**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NÔNG LÂM TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Logo

Description automatically generated**

**TÀI LIỆU YÊU CẦU MÔN**

**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**Chủ đề: Học tiếng Anh online**

Tên sinh viên: Tạ Anh Kiệt MSSV: 18130119

Bùi Anh Tuấn MSSV: 18130267

Nguyễn Hồng Viên MSSV: 18130282

Gv hướng dẫn: Vũ Đức Công Song

**HK2 2020 - 2021**

**Mục lục**

[I. Mục tiêu 1](#_Toc75296083)

[II. Phân tích 1](#_Toc75296084)

[III. Thiết kế 2](#_Toc75296085)

[1. Use – case diagram 2](#_Toc75296086)

[2. Mô tả use – case 3](#_Toc75296087)

[3. Sequence diagram 8](#_Toc75296088)

[4. Class diagram 10](#_Toc75296089)

[5. Test case 11](#_Toc75296090)

## 

# Mục tiêu

* + - Xây dựng một website học tiếng Anh dành cho lứa tuổi 5 – 10.
    - Các bài học đa dạng về chủ đề, màu sắc, hình ảnh âm thanh tạo hứng thú khi học.
    - Có các bài kiểm tra đánh giá trình độ theo từng bài học, theo từng chủ đề.
    - Có thể truy cập website học tiếng Anh thông qua nhiều nền tảng như smart phone, tablet, PC, laptop.

# Phân tích

1. **Ứng dụng này là gì ?**

Ngày nay bên cạnh các phương pháp cho trẻ học tiếng anh truyền thống như là đến trung tâm tiếng anh hay cho trẻ học với gia sư thì bố mẹ có thể chủ động sử dụng các phần mềm học tiếng anh cho trẻ học tại nhà. Đây là một phương pháp rất thú vị cũng là cách mà bố mẹ có thể tham gia học tiếng Anh cùng bé**.**

**Website được xây dựng để cung cấp các bài học đa dạng cho bé một vốn từ vựng theo chủ đề, màu sắc, hình ảnh âm thanh giúp bé ghi nhớ một cách dễ dàng.**

1. **Ứng dụng dành cho ai ?**

Dành cho đối tượng học sinh có lứa tuổi từ 5 đến 10 tuổi. Phụ huynh có thể dễ dàng hướng dẫn cho con mình cách sử dụng.

1. **Ứng dụng sử dụng như thế nào ?**

* Người dùng sử dụng một thiết bị thông minh có kết nối mạng như là smart phone, tablet, PC hoặc laptop để truy cập vào trang web.
* Để sử dụng được hệ thống người dùng phải đăng kí một tài khoản. Sau đó người dùng đăng nhập vào hệ thống để bắt đầu hoặc tiếp tục học bài học của mình.

1. **Ứng dụng này sử dụng khi nào ?**

Bất cứ khi nào người dùng muốn, và truy cập vào website.

1. **Ứng dụng này sử dụng ở đâu ?**

Ứng dụng được xây dựng trên nền tảng website. Người dùng có thể truy cập ở bất cứ đâu thông qua địa chỉ của website.

1. **Tại sao lại sử dụng ứng dụng ?**

* Nhờ sự phát triển của Internet, app học tiếng anh cho trẻ có thể dễ dàng truy cập mọi lúc mọi nơi, miễn là có mạng Internet.
* Không cần phải đi xa cũng có thể học tiếng anh tại nhà nhờ vào kho tàng kiến thức có sẵn trên app học tiếng anh cho trẻ.
* Website hướng đến tạo sự thích thú thông qua các hình ảnh, âm thanh và đồng thời tạo môi trường canh tranh thông qua các bài kiếm tra để người học có thể kiếm tra được trình độ của mình.

# Thiết kế

## Use – case diagram

Diagram

Description automatically generated

## Mô tả use – case

1. Đăng kí tài khoản

|  |  |
| --- | --- |
| Tên | Đăng kí tài khoản |
| Actors | User |
| Tóm tắt | Người dùng trước khi sử dụng trang web bắt buộc phải có một tài khoản để sử dụng. |
| Điều kiện tiên quyết | Không có |
| Sự kiện bắt đầu | Ở usecase đăng nhập nếu chưa có tài khoản. |
| Sự kiện kết thúc | User có thể đăng nhập vào hệ thống bằng tài khoản vừa đăng kí |

Dòng sự kiện chính:

* Usecase này bắt đầu khi người dùng muốn đăng kí tài khoản vào hệ thống.
* Người dùng đang ở giao diện trang chủ của website click chuột vào Đăng kí.
* Sau khi click chuột button sẽ gửi request cho View, sau đó sẽ trả về form đăng kí trên màn hình.
* Người dùng nhập username vào form, thông tin sẽ được gửi về UserController.
* Từ Controller sẽ gọi đến lệnh validUser() của UserDao.Từ UserDao sẽ thực hiện câu lệnh query xuống Database để kiểm tra xem username có tồn tại hay chưa
* Sau khi thự hiện câu lệnh, kết quả sẽ được gửi về cho UserDao và UserController.Nếu username tồn tại thì UserController sẽ render lên View, View sẽ trả về màn hình thông báo username không hợp lệ, yêu cầu người dùng nhập lại một username khác.Còn TH username chưa tồn tại thì view sẽ trả về màn hình thông báo hợp lệ.
* Người dùng nhập các thông tin còn lại và nhấn vào button đăng kí, yêu cầu sẽ được gửi cho UserController. Từ Controller sẽ gọi đển lệnh addUser() của UserDao. Từ UserDao thực hiện câu lệnh query xuống database và thông tin tài khoản người dùng sẽ được lưu vào data.
* Nếu thêm vào thành công thì sẽ gửi thông báo lần lượt cho UserDao, UserController, và thông báo đăng kí thành công tại trang đăng kí.

