**Trần Minh Khoa TÊN ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ PHÓNG HỌP ÁP DỤNG TẠI TRUNG TÂM CODEGYM ĐÀ NẴNG 2022**

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**KHOA CÔNG NGHỆ SỐ**

**ĐỒ ÁN TỐT NGHIỆP**

**ĐẠI HỌC**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI:**

**XÂY DỰNGHỆ THỐNG WEBSITE QUẢN LÝ PHÒNG HỌP ÁP DỤNG TẠI TRUNG TÂM CODEGYM ĐÀ NẴNG**

Người hướng dẫn: **ThS. Đỗ Phú Huy**

Sinh viên thực hiện: **Trần Minh Khoa**

Mã sinh viên: **1811505310221**

Lớp**: 18T2**

**Đà Nẵng, 06/2022**

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**KHOA CÔNG NGHỆ SỐ**

**ĐỒ ÁN TỐT NGHIỆP**

**ĐẠI HỌC**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI:**

**XÂY DỰNG HỆ THỐNG WEBSITE QUẢN LÝ PHÒNG HỌP ÁP DỤNG TẠI TRUNG TÂM CODEGYM ĐÀ NẴNG**

Người hướng dẫn: **ThS. Đỗ Phú Huy**

Sinh viên thực hiện: **Trần Minh Khoa**

Mã sinh viên: **1811505310221**

Lớp**: 18T2**

**Đà Nẵng, 06/2022**

**NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN**

**NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN**

**TÓM TẮT**

Tên đề tài: Xây dựng hệ thống website quản lý phòng họp áp dụng tại Trung Tâm CodeGym Đà Nẵng.

Sinh viên thực hiện: Trần Minh Khoa

Mã SV: 1811505310221 Lớp: 18T2

Đồ án với đề tài Website quản lý phòng họp áp dụng tại trung tâm CodeGym Đà Nẵng được xây dựng với ngôn ngữ lập trình Java, sử dụng Spring framework, Angular và dùng hệ quản trị cơ sở dữ liệu MySQL cùng các plugin hỗ trợ như Bootstrap, JQuery, Ajax.

Website với các chức năng chính như: Cập nhật lịch họp, các phòng đã được đặt. Cho phép xem thông tin chi tiết của mỗi phòng họp. Tính năng lọc danh sách phòng họp theo tên, thời gian để đỡ rối mắt khi lịch họp dày đặc. Có thanh calender cho phép tìm kiếm theo từng ngày.Đặt phòng họp theo nhu cầu.Không cho phép người dùng đặt lịch họp 2 ngày liên tục, mà cần admin cần kiểm duyệt. Cho phép user xem chi tiết tổng quan thông tin chi tiết liên quan về phòng họp. Thông báo trạng thái của phòng, phòng hỏng, sức chứa, hình ảnh từng phòng.Xem tài sản có trong phòng được phép sử dụng. Cho phép thay đổi và chỉnh sửa phòng họp khi trạng thái đang chờ, chưa xét duyệt.Thống kê số lần sử dụng của các phòng theo thời gian tuỳ chỉnh.Phản hồi về phòng họp như điều hoà nóng, hay phản hồi các lỗi kỹ thuật của hệ thống để gửi về cho admin dễ quản lý và giải quyết.Quản lý tài sản trong mỗi phòng họp. Quản lý người dùng và còn rất nhiều tính năng khác.

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Giảng viên hướng dẫn: ThS. Đỗ Phú Huy

Sinh viên thực hiện: Trần Minh Khoa Mã SV: 1811505310221

***1. Tên đề tài:***

Xây dựng hệ thống website quản lý phòng họp áp dụng tại Trung Tâm CodeGym Đà Nẵng

***2. Các số liệu, tài liệu ban đầu:***

* Dựa trên qui định upload tài liệu, các qui tắc đặt câu hỏi.
* Tài liệu tham khảo: <https://levunguyen.com/>

***3. Nội dung chính của đồ án:***

Quá trình xây dựng website quản lý phòng họp áp dụng tại trung tâm CodeGym Đà Nẵng:

* Thu thập thông tin tài liệu liên quan và khảo sát thực tế.
* Phân tích thiết kế các chức năng của hệ thống.
* Phân tích thiết kế cơ sơ dữ liệu.
* Thiết kế giao diện cho các chức năng.
* Xây dựng hệ thống website.
* Kiểm thử chương trình.
* Hoàn thành báo cáo tổng hợp.

***4. Các sản phẩm dự kiến***

* Website quản lý phòng họp.
* File báo cáo hoàn chỉnh.

***5. Ngày giao đồ án: 14/03/2022***

***6. Ngày nộp đồ án: 14/06/2022***

|  |  |
| --- | --- |
|  | *Đà Nẵng, ngày tháng năm 2022* |
| **Trưởng Bộ môn** | **Người hướng dẫn** |

**LỜI NÓI ĐẦU**

Lời đầu tiên, em xin gởi lời cảm ơn chân thành đến quý thầy cô đã giúp đỡ chúng em thực hiện đề tài này. Đặc biệt là ThS. Đỗ Phú Huy đã tận tình giúp đỡ em trong suốt quá trình thực hiện đề tài tốt nghiệp này.

Đồng thời, chúng em cũng xin cảm ơn quý thầy cô thuộc ngành Công Nghệ Thông Tin – Khoa công nghệ số, trường Đại học Sư Phạm Kỹ Thuật - Đại Học Đà Nẵng đã truyền đạt những kiến thức cần thiết và những kinh nghiệm quý báu cho chúng em trong suốt thời gian 4 năm trên giảng đường để em có thể thực hiện tốt đề tài này. Đặc biệt, em xin gởi lời cảm ơn chân thành tới TS. Hoàng Thị Mỹ Lệ, ThS. Đỗ Phú Huy, cùng ThS. Lê Vũ - giáo viên chủ nhiệm lớp 18T2 đã giúp đỡ em rất nhiều trong quá trình học tập và công việc.

Trong quá trình thực hiện đề tài, do kiến thức và thời gian còn hạn chế nên không thể tránh khỏi những sai sót. Vì vậy em mong quý thầy, cô thông cảm và góp ý để em có thể hoàn thiện đề tài. Và những lời góp ý đó có thể giúp em có thể tránh được những sai lầm sau này.

Em xin chân thành cảm ơn!

Đà Nẵng, ngày tháng năm 2022

**Sinh viên thực hiện**

**Trần Minh Khoa**

**LỜI CAM ĐOAN**

Em xin cam đoan:

* Những nội dung trong luận văn này do em thực hiện dưới sự hướng dẫn của thầy Đỗ Phú Huy.
* Mọi tham khảo trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.
* Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, em xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

***Trần Minh Khoa***

**MỤC LỤC**

**DANH SÁCH CÁC BẢNG, HÌNH VẼ**

**DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT**

MỞ ĐẦU

Với nhu cầu cấp thiết đặt ra là cần xây dựng hệ thống quản lý phòng họp cho trung tâm CodeGym Đà Nẵng nhằm tạo điều kiện thuận lợi cho các giảng viên sắp xếp phòng họp được tối ưu hơn, tránh mất thời gian tìm kiếm phòng cho mỗi lần cần sử dụng đến phòng họp.

