**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

-----🙞🙜🕮🙞🙜-----

****

**Phát triển các hệ thống thông minh**

|  |  |
| --- | --- |
| **Mã đề tài**: | 24 |
| **Tên đề tài**: | Hệ thống tự động phân loại email theo chủ đề |
| **Nhóm học phần**: | 01 |
| **Nhóm môn học**: | 03 |
| **Sinh viên:** Trần Quang Hưng – B19DCCN333 | |
| **Thực hiện các module chính:**   * Module quản lý mẫu (một phần Server) * Module các chức năng phía Client | |

**Hà Nội 2023**

**Mục lục**

[**1.** **Xây dựng dữ liệu và đánh giá thuật toán/mô hình sử dụng** 4](#_Toc142312725)

[**1.1.** **Bộ dữ liệu** 4](#_Toc142312726)

[**1.1.1.** **Dữ liệu thô** 4](#_Toc142312727)

[**1.1.2.** **Dữ liệu đã được làm sạch** 4](#_Toc142312728)

[**1.2.** **Thuật toán huấn luyện học máy** 5](#_Toc142312729)

[**2.** **Thiết kế hệ thống** 5](#_Toc142312730)

[**2.1.** **Cơ sở dữ liệu** 5](#_Toc142312731)

[**2.2.** **Phía Server** 6](#_Toc142312732)

[**2.2.1.** **Biểu đồ lớp** 6](#_Toc142312733)

[**2.2.1.1.** **Module quản lý mẫu (theo dõi/đánh dấu email/thêm mới)** 6](#_Toc142312734)

[**2.2.1.2.** **Các API cho User** 8](#_Toc142312735)

[**2.2.1.3.** **Các API cho Email** 9](#_Toc142312736)

[**2.2.1.4.** **Các API cho Label** 10](#_Toc142312737)

[**2.2.1.5.** **Các API cho Sample** 11](#_Toc142312738)

[**2.2.2.** **Biểu đồ tuần tự** 12](#_Toc142312739)

[**2.2.2.1.** **Chức năng quản lý mẫu** 12](#_Toc142312740)

[**2.2.2.1.1.** **Xem danh sách mẫu** 12](#_Toc142312741)

[**2.2.2.1.2.** **Đánh dấu email là mẫu** 14](#_Toc142312742)

[**2.2.2.1.3.** **Thêm mẫu mới** 15](#_Toc142312743)

[**2.2.2.2.** **Các API cho User** 16](#_Toc142312744)

[**2.2.2.3.** **Các API cho Email** 17](#_Toc142312745)

[**2.2.2.4.** **Các API cho Label** 18](#_Toc142312746)

[**2.2.2.5.** **Các API cho Sample** 20](#_Toc142312747)

[**2.3.** **Phía Client** 21](#_Toc142312748)

[**2.3.1.** **Biểu đồ lớp** 21](#_Toc142312749)

[**2.3.1.1.** **Chức năng quản lý email (theo dõi/gán nhãn/gửi)** 21](#_Toc142312750)

[**2.3.1.2.** **Chức năng quản lý nhãn (thêm/xóa)** 22](#_Toc142312751)

[**2.3.2.** **Biểu đồ tuần tự** 23](#_Toc142312752)

[**2.3.2.1.** **Chức năng quản lý email** 23](#_Toc142312753)

[**2.3.2.1.1.** **Theo dõi danh sách email** 23](#_Toc142312754)

[**2.3.2.1.2.** **Gán nhãn cho email** 24](#_Toc142312755)

[**2.3.2.1.3.** **Gửi email** 25](#_Toc142312756)

[**2.3.2.2.** **Chức năng quản lý nhãn (thêm/xóa)** 26](#_Toc142312757)

[**3.** **Cài đặt hệ thống** 27](#_Toc142312758)

[**3.1.** **Mô tả chung** 27](#_Toc142312759)

[**3.2.** **Kết quả chạy** 27](#_Toc142312760)

[**3.2.1.** **Phía Server** 27](#_Toc142312761)

[**3.2.1.1.** **Xem danh sách mẫu** 28](#_Toc142312762)

[**3.2.1.2.** **Đánh dấu email là mẫu** 28](#_Toc142312763)