Dòng sự kiện ngoại lệ:

[User nhập sai định dạng thông tin cần nhập]

Người dùng đăng kí tài khoản cần nhập đúng định dạng của các thông tin như : tên đăng nhập phải có độ dài hơn 5 chữ cái (phải là chữ hoặc chữ và số), mật khẩu phải có độ dài hơn 6 phải có chữ và số. Nếu user nhập sai hệ thống sẽ báo cho user cần nhập lại cho đúng định dạng.

1. Đăng nhập tài khoản

|  |  |
| --- | --- |
| Tên | Đăng nhập tài khoản |
| Actors | User |
| Tóm tăt | Người dùng muốn sử dụng hệ thống website để học bài |
| Điều kiện tiên quyết | Người dùng đã đăng kí tài khoản. |
| Sự kiện bắt đầu | Đang ở giao diện đăng nhập, các chức năng khác không được hiển thị. |
| Sự kiện kết thúc | nếu use case thành công actor có thể sử dụng các chức năng được phân quyền. |

Dòng sự kiện chính:

* Use case này bắt đầu khi một người dùng muốn đăng nhập vào hệ thống.
* Người dùng đang ở trang chủ, click vào button đăng nhập. Yêu cầu sẽ được gửi cho View, sau đó sẽ trả về form đăng nhập lên màn hình.
* Người dùng nhập Username và password vào form đăng nhập rồi bấm button login.Thông tin đăng nhập sẽ được gửi về UserController.
* Từ Controller gọi phương thức checkLogin() của UserDao. Từ UserDao thực hiện câu lệnh query xuống database xem thông tin đăng nhập của người dùng có đúng hay không.
* Nếu kiểm tra không hợp lệ thì lần lượt gửi thông báo cho UserDao và UserController, thông báo đăng nhập không thành công sẽ được hiện lên màn hình, yêu cầu người dùng nhập lại thông tin đăng nhập.
* Nếu kiểm tra hợp lệ thì lần lượt gửi thông báo cho UserDao và UserController, thông báo đăng nhập thành công sẽ được hiện lên màn hình

Dòng sự kiện ngoại lệ:

[ User nhập sai thông tin ]

nếu trong dòng sự kiện chính, người dùng nhập sai tên đăng nhập, mật khẩu, hệ thống sẽ hiển thị thông báo lỗi. Người dùng có thể chọn trở về đầu dòng sự kiện chính hoặc

hủy bỏ việc đăng nhập, lúc này usecase kết thúc

1. Thêm bài học

|  |  |
| --- | --- |
| Tên | Thêm bài học |
| Actors | Admin |
| Tóm tắt | Người admin sẽ thêm bài học dựa vào thông tin đã được chuẩn bị từ trước. |
| Điều kiện tiên quyết | Người admin phải đăng nhập bằng phân quyền admin. |
| Sự kiện bắt đầu | Người admin nhấn vào nút thêm bài học. |
| Sự kiện kết thúc | Bài học được lưu vào cơ sở dữ liệu |

Dòng sự kiện chính:

* Tại giao diện danh sách bài học của Admin sẽ có một nút chọn thêm bài học hoặc là thêm bài kiểm tra.
* Admin chọn vào phần thêm bài học hoặc là thêm bài kiểm tra.
* Nếu là bài học thì người dùng thêm vào nội dung, hình ảnh, âm thanh và chủ đề. Nếu là bài kiểm tra thì người dùng điền vào câu hỏi, hình ảnh, âm thanh, chủ đề, 4 đáp án và câu trả lời đúng.
* Yêu cầu: nội dung là dạng chuỗi có độ dài tùy ý. Hình ảnh và âm thanh là link dạng nhúng, có thể sử dụng Google Drive để làm link nhúng vào. Chủ đề thì chọn trong các lựa chọn có sẵn.
* Đối với bài học:
  + Các thông tin về nội dung, hình ảnh âm thanh và chủ đề sẻ được gửi về trang LessonController.
  + Tại LessonController sẽ mapping những thông tin được gửi về thảnh một Lesson Model.
  + Tại LessonController sau khi có Lesson Model sẽ gọi đến LessonService để gọi hàm saveLesson(Lesson).
  + Tại LessonService sẽ gọi đến LessonDAO sử dụng hàm saveLesson(Lesson).
  + Tại LessonDAO sẽ sử dụng câu lệnh Insert xuống DataBase.
  + Nếu insert thất bại sẽ trả về thông báo lần lượt cho LessonDAO, LessonService, LessonController và hiển thại thất bại tại trang thêm bài học.
  + Nêu Insert thành công sẽ trả về thông báo thành công cho lần lượt LessonDAO, LessonService, LessonController và hiển thị thông báo thành công tại trang thêm bài học.
* Thông tin về bài học sẽ được lưu trong DataBase.
* Đối với bài kiểm tra:
  + Các thông tin về câu hỏi, hình ảnh, âm thanh, chủ đề, 4 đáp án và đáp án đúng sẽ được gửi về trang LessonController.
  + Tại LessonController sẽ mapping những thông tin được gửi về thành một Test Model.
  + Tại LessonController sẽ gọi đến LessonService để sử dụng hàm saveTest(test).
  + Tại LessonController sẽ gọi đến LessonDAO để sử dụng hàm saveTest(test).
  + Tại LessonController sẽ insert xuống DataBase để thêm mới bài kiểm tra.
  + Nếu insert thất bại sẽ trả về thông báo lần lượt cho LessonDAO, LessonService, LessonController và hiển thại thất bại tại trang thêm bài học.
  + Nêu Insert thành công sẽ trả về thông báo thành công cho lần lượt LessonDAO, LessonService, LessonController và hiển thị thông báo thành công tại trang thêm bài học.
* Thông tin sẽ được lưu trong DataBase.