Phạm vi nghiên cứu đề tài: Website quản lý phòng họp cho trung tâm CodeGym Đà Nẵng

Cấu trúc đồ án tốt nghiệp gồm các phần như sau:

- Mở đầu: Giới thiệu lý do và các mục tiêu mà đề tài cần giải quyết, phạm vi nghiên cứu của đề tài. Giới thiệu tóm tắt nội dung sẽ được trình bày trong các chương tiếp theo.

- Chương 1: Nghiên cứu,tìm hiểu và đưa ra lý do chọn đề tài, mục tiêu và mục đích khi xây dựng và phát triển đề tài. Xác định rõ phạm vi và đối tượng hướng đến, giải pháp công nghệ để triển khai, xây dựng đề tài, đồng thời phân tích đặc tả yêu cầu nghiệp vụ.

- Chương 2: Tìm hiểu, giới thiệu tổng quát các kiến thức về ngôn ngữ lập trình Java, Typescript, Spring Framework, Angular Framework và hệ quản trị cơ sở dữ liệu MySQL.

- Chương 3: Phân tích các tác nhân và chức năng của từng tác nhân, yêu cầu phi chức năng của hệ thống. Thiết kế sơ đồ use-case, sơ đồ hoạt động, sơ đồ ERD. Thiết kế cơ sở dữ liệu. Xây dựng kịch bản cho từng use-case trong hệ thống.

- Chương 4: Xây dựng giao diện và chức năng của hệ thống.

- Kết luận: Kết luận chung cho các chương trong đồ án. Trình bày những vấn đề đã giải quyết đồng thời trình bày hướng phát triển.

CHƯƠNG I. TỔNG QUAN ĐỀ TÀI

* 1. Tổng quan đề tài

Hiện nay, hệ thống quản lý phòng họp tại trung tâm codegym Đà nẵng chưa được triển khai, mỗi khi giảng viên hoặc tutor cần tìm phòng trống để họp thì phải chạy đi kiếm phòng, việc này tốn khá nhiều thời gian và công sức. Từ đó, nhu cầu cấp thiết đặt ra là cần xây dựng hệ thống quản lý phòng họp nhằm tạo điều kiện thuận lợi cho giảng viên và học viên tiết kiệm được thời gian, tránh mất công sức tìm phòng trống.

Từ lý do trên, em chọn đề tài “Website quản lý phòng họp áp dụng cho trung tâm CodeGym Đà Nẵng” làm hướng nghiên cứu cho đề tài.

* 1. Mục tiêu đề tài

Xây dựng website quản lý phòng họp áp dụng cho tại trung tâm CodeGym Đà Nẵng. Với các chức năng cụ thể :

* Đối với user: chỉ có quyền xem thông tin của user đó
  + Xem danh sách các phòng họp, xem thông tin chi tiết từng phòng họp.
  + Tìm kiếm phòng họp (tìm kiếm các phòng trống), lọc danh sách phòng họp theo tên, thời gian.
  + Đăng ký phòng (đăng ký trên thanh calendar hoặc trong item dropdown).
  + Xem lịch sử đăng ký phòng.
  + Thống kê số lần đặt phòng họp (theo danh sách, theo biểu đồ).
  + Xem danh sách phản hồi.
  + Phản hồi phòng họp, phản hồi lỗi kỹ thuật (gửi về cho admin).
  + Xem thông tin cá nhân, đổi mật khẩu, đăng nhập, đăng xuất
  + Xem thông báo.
* Đối với admin: xem được tất cả thông tin của user
  + Quản lý phòng họp (thêm, sửa, xóa, tìm kiếm, xem chi tiết).
  + Tìm kiếm phòng họp (tìm kiếm các phòng trống), lọc danh sách phòng họp theo tên, thời gian.
  + Đăng ký phòng (đăng ký trên thanh calendar hoặc trong item dropdown).
  + Quản lý lịch sử đăng ký phòng (tìm kiếm nâng cao).
  + Thống kê số lần đặt phòng họp (theo danh sách, theo biểu đồ).
  + Xem danh sách phản hồi, tìm kiếm (phản hồi theo user, trạng thái), xóa phản hồi.
  + Xử lý phản hồi phòng họp (gửi lại cho user)
  + Xem thông tin cá nhân, đổi mật khẩu, đăng nhập , đăng xuất
  + Xem thông báo.
  + Quản lý người dùng.
  + Quản lý tài sản trong mỗi phòng họp.
  1. Phạm vi nghiên cứu

Đề tài áp dụng cho trung tâm codegym Đà Nẵng và sau này có thể mở rộng ra phạm vi các trung tâm dạy và học khác, có thể áp dụng tại các trường Đại Học, Cao Đẳng.

* 1. Phương pháp nghiên cứu
     + Phương pháp nghiên cứu lý thuyết thu thập thông tin qua sách, các tài liệu, trang web để tìm được các cơ sở lý thuyết liên quan vấn đề mình nghiên cứu.
     + Phương pháp triển khai thực nghiệp: xây dựng website
  2. Giải pháp công nghệ
     + Ngôn ngữ lập trình: Java - Spring framework, TypeScript – Angular framework.
     + Hệ quản trị cơ sở dữ liệu: MySQL.
     + Công cụ hỗ trợ: StarUML, IntelliJ IDEA, WebStorm, MySQL Workbench
  3. Đặc tả yêu cầu nghiệp vụ
     1. User
* Xem danh sách các phòng họp, xem thông tin chi tiết từng phòng họp.
* Tìm kiếm phòng họp (tìm kiếm các phòng trống), lọc danh sách phòng họp.
* Đăng ký phòng.
* Xem lịch sử đăng ký phòng.
* Thống kê.
* Xem danh sách phản hồi.
* Phản hồi phòng họp, phản hồi lỗi kỹ thuật (gửi về cho admin).
* Xem thông tin cá nhân, đổi mật khẩu, đăng nhập, đăng xuất
* Xem thông báo.
  + 1. Admin
* Quản lý phòng họp.
* Tìm kiếm phòng họp (tìm kiếm các phòng trống), lọc danh sách phòng họp.
* Đăng ký phòng.
* Quản lý lịch sử đăng ký phòng.
* Thống kê số lần đặt phòng họp.
* Xem danh sách phản hồi, tìm kiếm, xóa phản hồi.
* Xử lý phản hồi phòng họp (gửi lại cho user)
* Xem thông tin cá nhân, đổi mật khẩu, đăng nhập , đăng xuất
* Xem thông báo.
* Quản lý người dùng.
* Quản lý tài sản trong mỗi phòng họp.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

2.1. Ngôn ngữ lập trình Java

2.1.1. Java là gì?

Java là một ngôn ngữ lập trình dạng lập trình hướng đối tượng (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy. Cú pháp Java được vay mượn nhiều từ C và C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.[6]