[**3.2.1.3.** **Thêm mẫu mới** 28](#_Toc142312764)

[**3.2.2.** **Phía Client** 29](#_Toc142312765)

[**3.2.2.1.** **Đăng nhập** 29](#_Toc142312766)

[**3.2.2.2.** **Xem danh sách email** 29](#_Toc142312767)

[**3.2.2.3.** **Gán nhãn cho email** 30](#_Toc142312768)

[**3.2.2.4.** **Gửi email** 30](#_Toc142312769)

[**3.2.2.5.** **Quản lý nhãn** 31](#_Toc142312770)

1. **Xây dựng dữ liệu và đánh giá thuật toán/mô hình sử dụng**

Do khi làm việc nhóm, phần lớn nội dung của phần này là do một thành viên trong nhóm của em đóng góp, nên em sẽ chỉ giới thiệu sơ qua về bộ dữ liệu và thuật toán sử dụng cho mô hình học máy để tài liệu có tính liên kết.

* 1. **Bộ dữ liệu**
     1. **Dữ liệu thô**

Nguồn: [The Enron Email Dataset](https://www.kaggle.com/datasets/wcukierski/enron-email-dataset)

* Có tổng cộng **517401** mẫu.
* Có tổng cộng **5336** nhãn.
* Mỗi email có **1** nhãn.
* Có hai cột dữ liệu là “file” và “message”, trong đó “file” là nhãn.
* Với “message”, nội dung của một mẫu thường sẽ chứa:
  + Message-ID: Id của email.
  + Date: Ngày gửi.
  + From: Email người gửi.
  + To: Email người nhận.
  + Subject: Tiêu đề.
  + Mime-Version: Tiêu chuẩn, phần mở rộng của email cho phép người dùng trao đổi các tệp tin và dữ liệu.
  + Content-Type: Tiêu đề mô tả phương tiện bên trong nội dung thư.
  + Content-Transfer-Encoding: Tiêu đề cho thấy loại lược đồ mã hóa nhị phân thành dạng văn bản đã được dùng ngoài mã hóa ban đầu chỉ định trong tiêu đề nội dung hay chưa.
  + X-From: Tên người gửi.
  + X-To: Tên người nhận.
  + X-cc: Bản sao.
  + X-bcc: Các địa chỉ cùng nhận thư.
  + X-Folder: Folder của mail.
  + X-FileName: File đính kèm.
  + Body: Nội dung email.
    1. **Dữ liệu đã được làm sạch**

Nguồn: [The Enron Email Dataset - Cleaned Data](https://www.kaggle.com/code/ankur561999/data-cleaning-enron-email-dataset/output?select=cleaned_data.csv)

* Có tổng cộng **489236** mẫu.
* Có tổng cộng **1767** nhãn.
* Mỗi email có **1** nhãn.
* Độ dài của một mail:
  + Ngắn nhất: 1 ký tự.
  + Trung bình: khoảng 1811 ký tự.
  + Dài nhất: 2011422 ký tự.
* Có 3 cột dữ liệu là “subject”, “body” và “X-Folder”. Chúng lần lượt là tiêu đề của email, nội dung email và nhãn của email.
  1. **Thuật toán huấn luyện học máy**

Sau khi thử huấn luyện với tập dữ liệu nhỏ và đánh giá các thuật toán *Gaussian NB, Multinomial NB, Decision Tree, SVM, AdaBoost, ANN*, chúng em đã được kết quả như sau:

A screenshot of a computer

Description automatically generated

Do có độ chính xác cao nhất và thời gian huấn luyện chỉ chậm hơn Gaussian NB nên chúng em thống nhất sử dụng thuật toán SVM để huấn luyện mô hình.