Dòng sự kiện phụ:

[ Người dùng muốn thay đổi từ thêm bài học thành thêm bài kiểm tra và ngược lại ]

* Người dùng đang ở giao diện thêm bài học hoặc thêm bài kiểm tra.
* Người dùng nhấn quay lại và chọn lại phần mình muốn thêm.
* Thực hiện dòng sự kiện chính.

Dòng sự kiện ngoại lệ:

[Người admin thoát ra trong lúc nhập thông tin]

* Toàn bộ thông tin không được lưu lại và không được lưu xuống cơ sở dữ liệu

[Không thể lưu bài xuống cơ sở dữ liệu]

* Toàn bộ thông tin không được lưu xuống cơ sở dữ liệu.
* Xuất ra thông báo tại màn hình lưu bài là đã lưu bài thất bại.

1. Chỉnh sửa bài học

|  |  |
| --- | --- |
| Tên | Chỉnh sửa bài học |
| Actors | Admin |
| Tóm tăt | Người dùng mong muốn thay đổi nội dung bài học hoặc nội dung bài kiểm tra. |
| Điều kiện tiên quyết | Người admin phải đăng nhập vào hệ thống với phân quyền admin. |
| Sự kiện bắt đầu | Người admin chọn vào phần chỉnh sửa trong giao diện của bài học hay câu hỏi. |
| Sự kiện kết thúc | Thông tin được chỉnh sửa và lưu vào cơ sở dữ liệu. |

Dòng sự kiện chính:

* Tại giao diện danh sách bài học, bài kiểm tra, người dùng sẽ nhấn vào bài học, bài kiểm tra.
* Từ trang danh sách sẽ gửi về cho trang LessonController một ID của bài.
* Từ LessonController sẽ gọi đến LessonService để sử dụng hàm findById(id).
* LessonService sẽ gọi đến LessonDAO để sử dụng hàm findById(id).
* Từ LessonDAO sẽ query đến DataBase để lấy về thông tin bài học theo mã ID được cung cấp từ trước.
* DataBase sẽ trả về một Lesson/Test model.
* Model sẽ được gửi về LessonDAO đếm LessonController và hiển thị thông tin lên trang chi tiết bài học/ bài kiểm tra.
* Tại trang chi tiết bài học người Admin sẽ thực hiện thao tác thay đổi các thông tin mà mình muốn thay đổi.
* Tại trang chi tiết bài học người admin thay đổi thông tin theo mong muốn của mình và nhấn vào nút lưu thay đổi.
* Nếu là bài học
  + Các thông tin về nội dung, hình ảnh âm thanh và chủ đề sẻ được gửi về trang LessonController.
  + Tại LessonController sẽ mapping những thông tin được gửi về thảnh một Lesson Model.
  + Tại LessonController sau khi có Lesson Model sẽ gọi đến LessonService để gọi hàm adjustLesson(Lesson).
  + Tại LessonService sẽ gọi đến LessonDAO sử dụng hàm adjustLesson(Lesson).
  + Tại LessonDAO sẽ sử dụng câu lệnh Update xuống DataBase.
  + Nếu insert thất bại sẽ trả về thông báo lần lượt cho LessonDAO, LessonService, LessonController và hiển thại thất bại tại trang thêm bài học.
  + Nêu Insert thành công sẽ trả về thông báo thành công cho lần lượt LessonDAO, LessonService, LessonController và hiển thị thông báo thành công tại trang thêm bài học và thông tin sẽ được cập nhật trong DataBase.
* Nếu là bài kiểm tra
  + Các thông tin về câu hỏi, hình ảnh, âm thanh, chủ đề, 4 đáp án và đáp án đúng sẽ được gửi về trang LessonController.
  + Tại LessonController sẽ mapping những thông tin được gửi về thành một Test Model.
  + Tại LessonController sẽ gọi đến LessonService để sử dụng hàm saveTest(test).
  + Tại LessonController sẽ gọi đến LessonDAO để sử dụng hàm saveTest(test).
  + Tại LessonController sẽ insert xuống DataBase để thêm mới bài kiểm tra.
  + Nếu insert thất bại sẽ trả về thông báo lần lượt cho LessonDAO, LessonService, LessonController và hiển thại thất bại tại trang thêm bài học.
  + Nêu Insert thành công sẽ trả về thông báo thành công cho lần lượt LessonDAO, LessonService, LessonController và hiển thị thông báo thành công tại trang thêm bài học và thông tin sẽ được cập nhật trong DataBase.
* Hệ thống sẽ hiện thông báo thành công và lưu thay đổi vào cơ sở dữ liệu.