2.1.2. Lịch sử phát triển

Java được khởi đầu bởi James - Gosling và bạn đồng nghiệp ở Sun Microsystems năm 1991. Ban đầu ngôn ngữ này được gọi là Oak (có nghĩa là cây sồi) do bên ngoài cơ quan của ông Gosling có trồng nhiều loại cây này. Java được phát hành vào năm 1994. Sau khi Oracle mua lại công ty Sun Microsystems năm 2009 - 2010, Oracle đã mô tả họ là "người quản lý công nghệ Java với cam kết không ngừng để bồi dưỡng một cộng đồng tham gia và minh bạch". Lịch sử phiên bản java:

* JDK 1.0 (23 tháng 01, 1996)
* JDK 1.1 (19 tháng 2, 1997)
* J2SE 1.2 (Playground) 08 tháng 12, 1998
* J2SE 1.3 (Kestrel) 08 tháng 5, 2000
* J2SE 1.4.0 (Merlin) 06 tháng 02, 2002
* J2SE 5 (1.5.0) (Tiger) 30 tháng 9, 2004
* Java SE 6 (còn gọi là Mustang), được công bố 11 tháng 12 năm 2006.
* Java SE 7 (còn gọi là Dolphin), được bắt đầu từ tháng 8 năm 2006 và công bố ngày 28 tháng 7 năm 2011.
* Java SE 8 ngày 18 tháng 3 năm 2014
* Java SE 9 ngày 21 tháng 9 năm 2017
* Java SE 10 ngày 20 tháng 3 năm 2018
* Java SE 11 ngày 25 tháng 9 năm 2018
* Java SE 12 ngày 19 tháng 3 năm 2019
* Java SE 13 ngày 17 tháng 9 năm 2019
* Java SE 14 ngày 17 tháng 3 năm 2020
* Java SE 15 ngày 15 tháng 9 năm 2020
* Java SE 16 ngày 16 tháng 3 năm 2021
* Java SE 17 ngày 14 tháng 9 năm 2021

2.2. Spring Framework

2.2.1. Đôi nét về lịch sử

Vào năm 2002, Spring Framework phát hành phiên bản đầu tiên bởi Rod Johnson.

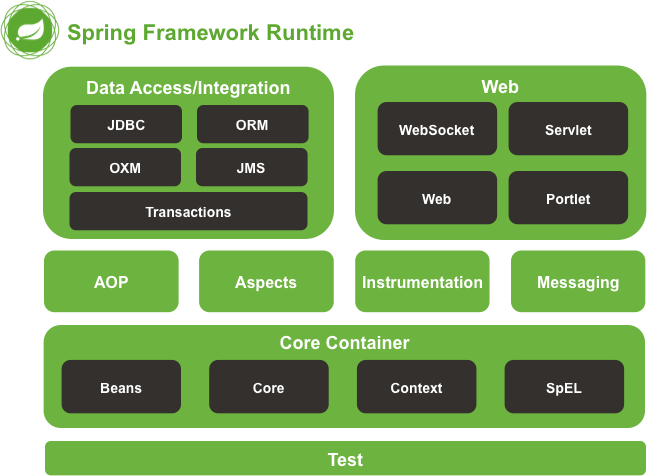
Việc xây dựng các ứng dụng doanh nghiệp trở nên đơn giản và dễ dàng hơn. Rod Johnson đã cho xuất bản một cuốn sách được phổ biến rộng rãi có nhan đề "J2EE Development without EJB". Điều này đã cho Spring Framework rất nhiều động lực để thay thế EJB.[2]

Ngày nay, Spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring Framework.

Để ngăn chặn sự phức tạp trong phát triển các ứng dụng, Spring Framework thường dựa trên các quan điểm như sau:

* Đơn giản hóa công việc phát triển thông qua việc sử dụng các đối tượng Java đơn giản hay còn được gọi là POJO (Plain Old Java Object).
* Nới lỏng ràng buộc giữa các thành phần thông qua việc sử dụng Dependency Injection và viết các interface.
* Tiếp cận lập trình khai báo bằng cách sử dụng các quy tắc (convention) và các khía cạnh (aspect) chung.
* Giảm thiểu các mã nghi thức và soạn sẵn (boilerplate) thông qua việc sử dụng các khuôn mẫu (template) và các khía cạnh.

2.2.2. Tổng quan về Spring

* Spring Core: Spring Core chính là thành phần trung tâm, cốt lõi của Spring Framework. Đây chính là nền tảng để xây dựng nên các thành phần khác.
* Spring Bean: Spring Bean là trung tâm của Spring Core và là trái tim của một ứng dụng Spring.
* Dependency Injection (DI): Dependency Injection (có thể dịch tiêm các thành phần phụ thuộc) là một sức mạnh nổi bật của Spring Framework.
* Spring Context: Spring Context mang mọi thứ lại với nhau.
* Spring Expression Language (SpEL): Spring Expression Language là một ngôn ngữ ngắn gọn giúp cho việc cấu hình Spring Framework trở nên linh hoạt hơn.
* Các dự án trong Spring Framework: Spring Framework là một tập hợp của nhiều dự án con.
* Spring MVC: Spring MVC được thiết kế dành cho việc xây dựng các ứng dụng nền tảng web.
* Spring Data: Cung cấp một cách tiếp cận đúng đắn để truy cập dữ liệu từ cơ sở dữ liệu quan hệ, phi quan hệ, map-reduce, …
* Spring Security: Dự án này cung cấp các cơ chế xác thực (authentication) và phân quyền (authorization) cho ứng dụng.
* Spring Boot: là một framework giúp phát triển cũng như chạy ứng dụng một cách nhanh chóng.
* Spring Batch: tạo các lịch trình (scheduling) và tiến trình (processing).
* Spring Integration: là một implementation của Enterprise Integration Patterns (EIP).
* Spring XD: đơn giản hóa công việc phát triển các ứng dụng Big Data.
* Spring Social: kết nối ứng dụng của bạn với các API bên thứ ba của Facebook, Twitter, Linkedin, ...