1. **Thiết kế hệ thống**
   1. **Cơ sở dữ liệu**

A diagram of a computer

Description automatically generated

* 1. **Phía Server**
     1. **Biểu đồ lớp**
        1. **Module quản lý mẫu (theo dõi/đánh dấu email/thêm mới)**

A computer screen shot of a computer program

Description automatically generated

Mô tả module:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Chức năng** |
| JpaRepository<T> | findById(id) | Tìm trong CSDL và trả về T có id là id được truyền vào |
| findAll() | Tìm trong CSDL và trả về tất cả các bản ghi |
| SampleRepository | findByEmailId(id) | Tìm trong CSDL và trả về Sample với email có id là id được truyền vào |
| SampleController | showSampleList() | GetMapping với giao diện danh sách email |
| showEmailDetail(id) | GetMapping với giao diện chi tiết email |
| showNewSampleView() | GetMapping với giao diện tạo mẫu mới |
| update(isSampleMarks) | PostMapping với hành động yêu cầu lưu trạng thái của email có phải mẫu hay không |
| newSample(email) | PostMapping với hành động yêu cầu lưu tạo mới mẫu |

* + - 1. **Các API cho User**

A screenshot of a computer program

Description automatically generated

Mô tả module:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Chức năng** |
| UserRepository | findByUsernameAndPassword(u, p) | Tìm trong CSDL và trả về User có username là u và password là p |
| UserRestController | validate(user) | API xác thực đăng nhập cho user |

* + - 1. **Các API cho Email**

A computer screen shot of a computer program

Description automatically generated

Mô tả module:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Chức năng** |
| EmailRestController | findList(receiverId, labelId) | API trả về danh sách các Email có id của người nhận là receiverId, id của nhãn là labelId. |
| save(email) | API lưu email vào CSDL |

* + - 1. **Các API cho Label**

A computer screen shot of a computer program

Description automatically generated

Mô tả module:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Chức năng** |
| LabelOfUserRepository | findByUserIdAndLabelId(userId, labelId) | Tìm trong CSDL và trả về LabelOfUser có id của người dùng và id của nhãn trùng với userId và labelId |
| LabelRestController | add(user, label) | API lưu nhãn mới cho người dùng |
| delete(user, label) | API xóa nhãn cho người dùng |

* + - 1. **Các API cho Sample**

A screenshot of a computer program

Description automatically generated

Mô tả module:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Chức năng** |
| SampleRepository | findByTrained(isTrained) | Tìm trong CSDL và trả về danh sách Sample với thuộc tính isTrained có giá trị giống như giá trị truyền vào |
| SampleRestController | findAll() | API trả về danh sách tất cả các mẫu huấn luyện |
| findUntrained() | API trả về danh sách tất cả các mẫu huấn luyện chưa được dùng để huấn luyện |

* + 1. **Biểu đồ tuần tự**
       1. **Chức năng quản lý mẫu**
          1. **Xem danh sách mẫu**

A screenshot of a computer screen

Description automatically generated

Kịch bản:

1. **Admin mở hệ thống, lớp SampleController sẽ được gọi.**
2. Lớp SampleController gọi phương thức showSampleList() nhằm hiển thị giao diện.
3. Phương thức showSampleList() gọi lớp EmailRepository nhằm lấy danh sách các email.
4. Lớp EmailRepository gọi phương thức findAll().
5. Phương thức findAll() gọi lớp Email.
6. Lớp Email đóng gói dữ liệu.
7. Lớp Email trả kết quả cho phương thức findAll().
8. Lớp EmailRepository trả kết quả cho phương thức showSampleList().
9. Phương thức showSampleList() gọi lớp SampleRepository nhằm lấy danh sách các mẫu đã sẵn có.
10. Lớp SampleRepository gọi phương thức findAll().
11. Phương thức findAll() gọi lớp Sample.
12. Lớp Sample đóng gói dữ liệu.
13. Lớp Sample trả kết quả cho phương thức findAll().
14. Lớp SampleRepository trả kết quả cho phương thức showSampleList().
15. Lớp SampleController trả kết quả về cho lớp SampleList.html.
16. Lớp SampleList.html hiển thị giao diện cho Admin.
    * + - 1. **Đánh dấu email là mẫu**

A black background with white text and blue circles

Description automatically generated

Kịch bản:

1. **Admin đang ở giao diện danh sách mẫu và click vào một email trong danh sách.**
2. Lớp SampleList.html gọi lớp SampleController.
3. Lớp SampleController gọi phương thức showEmailDetail() nhằm hiển thị giao diện.
4. Phương thức showEmailDetail() gọi lớp EmailRepository nhằm lấy thông tin của email đã chọn.
5. Lớp EmailRepository gọi phương thức findById().
6. Phương thức findById() gọi lớp Email.
7. Lớp Email đóng gói dữ liệu.
8. Lớp Email trả kết quả cho phương thức findById().
9. Lớp EmailRepository trả kết quả cho phương thức showEmailDetail().
10. Lớp SampleController trả kết quả về lớp EmailDetail.html.
11. Lớp EmailDetail.html hiển thị giao diện cho Admin.
12. **Admin click vào checkbox để đánh dấu một email là mẫu.**
13. Lớp EmailDetail.html gọi lớp SampleController.
14. Lớp SampleController gọi phương thức update() nhằm cập nhật lại dữ liệu của mẫu.
15. Phương thức update() gọi lớp SampleRepository.
16. Lớp SampleRepository gọi phương thức save().
17. Phương thức save() gọi lớp Sample.
18. Lớp Sample đóng gói dữ liệu.
19. Lớp Sample trả kết quả cho phương thức save().
20. Lớp SampleRepository trả kết quả cho phương thức update().
21. Lớp SampleController trả kết quả về lớp EmailDetail.html.
22. Lớp EmailDetail.html hiển thị kết quả cho Admin.
    * + - 1. **Thêm mẫu mới**

A screenshot of a computer screen

Description automatically generated

Kịch bản:

1. **Admin đang ở giao diện danh sách mẫu và click chọn chức năng Tạo mẫu mới.**
2. Lớp SampleList.html gọi lớp SampleController.
3. Lớp SampleController gọi phương thức showNewSampleView() nhằm hiển thị giao diện.
4. Phương thức showNewSampleView() gọi lớp LabelRepository nhằm lấy danh sách các nhãn để có thể tạo mẫu mới.
5. Lớp LabelRepository gọi phương thức findAll().
6. Phương thức findAll() gọi lớp Label.
7. Lớp Label đóng gói dữ liệu.
8. Lớp Label trả kết quả cho phương thức findAll().
9. Lớp LabelRepository trả kết quả cho phương thức showNewSampleView().
10. Lớp SampleController trả kết quả về lớp NewSample.html.
11. Lớp NewSample.html hiển thị giao diện cho Admin.
12. **Admin điền dữ liệu vào các trường nhập liệu được yêu cầu rồi nhấn hoàn thành.**
13. Lớp NewSample.html gọi lớp SampleController.
14. Lớp SampleController gọi phương thức newSample() nhằm tạo một email mới và đặt email đó là một mẫu huấn luyện.
15. Phương thức newSample() gọi lớp EmailRepository.
16. Lớp EmailRepository gọi phương thức save().
17. Phương thức save() gọi lớp Sample().
18. Lớp Email đóng gói dữ liệu.
19. Lớp Email trả kết quả cho phương thức save().
20. Lớp EmailRepository trả kết quả cho phương thức newSample().
21. Phương thức newSample() gọi lớp SampleRepository.
22. Lớp SampleRepository gọi phương thức save().
23. Phương thức save() gọi lớp Sample().
24. Lớp Sample đóng gói dữ liệu.
25. Lớp Sample trả kết quả cho phương thức save().
26. Lớp SampleRepository trả kết quả cho phương thức newSample().
27. Lớp SampleController trả kết quả về lớp NewSample.html.
28. Lớp NewSample.html hiển thị kết quả cho Admin.
    * + 1. **Các API cho User**

A screenshot of a computer screen

Description automatically generated

Kịch bản:

1. **Client gọi API yêu cầu xác thực khi đăng nhập tới lớp UserRestController.**
2. Lớp UserRestController gọi phương thức validate().
3. Phương thức validate() gọi lớp UserRepository.
4. Lớp UserRepository gọi phương thức findByUsernameAndPassword().
5. Phương thức findByUsernameAndPassword() gọi lớp User.
6. Lớp User đóng gói dữ liệu.
7. Lớp User trả kết quả cho phương thức findByUsernameAndPassword().
8. Lớp UserRepository trả kết quả cho phương thức validate().
9. Lớp UserRestController trả kết quả cho Client.
   * + 1. **Các API cho Email**

A screen shot of a computer

Description automatically generated

Kịch bản:

1. **Client gọi API yêu cầu lấy danh sách email (findAll) hoặc lấy email nhất định (findById) hoặc lưu một email (save) tới lớp EmailRestController.**
2. Lớp EmailRestController gọi phương thức findAll() / findById() / save().
3. Phương thức findAll() / findById() / save() gọi lớp EmailRepository.
4. Lớp EmailRepository gọi phương thức findAll() / findById() / save().
5. Phương thức findAll() / findById() / save() gọi lớp Email.
6. Lớp Email đóng gói dữ liệu.
7. Lớp Email trả kết quả cho phương thức findAll() / findById() / save().
8. Lớp EmailRepository trả kết quả cho phương thức findAll() / findById() / save().
9. Lớp EmailRestController trả kết quả cho Client.
   * + 1. **Các API cho Label**

A black background with white text and blue dots

Description automatically generated

Kịch bản:

1. **Client gọi API yêu cầu lấy danh sách nhãn của một người dùng tới lớp LabelRestController.**
2. Lớp LabelRestController gọi phương thức findAllByUser().
3. Phương thức findAllByUser() gọi lớp LabelOfUserRepository.
4. Lớp LabelOfUserRepository gọi phương thức findAllByUserId().
5. Phương thức findAllByUserId() gọi lớp LabelOfUser.
6. Lớp LabelOfUser đóng gói dữ liệu.
7. Lớp LabelOfUser trả kết quả cho phương thức findAllByUserId().
8. Lớp LabelOfUserRepository trả kết quả cho phương thức findAllByUser().
9. Lớp UserRestController trả kết quả cho Client.
10. **Client gọi API yêu cầu thêm một nhãn mới cho người dùng tới lớp LabelRestController.**
11. Lớp LabelRestController gọi phương thức add().
12. Phương thức add() gọi lớp LabelRepository.
13. Lớp LabelRepository gọi phương thức save().
14. Phương thức save() gọi lớp Label.
15. Lớp Label đóng gói dữ liệu.
16. Lớp Label trả kết quả cho phương thức save().
17. Lớp LabelRepository trả kết quả cho phương thức add().
18. Phương thức add() gọi lớp LabelOfUserRepository.
19. Lớp LabelOfUserRepository gọi phương thức save().
20. Phương thức save() gọi lớp LabelOfUser.
21. Lớp LabelOfUser đóng gói dữ liệu.
22. Lớp LabelOfUser trả kết quả cho phương thức save().
23. Lớp LabelOfUserRepository trả kết quả cho phương thức add().
24. Lớp UserRestController trả kết quả cho Client.
25. **Client gọi API yêu cầu xóa nhãn cho một người dùng tới lớp LabelRestController.**
26. Lớp LabelRestController gọi phương thức delete().
27. Phương thức delete() gọi lớp LabelOfUserRepository.
28. Lớp LabelOfUserRepository gọi phương thức delete().
29. Lớp LabelOfUserRepository trả kết quả cho phương thức delete().
30. Lớp UserRestController trả kết quả cho Client.
    * + 1. **Các API cho Sample**

A screenshot of a computer

Description automatically generated

Kịch bản:

1. **Client gọi API yêu cầu lấy danh sách mẫu theo yêu cầu nhất định tới lớp SampleRestController.**
2. Lớp SampleRestController gọi phương thức findAll() / findUntrained().
3. Phương thức findAll() / findUntrained() gọi lớp SampleRepository.
4. Lớp SampleRepository gọi phương thức findAll() / findByTrained().
5. Phương thức findAll() / findUntrained() gọi lớp Sample.
6. Lớp Sample đóng gói dữ liệu.
7. Lớp Sample trả kết quả cho phương thức findAll() / findByTrained().
8. Lớp SampleRepository trả kết quả cho phương thức findAll() / findUntrained().
9. Lớp SampleRestController trả kết quả cho Client.
   1. **Phía Client**
      1. **Biểu đồ lớp**
         1. **Chức năng quản lý email (theo dõi/gán nhãn/gửi)**

