleteSample() và trả về giao diện trang chủ cho người dùng

A screenshot of a computer

Description automatically generated

*Kết quả xóa mẫu thành công*