Hình 1: Mô hình Spring Framework Runtime

2.3. Tổng quan về hệ quản trị cơ sở dữ liệu MySQL

* MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, …
* MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).[4]

2.4. Restful API (RestController Spring Boot)

* Khác với @Controller là sẽ trả về một template. @RestController trả về dữ liệu dưới dạng JSON. Các đối tượng trả về dưới dạng Object sẽ được Spring Boot chuyển thành JSON.
* Các đối tượng trả về rất đa dạng, bạn có thể trả về List, Map, … Spring Boot sẽ convert hết chúng thành JSON, mặc định sẽ dùng Jackson converter để làm điều đó. Nếu bạn muốn API tùy biến được kiểu dữ liệu trả về, bạn có thể trả về đối tượng ResponseEntity của Spring cung cấp. Đây là đối tượng cha của mọi response và sẽ wrapper các object trả về.
* Vì xây dựng API, nên các thông tin từ phía Client gửi lên Server sẽ nằm trong Body, và cũng dưới dạng JSON luôn. Tất nhiên là Spring Boot sẽ làm giúp chúng ta các phần nặng nhọc, nó chuyển chuỗi JSON trong request thành một Object Java. bạn chỉ cần cho nó biết cần chuyển JSON thành Object nào bằng Annotation @RequestBody
* Ngoài thông tin trong Body của request, thì cái chúng ta cần chính là cái con số 12 nằm trong URL. Phải lấy được con số đó thì mới biết được đối tượng To-do cần thao tác là gì. Lúc đó Anotation @PathVariable tham chiến. [5]

2.5. Bootstrap và responsive

Bootstrap là front-end framework, là một bộ sưu tập miễn phí các công cụ để tạo ra các trang web và các ứng dụng web. Bootstrap bao gồm HTML và CSS dựa trên các mẫu thiết kế cho kiểu chữ, hình thức, các button và các thành phần giao diện khác, cũng như mở rộng tùy chọn JavaScript. Boostrap định nghĩa sẵn các class CSS giúp người thiết kế giao diện website tiết kiệm rất nhiều thời gian. Các thư viện Bootstrap có những đoạn mã sẵn sàng cho chúng ta áp dùng vào website của mình mà không phải tốn quá nhiều thời gian để tự viết. Với Bootstrap, việc phát triển giao diện website để phù hợp với đa thiết bị trở nên dễ dàng hơn bao giờ hết. Bootstrap cung cấp tính năng responsive và mobile first, nghĩa là làm cho trang web có thể tự co giãn để tương thích với mọi thiết bị khác nhau, từ điện thoại di động đến máy tính bảng, máy tính xách tay, máy tính để bàn, ... Một khía cạnh khác là responsive web design làm cho trang web cung cấp được trải nghiệm tuyệt vời cho người dùng trên nhiều thiết bị, kích thước màn hình khác nhau. Một trang có thể hoạt động tốt bất kể sự biến đổi sẽ cung cấp một trải nghiệm người dùng tốt và nhất quán hơn một trang được thiết kế cho một loại thiết bị và kích thước màn hình cụ thể.[1]

2.6. Typescript và Angular Framework

2.6.1. TypeScript

* TypeScript (TS) là một superset của JavaScript (JS), được phát triển bởi Microsoft. Có thể transpile thành code JS để chạy trên môi trường của Browser hoặc Nodejs.
* Tuân thủ chặt chẽ specs mà ECMAScript (ES) đề ra, do đó tất cả những đoạn code hợp lệ trong JS thì sẽ hợp lệ ở TS.
* Support rất nhiều tính năng nâng cao trong các bản ES mới nhất.
* TypeScript đang được hỗ trợ rất mạnh, từ cộng đồng, IDE/Editor, đến các library/framework.

2.6.2. Angular

* Angular được xe là một open source (mã nguồn mở) hay frameworks miễn phí chuyên dụng cho công việc thiết kế web. Angular được phát triển từ những năm 2009 và được duy trì bởi Google. Frameworks này được xem là frameworks front end mạnh mẽ nhất chuyên dụng bởi các lập trình viên cắt HTML cao cấp.[3]
* Angular được ứng dụng rộng rãi với mục đích xây dựng project Single Page Application (SPA). Hiện tại, Version stable của Angular là Angular 9 (released on February 7, 2020) với TypeScript 3.6 và 3.7.
* Kiến trúc Angular:
* Component sử dụng data binding để lấy dữ liệu từ Component sang View (template). Chúng ta sử dụng HTML đặc biệt được biết đến là cú pháp Angular Template.
* Ở bên phải chúng ta có Angular Service. Angular Service cung cấp các service cho Component của chúng ta giống như lấy dữ liệu ra từ database sử dụng TaskService, các sự kiện logging trong ứng dụng sử dụng LoggerService và tạo request HTTP đến backend server sử dụng HTTP Service.
* Trách nhiệm cung cấp các thể hiện của Service cho Component gọi là Angular Injector. Nó inject các service vào component sử dụng Dependency Injection.
* Chúng ta có các Directive, giúp chúng ta thao tác với cấu trúc và style trong ứng dụng. Các directive giúp chúng ta transform DOM theo yêu cầu.
* Component: Trong một trang web đc xây dựng bằng angular, thì chia thành các khối hiển thị và mỗi khối có code xử lý business riêng, mỗi thành phần như vậy được gọi là component. Các thành phần: template + class|properties|method + metadata. Component gồm có: selector, templateUrl, styleUrl
* Template: Component cần một view để hiển thị. Template định nghĩa view. Template chỉ là một tập con của HTML, nó chỉ cho Angular biết làm sao để hiển thị view. Nó là một trang HTML chuẩn sử dụng các thẻ H1, H2 …
* Directive: Là thành phần trong ứng dụng Angular, được dung để tang sức mạnh và mở rộng tính năng cho HTML, Directive giúp người dung thay đổi cấu trúc của thuộc tính của thẻ HTML.
* Module: Angular App được chia thành các Module, chúng được gọi là NgModule. Trong NgModule có thể bao gồm: Component, Pipe, Directive, Service

CHƯƠNG III. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Khảo sát hệ thống.

3.1.1. Khảo sát thực tế

Hiện nay, hệ thống quản lý phòng họp tại trung tâm codegym Đà nẵng chưa được triển khai, mỗi khi giảng viên hoặc tutor cần tìm phòng trống để họp thì phải chạy đi kiếm phòng, việc này tốn khá nhiều thời gian và công sức. Từ đó, nhu cầu cấp thiết đặt ra là cần xây dựng hệ thống quản lý phòng họp nhằm tạo điều kiện thuận lợi cho giảng viên và học viên tiết kiệm được thời gian, tránh mất công sức tìm phòng trống.

3.1.2. Khảo sát người dùng

Trong bối cảnh học viên ngày càng đông, nhu cầu sử dụng phòng lại tăng, mỗi khi cần phòng phải chạy đi chạy lại tìm kiếm phòng trống gây mất rất nhiều thời gian và công sức.