Dòng sự kiện ngoại lệ:

[ Người admin thoát ra trong lúc thực hiện thay đổi ]

* Toàn bộ thông tin vẫn được giữ nguyên trong cơ sở dữ liệu.

1. Hiển thị bài học, bài kiểm tra

|  |  |
| --- | --- |
| Tên | Hiển thị bài học, bài kiểm tra |
| Actors | Người dùng |
| Tóm tăt | Người dùng muốn xem nội dung bài học, bài kiểm tra của 1 chủ đề trong hệ thống |
| Điều kiện tiên quyết | Người dùng phải đăng nhập vào tài khoản đã đăng ký trên hệ thống |
| Sự kiện bắt đầu | Người dùng chọn vào 1 chủ đề mà mình muốn học |
| Sự kiện kết thúc | Thông tin bài học, bài kiểm tra sẽ hiển thị lên |

Dòng sự kiện chính:

* Đối với bài kiểm tra:

+ Người dùng click chuột vào chủ đề mà mình muốn học

+ Hệ thống sẽ chuyển trang hiện tại đến trang nội dung bài học

+ Yêu cầu của người dùng sẽ được chuyển đến class LessonController để được thực hiện

+ LessonController sẽ gọi đến hàm getLesson() của lớp LessonService

+ LessonService sẽ gọi đến hàm getLesson() của LessonDAO để thực hiện truy vấn cơ sở dữ liệu

+ Nếu quá trình thành công: hàm getLesson() sẽ trả về kết quả true cho các lớp và hiển thị nội dung kết quả lên trang nội dung bài học

+ Nếu quá trình thất bại: hàm getLesson() sẽ trả về kết quả false cho các lớp và không hiển thị nội dung lên trang nội dung bài học

* Đối với bài kiểm tra

+ Người dùng click chuột vào button “Do Test” sau khi đã học xong toàn bộ nội dung bài học

+ Hệ thống sẽ chuyển đến trang chứa nội dung bài kiểm tra

+ Yêu cầu của người dùng sẽ được gửi đến class TestController để được thực hiện

+ TestController sẽ gọi đến hàm getTest() của class TestService, TestServive gọi đến hàm getTest() của TestDAO để thực hiện quá trình truy vấn cơ sở dữ liệu

+ Nếu quá trình thành công: hàm getTest() sẽ trả về kết quả true và hiển thị nội dung lên trang nội dung bài kiểm tra

+ Nếu quá trỉnh thất bại: hàm getTest() sẽ trả về kết quả false và không có nội dung nào được hiển thị lên trang nội dung bài kiểm tra

Dòng sự kiện ngoại lệ:

[ người dùng thoát ra trong lúc đang học một chủ đề nào đó có trong hệ thống ]

* Toàn bộ quá trình học trước đó của người dùng sẽ được lưu lại

1. Quản lí tài khoản

|  |  |
| --- | --- |
| Tên | Quản lí tài khoản |
| Actors | Admin |
| Tóm tăt | Chức năng này giúp admin dễ dàng quản lí tất cả các tài khoản có trong hệ thống |
| Điều kiện tiên quyết | Admin phải đăng nhập vào tài khoản admin của mình trên hệ thống (tài khoản này là duy nhất) |
| Sự kiện bắt đầu | Admin chọn vào chức năng mà mình muốn thực hiện: thêm, xóa, hiển thị thông tin |
| Sự kiện kết thúc | Mọi thao tác sau khi admin nhấn lưu hoặc đồng ý sẽ được lưu xuống cơ sở dữ liệu |

Dòng sự kiện chính:

* Admin chọn vào 1 trong 3 chức năng mà mình muốn thực hiện: thêm tài khoản, xóa tài khoản, hiển thị thông tin chi tiết của tài khoản

+ Thêm tài khoản:

++ admin click chuột vào button thêm tài khoản, hệ thống sẽ chuyển trang hiện tại

đến trang điền thông tin

++ admin điền đầy đủ thông tin sau đó nhấn vào button “Save”

++ Toàn bộ thông tin admin điền vào sẽ được gửi đến lớp UserController

++ UserController sẽ gọi đến phương thức saveUser() của lớp UserDAO thông qua lớp UserService

++ phương thức saveUser() sẽ thực hiện insert toàn bộ nội dung xuống cơ sở dữ liệu, nếu insert thành công: thông báo “Thành công!” sẽ được hiển thị lên màn hình, nếu insert không thành công: thông báo” Thất bại!” sẽ được hiển thị lên màn hình.