A screenshot of a computer

Description automatically generated

Mô tả module:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Chức năng** |
| EmailController | showEmailList() | GetMapping với giao diện danh sách email |
| showEmailDetail(id) | GetMapping với giao diện thông tin chi tiết của email |
| showSendEmailView() | GetMapping với giao diện gửi email |
| sendEmail(email) | PostMapping với hành động gửi email |
| changeLabel(email) | PostMapping với hành động đổi nhãn của email |
| EmailService | getAll(uid) | Gọi dịch vụ lấy danh sách tất cả các Email của người dùng có id là uid |
| getById(id) | Gọi dịch vụ lấy thông tin chi tiết của email có id là id truyền vào. |
| save(email) | Gọi dịch vụ lưu mới hoặc thay đổi cho email vào CSDL. |

* + - 1. **Chức năng quản lý nhãn (thêm/xóa)**

A screenshot of a computer

Description automatically generated

Mô tả module:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Chức năng** |
| LabelController | showView() | GetMapping với giao diện quản lý nhãn |
| addLabel(user, label) | PostMapping với hành động người dùng thêm nhãn mới |
| deleteLabel(user, label) | PostMapping với hành động người dùng xóa nhãn của bản thân |
| LabelService | add(user, label) | Gọi dịch vụ yêu cầu thêm nhãn mới cho người dùng |
| delete(user, label) | Gọi dịch vụ yêu cầu xóa nhãn của người dùng |
| getAllByUser(user) | Gọi dịch vụ lấy danh sách tất cả các nhãn của user truyền vào |

* + 1. **Biểu đồ tuần tự**
       1. **Chức năng quản lý email**
          1. **Theo dõi danh sách email**

A screenshot of a computer

Description automatically generated

Kịch bản:

1. **Client đăng nhập thành công, lớp EmailController được gọi để hiển thị giao diện.**
2. Lớp EmailController gọi phương thức showEmailList().
3. Phương thức showEmailList() gọi lớp EmailService.
4. Lớp EmailService gọi phương thức getAll().
5. Phương thức getAll() gọi API yêu cầu lấy danh sách tất cả các email của Client.
6. Dịch vụ trả kết quả cho phương thức getAll().
7. Lớp EmailService trả kết quả cho phương thức showEmailList().
8. Lớp EmailController trả kết quả cho lớp EmailList.html.
9. Lớp EmailList.html hiển thị giao diện cho Client.
   * + - 1. **Gán nhãn cho email**

A black background with white text and blue circles

Description automatically generated

Kịch bản:

1. **Client đang ở giao diện có chứa danh sách email của mình và click chọn một email.**
2. Lớp EmailList.html gọi lớp EmailController.
3. Lớp EmailController gọi phương thức showEmailDetail().
4. Phương thức showEmailDetail() gọi lớp EmailService.
5. Lớp EmailService gọi phương thức getById().
6. Phương thức getById() gọi API yêu cầu lấy thông tin chi tiết của một email.
7. Dịch vụ trả kết quả cho phương thức getById().
8. Lớp EmailService trả kết quả cho phương thức showEmailDetail().
9. Lớp EmailController trả kết quả cho lớp Detail.html.
10. Lớp Detail.html hiển thị giao diện cho Client.
11. **Client chọn một nhãn khác nhãn hiện tại cho email.**
12. Lớp Detail.html gọi lớp EmailController.
13. Lớp EmailController gọi phương thức changeLabel().
14. Phương thức changeLabel() gọi lớp EmailService.
15. Lớp EmailService gọi phương thức save().
16. Phương thức save() gọi API yêu cầu lưu lại thay đổi của email.
17. Dịch vụ trả kết quả cho phương thức save().
18. Lớp EmailService trả kết quả cho phương thức changeLabel().
19. Lớp EmailController trả kết quả cho lớp Detail.html.
20. Lớp Detail.html hiển thị giao diện cho Client.
    * + - 1. **Gửi email**

A group of blue circles and white text

Description automatically generated

Kịch bản:

1. **Client đang ở giao diện có chứa danh sách email của mình và click chọn chức năng gửi email.**
2. Lớp EmailList.html gọi lớp EmailController yêu cầu hiển thị giao diện.
3. Lớp EmailController gọi phương thức showSendEmailView().
4. Lớp EmailController trả kết quả cho lớp SendEmail.html.
5. Lớp SendEmail.html hiển thị giao diện cho Client.
6. **Client nhập nội dung và các thông tin yêu cầu rồi nhấn gửi.**
7. Lớp SendEmail.html gọi lớp EmailController yêu cầu thực hiện gửi email.
8. Lớp EmailController gọi phương thức sendEmail().
9. Phương thức sendEmail() gọi lớp EmailService.
10. Lớp EmailService gọi phương thức save().
11. Phương thức save() gọi API yêu cầu lưu lại email mới gửi.
12. Dịch vụ trả kết quả cho phương thức save().
13. Lớp EmailService trả kết quả cho phương thức sendEmail().
14. Lớp EmailController trả kết quả cho lớp SendEmail.html.
15. Lớp SendEmail.html hiển thị kết quả cho Client.
    * + 1. **Chức năng quản lý nhãn (thêm/xóa)**

A black background with white text and blue dots

Description automatically generated

Kịch bản:

1. **Client đang ở giao diện có chứa danh sách email của mình và click chọn chức năng quản lý nhãn.**
2. Lớp EmailList.html gọi lớp LabelController yêu cầu hiển thị giao diện.
3. Lớp LabelController gọi phương thức showView().
4. Phương thức showView() gọi lớp LabelService.
5. Lớp LabelService gọi phương thức getAllByUser().
6. Phương thức getAllByUser() gọi API yêu cầu lấy danh sách tất cả nhãn của Client.
7. Dịch vụ trả kết quả cho phương thức getAllByUser().
8. Lớp LabelService trả kết quả cho phương thức showView().
9. Lớp LabelController trả kết quả cho lớp LabelList.html.
10. Lớp LabelList.html hiển thị giao diện cho Client.
11. **Client điền tên một nhãn mới và nhấn thêm.**
12. Lớp EmailList.html gọi lớp LabelController yêu cầu thêm nhãn mới cho Client.
13. Lớp LabelController gọi phương thức addLabel().
14. Phương thức addLabel() gọi lớp LabelService.
15. Lớp LabelService gọi phương thức add().
16. Phương thức add() gọi API yêu cầu thêm nhãn mới cho Client.
17. Dịch vụ trả kết quả cho phương thức add().
18. Lớp LabelService trả kết quả cho phương thức addLabel().
19. Lớp LabelController trả kết quả cho lớp LabelList.html.
20. Lớp LabelList.html hiển thị kết quả cho Client.
21. **Client click xóa một nhãn.**
22. Lớp EmailList.html gọi lớp LabelController yêu cầu xóa nhãn cho Client.
23. Lớp LabelController gọi phương thức deleteLabel().
24. Phương thức deleteLabel() gọi lớp LabelService.
25. Lớp LabelService gọi phương thức delete().
26. Phương thức delete() gọi API yêu cầu xóa nhãn cho Client.
27. Dịch vụ trả kết quả cho phương thức delete().
28. Lớp LabelService trả kết quả cho phương thức deleteLabel().
29. Lớp LabelController trả kết quả cho lớp LabelList.html.
30. Lớp LabelList.html hiển thị kết quả cho Client.
31. **Cài đặt hệ thống**
    1. **Mô tả chung**

* Kiến trúc theo chiều ngang: MVC.
* Kiến trúc hướng dịch vụ: RESTful API.
* Framework, tool và engine sử dụng: Spring Boot, Thymeleaf, Lombok, Bootstrap, …
* Source code: [PTIT.4.3.ISD.Assignment on Github.](https://github.com/hlight-git/PTIT.4.3.ISD.Assignment.git)
  1. **Kết quả chạy**
     1. **Phía Server**
        1. **Xem danh sách mẫu**

**A screenshot of a email list

Description automatically generated**

* + - 1. **Đánh dấu email là mẫu**

**A screenshot of a computer

Description automatically generated**

* + - 1. **Thêm mẫu mới**

**A white background with black text

Description automatically generated**

* + 1. **Phía Client**
       1. **Đăng nhập**

**A screenshot of a computer

Description automatically generated**

* + - 1. **Xem danh sách email**

**A screenshot of a computer

Description automatically generated**

* + - 1. **Gán nhãn cho email**

**A screenshot of a computer

Description automatically generated**

* + - 1. **Gửi email**

**A screen shot of a computer

Description automatically generated**

* + - 1. **Quản lý nhãn**

**A screenshot of a computer

Description automatically generated**