3.2. Đặc tả yêu cầu phần mềm

3.2.1. Xác định các tác nhân

Đề tài gồm 2 tác nhân chính

* Quản trị viên (Admin)
* Người dùng (User)

3.2.2. Các yêu cầu chức năng

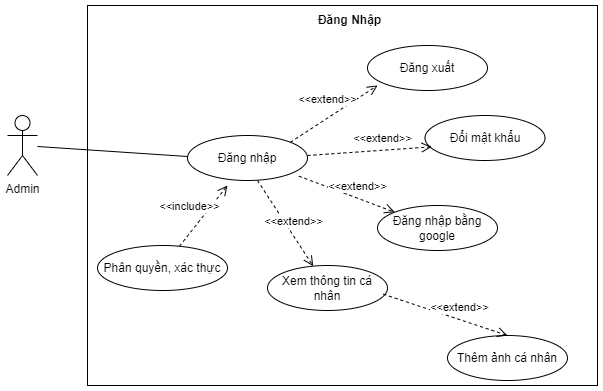
* User
* Xem danh sách các phòng họp, xem thông tin chi tiết từng phòng họp.
* Tìm kiếm phòng họp (tìm kiếm các phòng trống), lọc danh sách phòng họp.
* Đăng ký phòng.
* Xem lịch sử đăng ký phòng.
* Thống kê.
* Xem danh sách phản hồi.
* Phản hồi phòng họp, phản hồi lỗi kỹ thuật (gửi về cho admin).
* Xem thông tin cá nhân, đổi mật khẩu, đăng nhập, đăng xuất
* Xem thông báo.
* Admin
* Quản lý phòng họp.
* Tìm kiếm phòng họp (tìm kiếm các phòng trống), lọc danh sách phòng họp.
* Đăng ký phòng.
* Quản lý lịch sử đăng ký phòng.
* Thống kê số lần đặt phòng họp.
* Xem danh sách phản hồi, tìm kiếm, xóa phản hồi.
* Xử lý phản hồi phòng họp (gửi lại cho user)
* Xem thông tin cá nhân, đổi mật khẩu, đăng nhập , đăng xuất
* Xem thông báo.
* Quản lý người dùng.
* Quản lý tài sản trong mỗi phòng họp

3.2.3. Yêu cầu phi chức năng

* Giao diện đơn giản, dễ sử dụng.
* Có tính bảo mật, an toàn.
* Tốc độ xử lý nhanh chóng, dễ dàng.

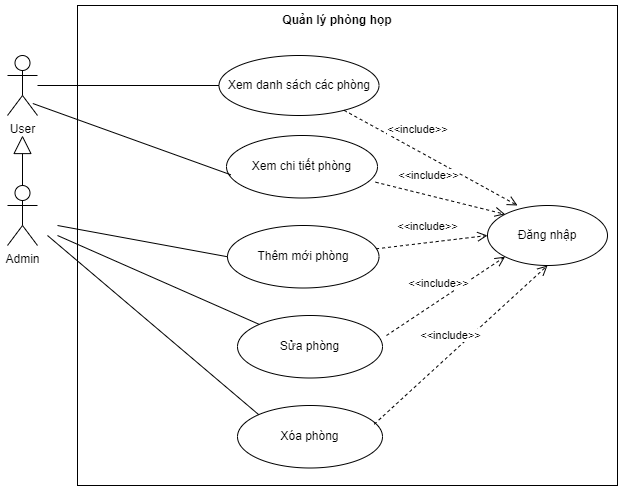
3.3. Biểu đồ Use Case

3.3.1 Đăng nhập



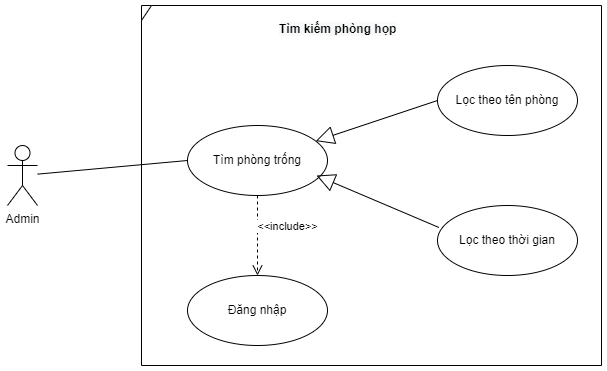
Hình 3.3.1 Use case đăng nhập

3.3.2 Quản lý phòng họp



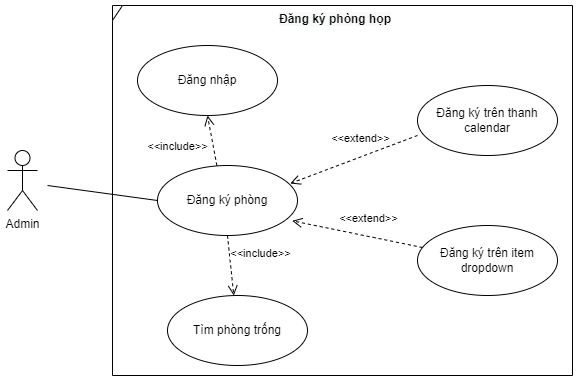
Hình 3.3.2 Use case quản lý phòng họp

3.3.3 Tìm kiếm phòng họp



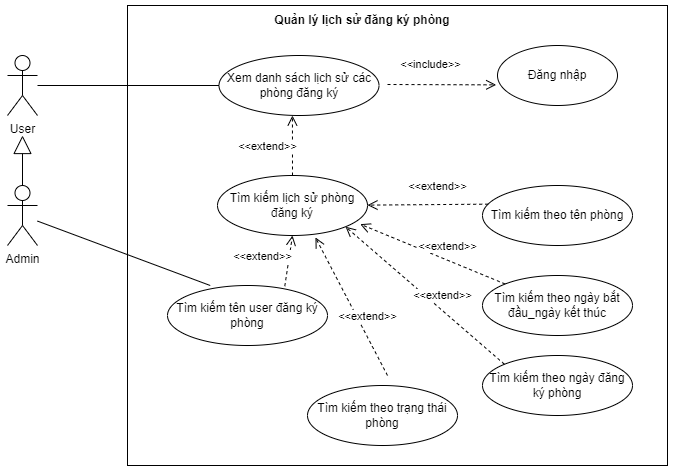
Hình 3.3.3 Use case tìm kiếm phòng họp

3.3.4 Đăng ký phòng họp



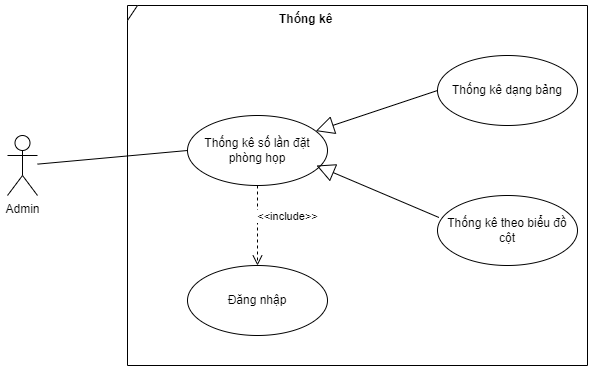
Hình 3.3.4 Use case đăng ký phòng họp

3.3.5 Quản lý lịch sử đăng ký phòng



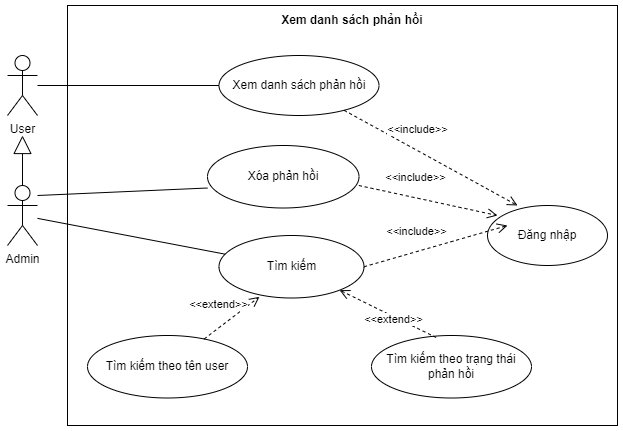
Hình 3.3.5 Use case lịch sử đăng ký phòng