+ Xóa tài khoản

++ admin click chuột vào button “xóa”, hệ thống sẽ hiển thị hộp thoại thông báo “Bạn có chắc muốn xóa tài khoản này khỏi hệ thống?”

++ nếu admin chọn “OK”, yêu cầu sẽ được gửi đến lớp UserController để xử lí, UserController sẽ gọi phương thức removeUser() của lớp UserDAO thông qua lớp UserService, phương thức removeUser() sẽ tiến hành truy vấn cơ sở dữ liệu để xóa tài khoản mà admin chọn. Nếu thành công sẽ trả về thông báo “Thành công!” lên màn hình, Nếu thất bại sẽ hiển thị thông báo “Thất bại!” lên màn hình.

++ Nếu admin chọn “HỦY”, hệ thống sẽ tự động hủy đi yêu cầu của admin, yêu cầu không được gửi đến lớp UserController, hộp thoại thông báo sẽ biến mất.

+ Xem thông tin chi tiết tài khoản

++ admin click chuột vào button “view”, hệ thống sẽ chuyển trang hiện tại đến trang chứa nội dung

++ yêu cầu của admin sẽ được gửi đến lớp UserController để được xử lí

++ UserController sẽ gọi đến phương thức getInfoUser() của lớp UserDAO thông qua lớp UserService, tại phương thức getInfoUser() sẽ thực hiện truy vấn xuống cơ sở dữ liệu để lấy dữ liệu lên. Nếu thành công, hàm getInfoUser() sẽ trả về kết quả true và nội dung sẽ được hiển thị lên trang thông tin tài khoản. Nếu thất bại, hàm getInfoUser() sẽ trả về kết quả false, trang thông tin tài khoản sẽ không có nội dung hiển thị

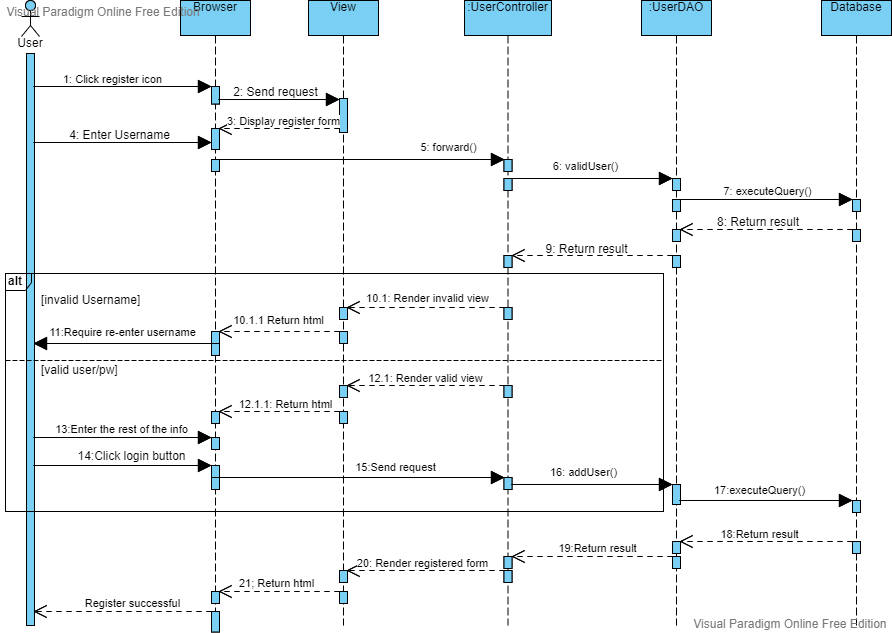
Dòng sự kiện ngoai lệ:

[ admin thoát ra trong lúc đang nhập thông tin tài khoản trong chức năng thêm tài khoản ]

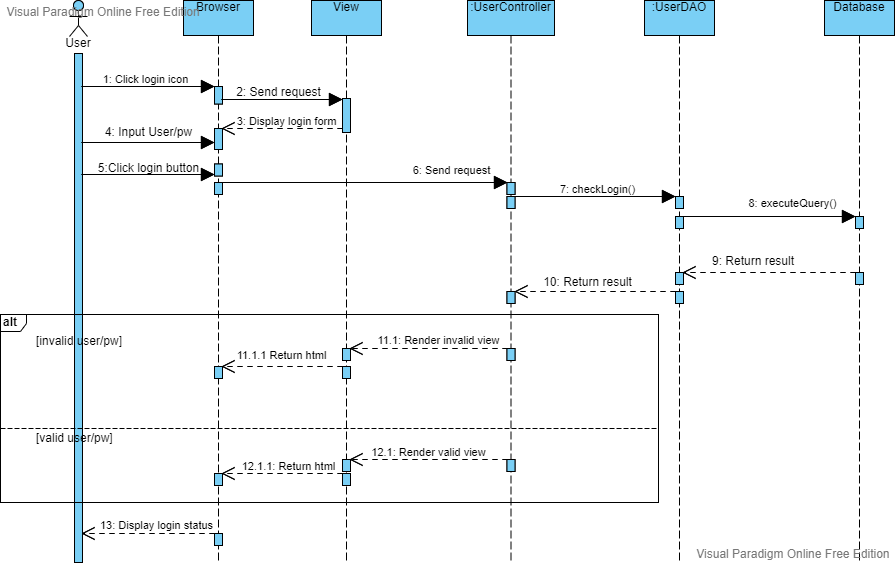
* Tất cả dữ liệu mà admin nhập sẽ bị mất và không được lưu xuống cơ sở dữ liệu