3.3.6 Thống kê



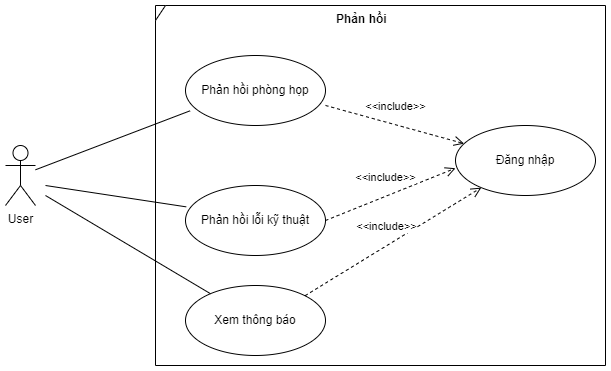
Hình 3.3.6 Use case thống kê

3.3.7 Xem danh sách phòng họp



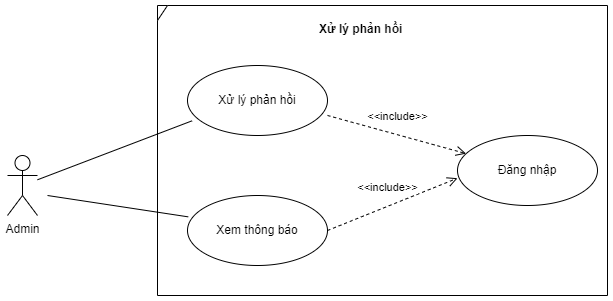
Hình 3.3.7 Use case xem danh sách phòng họp

3.3.8 Phản hồi



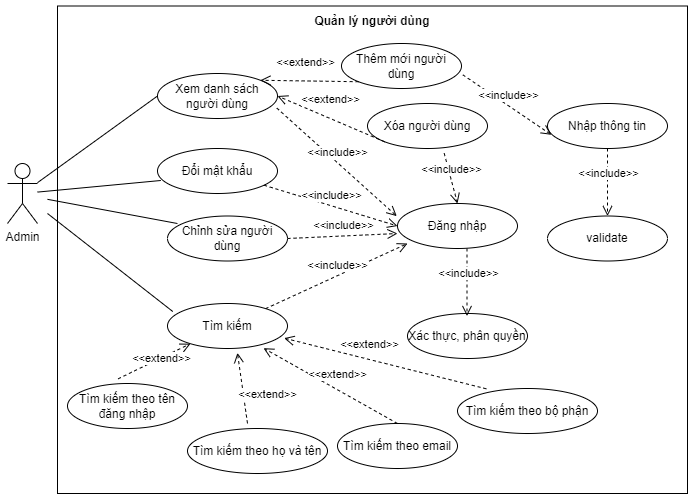
Hình 3.3.8 Use case phản hồi

3.3.9 Xử lý phản hồi



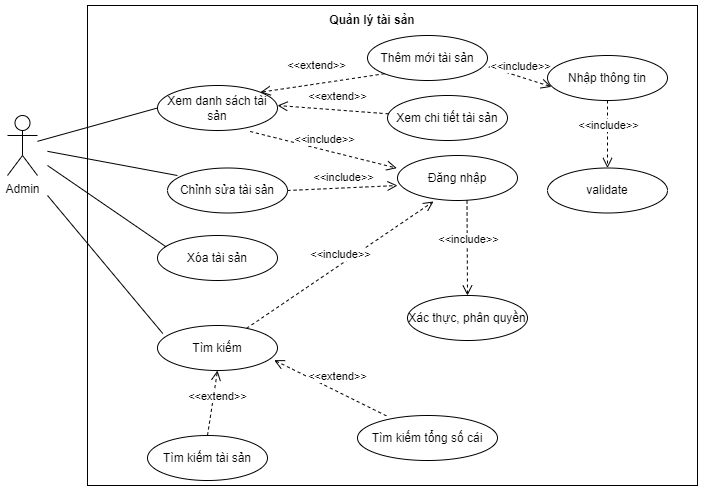
Hình 3.3.9 Use case xử lý phản hồi

3.3.10 Quản lý người dùng



Hình 3.3.10 Use case quản lý người dùng

3.3.11 Quản lý tài sản



Hình 3.3.11 Use case quản lý tài sản

3.4. Scenario cho Use Case

3.4.1 Scenario Use-case “Đăng nhập”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Đăng nhập |
|  | Description | Người dùng đã có tài khoản muốn đăng nhập vào Ứng dụng |
|  | Actors | User/Admin |
|  | Input | User/Admin đã có tài khoản (gmail)  Thông tin tài khoản |
|  | Output | Hệ thống thông báo đăng nhập thành công |
|  | Basic flow | 1. User vào ứng dụng 🡺 Bắt đầu use case 2. Actor nhập thông tin tài khoản (tên tài khoản, mật khẩu), hoặc chọn đăng nhập với google 3. Actor nhấn nút đăng nhập 4. Hệ thống kiểm tra và trả về trang Trang chủ (đã đăng nhập) 🡺 Kết thúc use case |
|  | Alternative flow  (nêu các bước phát sinh trong Basic flow) |  |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 2b.1 Actor nhấn thoát ứng dụng 🡺 Use case kết thúc mà chưa được đăng nhập  3b. Nhập sai tài khoản hoặc mật khẩu (hoặc mật khẩu không khớp với tài khoản) 🡺 Thông báo sai tài khoản hoặc mật khẩu 🡺Quay lại bước 2 |

Bảng 3.4.1 Scenario Use-case “Đăng nhập”

3.4.2 Scenario Use-case “Quản lý phòng họp”

|  |  |  |
| --- | --- | --- |
| **STT** | **Use case name** | **Quản lý phòng họp** |
|  | **Description** | Dùng để quản lý phòng họp của hệ thống |
|  | **Actors** | Admin,User |
|  | **Input** | 1. Actor đã đăng nhập thành công với quyền Admin    1. Thông tin phòng họp muốn cập nhật 2. Actor đã đăng nhập thành công với quyền User |
|  | **Output** | Thông báo cập nhật thành công |
|  | **Basic flow** | 1. Actor nhấn vào “PHÒNG HỌP” ở thanh header của hệ thống 🡺 Use case bắt đầu 2. Thực hiện các thao tác cần thiết (Thêm, Sửa, Xóa, Xem chi tiết phòng họp ) bằng những nút bên cạnh thông tin phòng họp 3. Actor nhấn nút “Lưu” 4. Hệ thống ghi nhận và cập nhật lại trang quản lý phòng họp 🡺 Use case kết thúc |
|  | **Alternative flow (nêu các bước phát sinh trong Basic flow)** | 3a. Hệ thống hỏi actor có chắc chắn với thay đổi hay không. Nếu có thì tiếp tục bước 4. Nếu không thì quay lại bước 3 |
|  | **Exception flow**  **(Nêu các trường hợp ngoại lệ của Basic flow)** | 2b. Nếu actor đăng nhập bằng quyền user chỉ được xem danh sách và xem chi tiết phòng họp không có quyền thêm, sửa, xóa phòng họp. |

Bảng 3.4.2 Scenario Use-case “Quản lý phòng họp”

3.4.3 Scenario Use-case “Tìm kiếm phòng họp”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Tìm kiếm phòng họp |
|  | Description | Người dùng muốn tìm kiếm phòng họp trống |
|  | Actors | Admin/User |
|  | Input | Người dùng đã truy cập vào trang Home |
|  | Output | Hệ thống xuất ra phòng họp còn trống |
|  | Basic flow | 1. Actor nhập thông tin cần tìm kiếm (tên phòng, ngày bắt đầu , ngày kết thúc, giờ bắt đầu, giờ kết thúc , số người tham dự, thiết bị) 2. Actor nhấn nút tìm kiếm 3. Hệ thống kiểm tra và trả kết quả theo yêu cầu tìm kiếm 🡺 Kết thúc use case |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) |  |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 3.a Nếu người dùng không nhập đủ thông tin tìm kiếm thì sẽ không hiện nút tìm kiếm |

Bảng 3.4.3 Scenario Use-case “Tìm kiếm phòng họp”

3.4.4 Scenario Use-case “Đăng ký phòng họp”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Đăng ký phòng họp |
|  | Description | Người dùng muốn đăng ký phòng họp |
|  | Actors | Admin/User |
|  | Input | Người dùng đã đăng nhập vào tài khoản và thực hiện đăng ký phòng |
|  | Output | Hệ thống thông báo đăng ký phòng họp thành công |
|  | Basic flow | 1. Người dùng chọn nút đăng ký phòng họp trên thanh calendar trang Home hoặc chọn đăng ký phòng họp trong item dropdown trên thanh header 🡺 Bắt đầu use case 2. Sau đó hiển thị ra form thông tin đăng ký phòng, người dùng bổ sung thông tin đăng ký phòng 3. Người dùng nhấn đăng ký để xác nhận đăng ký tại phòng đã chọn 4. Hệ thống kiểm tra và trả về thông báo (thành công) 🡺 Kết thúc use case |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) |  |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 3b.1 Actor nhấn hủy 🡺 Use case kết thúc  3b.2 Actor là User nếu đăng ký phòng quá 2 ngày thì hệ thống sẽ gửi thông báo cần xác nhận đăng ký phòng về cho admin, Admin duyệt đăng ký cho phép thì User mới sử dụng được phòng đăng ký. |

Bảng 3.4.4 Scenario Use-case “Đăng ký phòng họp”

3.4.5 Scenario Use-case “Quản lý lịch sử đăng ký phòng”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Quản lý lịch sử đăng ký phòng |
|  | Description | Dùng để quản lý lịch sử đăng ký phòng của hệ thống |
|  | Actors | Admin/User |
|  | Input | 1. Actor đã đăng nhập thành công với quyền admin   1.1 Thông tin lịch sử đăng ký phòng muốn cập nhật   1. Actor đã đăng nhập thành công với quyền User |
|  | Output | Thông báo cập nhật thành công |
|  | Basic flow | 1. Actor nhấn vào “Quản lý lịch sử đăng ký phòng” ở item dropdown của hệ thống 🡺 Use case bắt đầu 2. Actor sẽ thấy danh sách phòng Actor có dùng các công cụ hổ trợ tìm kiếm, tìm kiếm phòng muốn cập nhật 3. Thực hiện các thao tác cần thiết (xem, xóa) bằng những nút bên cạnh thông tin phòng 4. Actor nhấn nút “Đăng ký” 5. Hệ thống ghi nhận và cập nhật lại trang quản lý lịch sử phòng họp🡺 Use case kết thúc |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) | 4a. Hệ thống hỏi actor có chắc chắn với thay đổi hay không. Nếu có thì tiếp tục bước 5. Nếu không thì quay lại bước 3 |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 3b. Chỉ cho phép xóa các phòng chưa sử dụng, nếu xóa các phòng đã sử dụng hệ thống sẽ thông báo không được phép xóa phòng này. |

Bảng 3.4.5 Scenario Use-case “Quản lý lịch sử đăng ký phòng”

3.4.6 Scenario Use-case “Thống kê”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Thống kê |
|  | Description | Dùng để thống kê số lần đặt phòng của hệ thống |
|  | Actors | Admin/User |
|  | Input | 1. Actor đã đăng nhập thành công với quyền admin 2. Actor đã đăng nhập thành công với quyền User |
|  | Output | Hiển thị số lần đặt phòng theo table và theo biểu đồ cột |
|  | Basic flow | 1. Actor nhấn vào “thống kê” trên thanh header của hệ thống 🡺 Use case bắt đầu 2. Thực hiện chọn ngày bắt đầu và ngày kết thúc 3. Hệ thống kiểm tra và hiển thị số lần đặt phòng 🡺 Use case kết thúc |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) |  |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 3.b nếu không tìm thấy phòng hệ thống sẽ hiển thị danh sách rỗng. |

Bảng 3.4.6 Scenario Use-case “Thống kê”

3.4.7 Scenario Use-case “Xem danh sách phản hồi”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Quản lý lịch sử đăng ký phòng |
|  | Description | Dùng để quản lý danh sách phản hồi của hệ thống |
|  | Actors | Admin/User |
|  | Input | 1. Actor đã đăng nhập thành công với quyền admin    1. Thông tin phản hồi muốn cập nhật 2. Actor đã đăng nhập thành công với quyền User |
|  | Output | Thông báo cập nhật thành công |
|  | Basic flow | 1. Actor nhấn vào “phản hồi” ở thanh header của hệ thống 🡺 Use case bắt đầu 2. Actor sẽ thấy danh sách phản. Actor có dùng các công cụ hổ trợ tìm kiếm, tìm kiếm phản hồi muốn cập nhật 3. Thực hiện các thao tác cần thiết (xem,xóa) bằng những nút bên cạnh thông tin phản hồi. 4. Actor nhấn nút “OK” 5. Hệ thống ghi nhận và cập nhật lại trang phản hồi phòng họp🡺 Use case kết thúc |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) | 4a. Hệ thống hỏi actor có chắc chắn với thay đổi hay không. Nếu có thì tiếp tục bước 5. Nếu không thì quay lại bước 3 |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 3b. 1 Chỉ cho phép admin xóa các phản hồi đã xử lý , nếu xóa các phản hồi chưa xử lý hệ thống sẽ thông báo không được phép xóa phản hồi này.  3b. 2 User chỉ xem được các phản hồi của chính user đó gửi phản hồi đi. |