## Sequence diagram

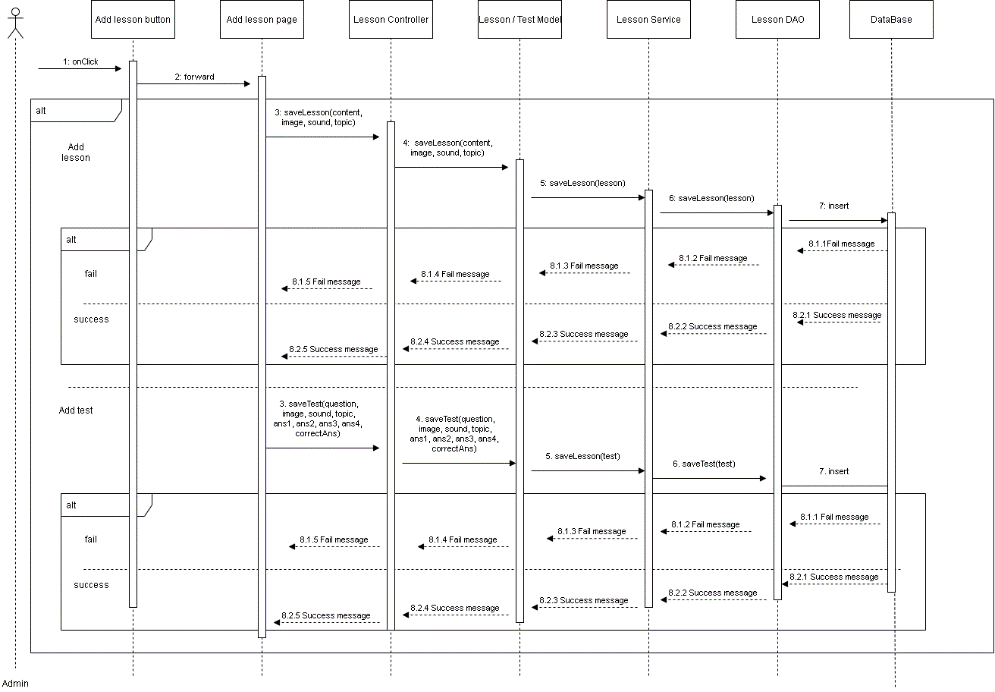
1. Đăng kí tài khoản



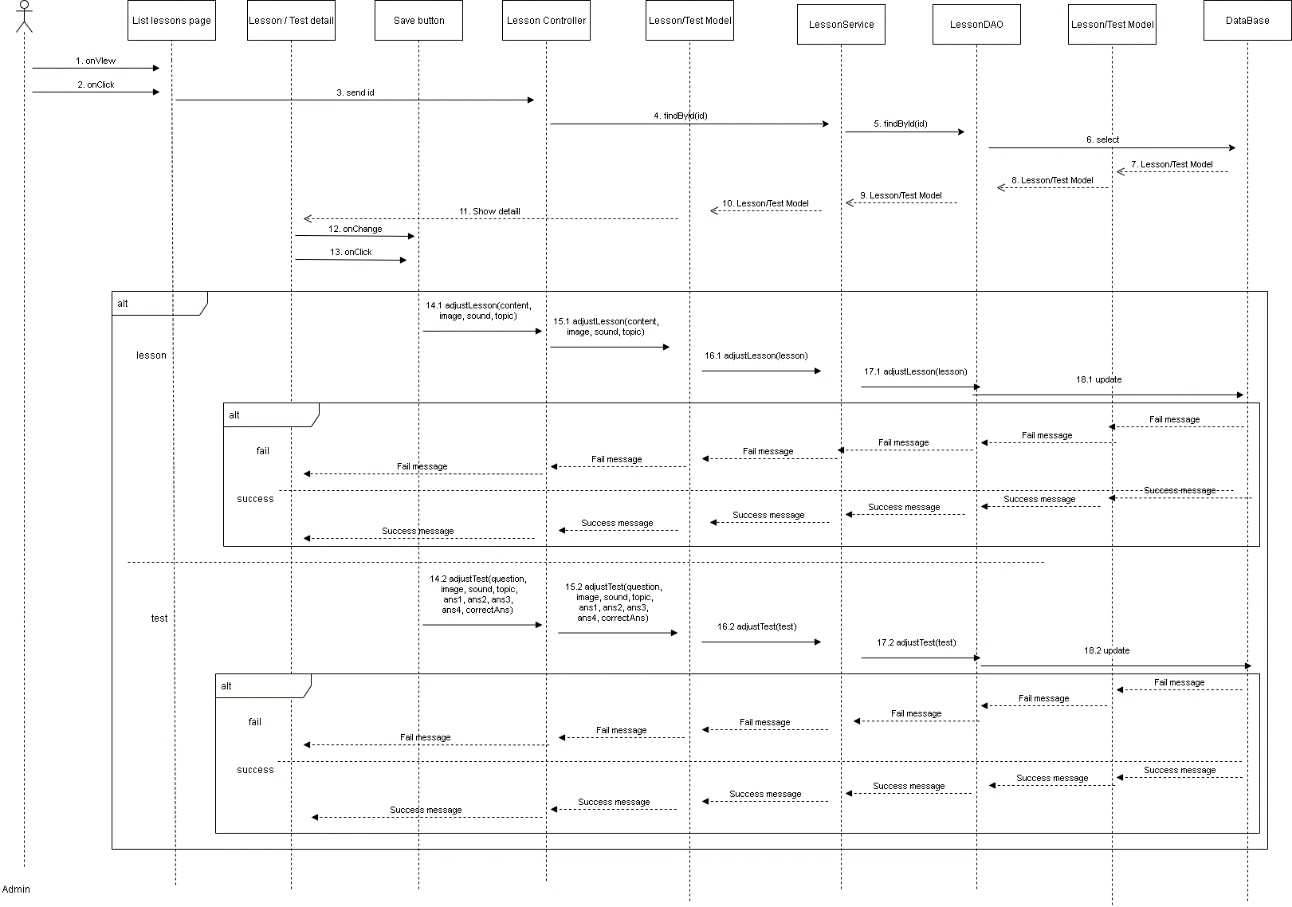
1. Đăng nhập tài Khoản



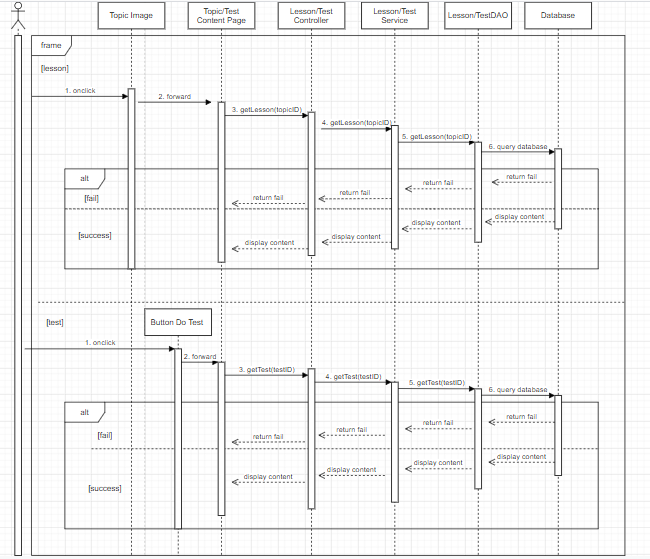
1. Thêm bài học



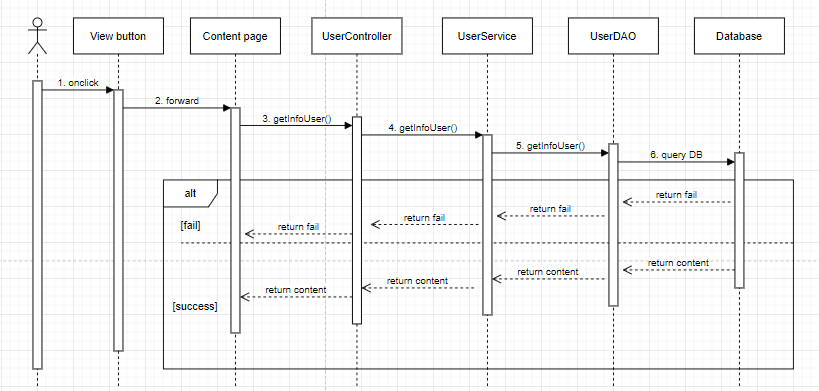
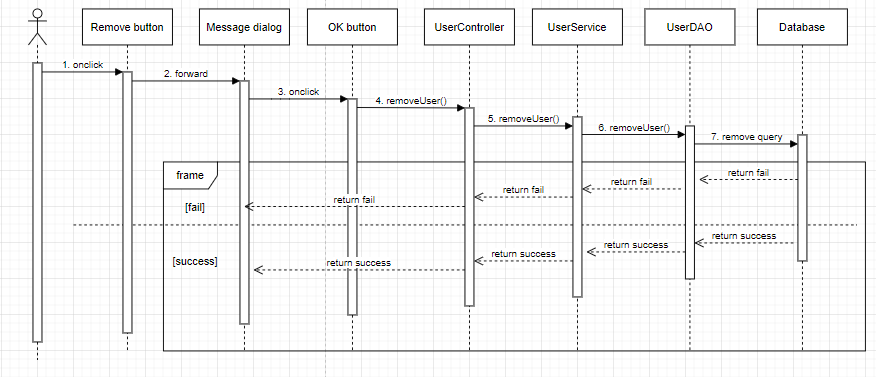
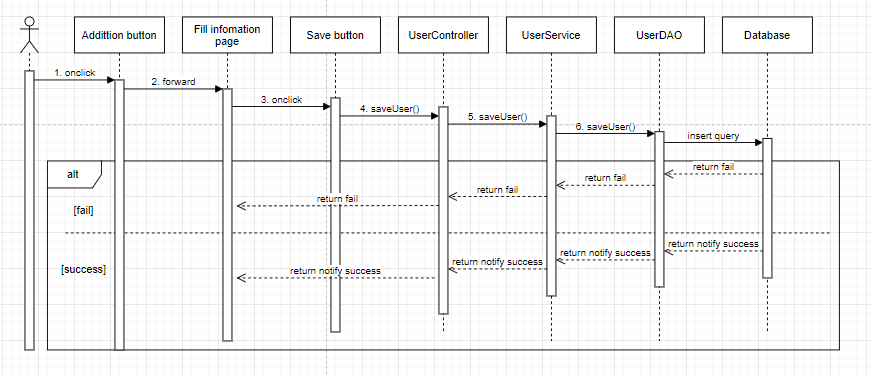
1. Chỉnh sửa bài học



1. Hiển thị bài học, bài kiểm tra

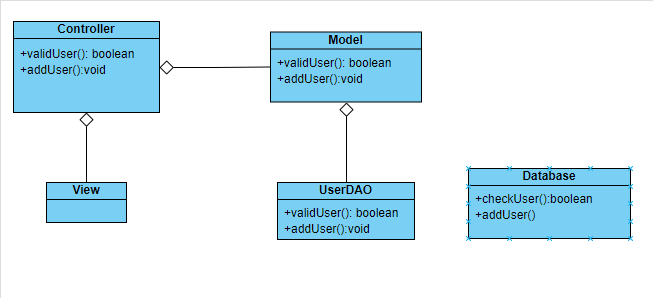


1. Quản lý tài khoản (thêm, xóa, xem thông tin tài khoản)

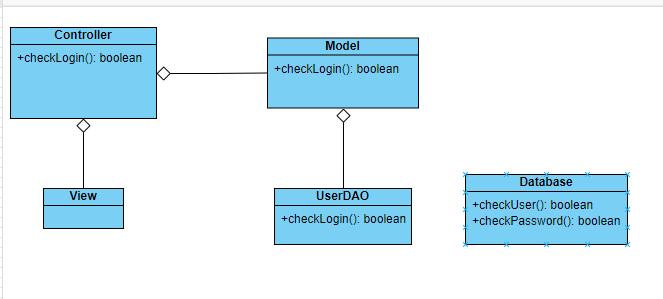


## Class diagram

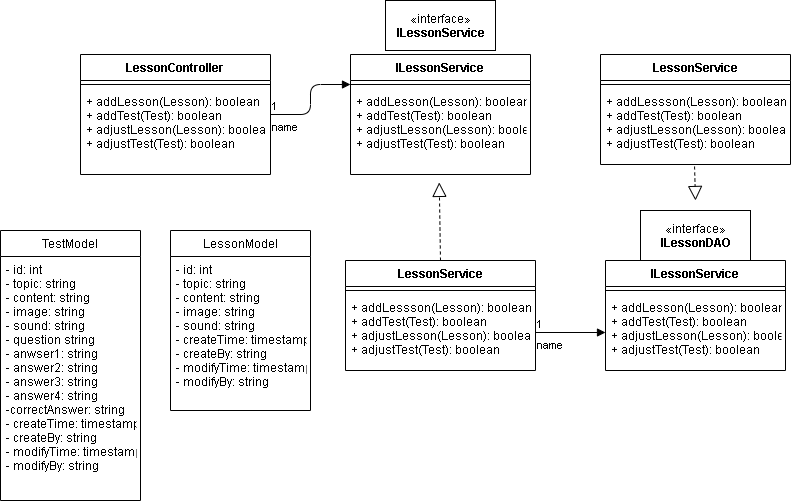
1. Đăng kí tài khoản



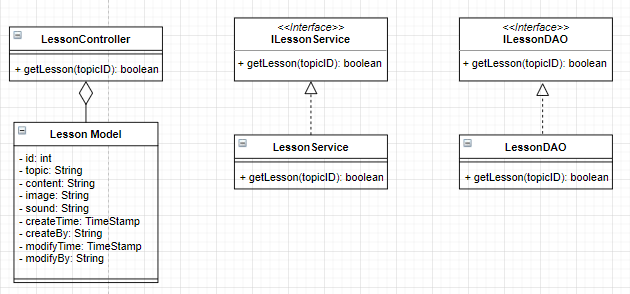
1. Đăng nhập tài khoản



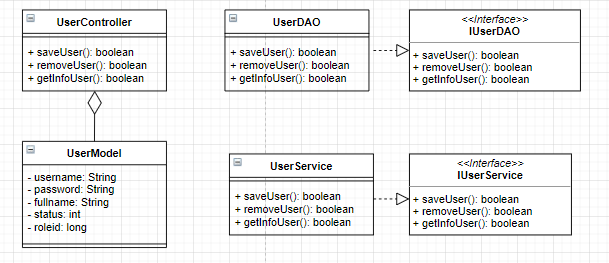
1. Thêm bài học



1. Hiển thị bài học, bài kiểm tra



1. Quản lý tài khoản



## Test case