Bảng 3.4.7 Scenario Use-case “Xem danh sách phản hồi”

3.4.8 Scenario Use-case “Phản hồi”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Phản hồi |
|  | Description | Dùng để phản hồi lỗi kỹ thuật hoặc lỗi phòng họp của hệ thống và xem các thông báo sau khi phản hồi được xử lý |
|  | Actors | User |
|  | Input | 1. Actor đã đăng nhập thành công với quyền User |
|  | Output | Thông báo gửi phản hồi thành công  Hiển thị thông tin chi tiết thông báo phản hồi từ admin gửi về |
|  | Basic flow | 1. Actor nhấn vào “Phản hồi lỗi kỹ thuật/Phản hồi phòng họp” ở thanh header của hệ thống 🡺 Use case bắt đầu 2. Thực hiện nhập các thông tin cần phản hồi (tiêu đề,tên phòng, loại lỗi, thông tin lỗi, hình ảnh). 3. Actor nhấn nút “Gửi phản hồi” 4. Hệ thống ghi nhận và gửi phản hồi đến cho admin🡺 Use case kết thúc |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) | 3a. 1 Hệ thống hỏi actor có chắc chắn với thay đổi hay không. Nếu có thì tiếp tục bước 4 . Nếu không thì quay lại bước 2. |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 2b. Khi gửi phản hồi lỗi kỹ thuật bắt buộc phải chọn ảnh, nếu không chọn ảnh button “Gửi phản hồi” sẽ disable. |

Bảng 3.4.8 Scenario Use-case “Phản hồi”

3.4.9 Scenario Use-case “Xử lý phản hồi”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Xử lý phản hồi |
|  | Description | Dùng để xử lý phản hồi lỗi kỹ thuật hoặc lỗi phòng họp của hệ thống và xem các thông báo sau khi nhận phản hồi. |
|  | Actors | Admin |
|  | Input | 1. Actor đã đăng nhập thành công với quyền Admin |
|  | Output | Thông báo gửi xử lý phản hồi thành công  Hiển thị thông tin chi tiết thông báo phản hồi từ User gửi đến. |
|  | Basic flow | 1. Actor nhấn vào “Phản hồi” ở thanh header của hệ thống , tiếp tục chọn icon xử lý 🡺 Use case bắt đầu 2. Thực hiện nhập các thông tin cần xử lý phản hồi (Chọn nhân viên gửi thông báo, Nội dung xử lý). 3. Actor nhấn nút “Xử lý” 4. Hệ thống ghi nhận và gửi xử lý phản hồi đến cho User🡺 Use case kết thúc |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) | 3a. 1 Hệ thống hỏi actor có chắc chắn với thay đổi hay không. Nếu có thì tiếp tục bước 4 . Nếu không thì quay lại bước 2 |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) |  |

Bảng 3.4.9 Scenario Use-case “Xử lý phản hồi”

3.4.10 Scenario Use-case “Quản lý người dùng”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Quản lý người dùng |
|  | Description | Dùng để quản lý người dùng của hệ thống |
|  | Actors | Admin |
|  | Input | 1. Actor đã đăng nhập thành công với quyền admin 2. Thông tin người dùng muốn cập nhật |
|  | Output | Thông báo cập nhật thành công |
|  | Basic flow | 1. Actor nhấn vào “Người dùng” ở header của hệ thống 🡺 Use case bắt đầu 2. Actor sẽ thấy danh sách người dùng. Actor có thể dùng các công cụ hổ trợ tìm kiếm, tìm kiếm tài khoản muốn cập nhật 3. Thực hiện các thao tác cần thiết (Thêm mới, chỉnh sửa, xóa người dùng) bằng những nút bên cạnh thông tin người dùng 4. Actor nhấn nút “OK” 5. Hệ thống ghi nhận và cập nhật lại trang quản lý người dùng 🡺 Use case kết thúc |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) | 4a. Hệ thống hỏi actor có chắc chắn với thay đổi hay không. Nếu có thì tiếp tục bước 5. Nếu không thì quay lại bước 3 |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) |  |

Bảng 3.4.10 Scenario Use-case “Quản lý người dùng”

3.4.11 Scenario Use-case “Quản lý tài sản”

|  |  |  |
| --- | --- | --- |
| STT | Use case name | Quản lý tài sản |
|  | Description | Dùng để quản lý tài sản của hệ thống |
|  | Actors | Admin |
|  | Input | 1. Actor đã đăng nhập thành công với quyền admin 2. Thông tin tài sản muốn cập nhật |
|  | Output | Thông báo cập nhật thành công |
|  | Basic flow | 1. Actor nhấn vào “TÀI SẢN” ở header của hệ thống 🡺 Use case bắt đầu 2. Actor sẽ thấy danh sách tài sản. Actor có thể dùng các công cụ hổ trợ tìm kiếm, tìm kiếm tài sản muốn cập nhật 3. Thực hiện các thao tác cần thiết (Xem chi tiết, Thêm mới, chỉnh sửa, xóa tài sản) bằng những nút bên cạnh thông tin tài sản. 4. Actor nhấn nút “OK” 5. Hệ thống ghi nhận và cập nhật lại trang quản lý tài sản 🡺 Use case kết thúc |
|  | Alternative flow (nêu các bước phát sinh trong Basic flow) | 4a. Hệ thống hỏi actor có chắc chắn với thay đổi hay không. Nếu có thì tiếp tục bước 5. Nếu không thì quay lại bước 3 |
|  | Exception flow  (Nêu các trường hợp ngoại lệ của Basic flow) | 4b. Nếu thực hiện xóa tài sản đang sử dụng, hệ thống sẽ thông báo không được xóa tài sản này, chỉ cho phép xóa những tài khoản chưa được xử dụng. |

Bảng 3.4.11 Scenario Use-case “Quản lý tài sản”

3.5. Sơ đồ hoạt động

3.5.1 Đăng nhập

Hình 3.5.1 Activity “Đăng nhập”

3.5.2 Quản lý phòng họp

Hình 3.5.2 Activity “Quản lý phòng họp”

3.5.3 Tìm kiếm phòng

Hình 3.5.3 Activity “Tìm kiếm phòng”

3.5.4 Đăng ký phòng

Hình 3.5.4 Activity “Đăng ký phòng”

3.5.5 Quản lý lịch sử đăng ký phòng

Hình 3.5.5 Activity “Quản lý lịch sử đăng ký phòng”

3.5.6 Thống kê

Hình 3.5.6 Activity “Thống kê”

3.5.7 Xem danh sách phòng họp

Hình 3.5.7 Activity “Xem danh sách phòng họp”

3.5.8 Phản hồi

Hình 3.5.8 Activity “Phản hồi”

3.5.9 Xử lý phản hồi

Hình 3.5.9 Activity “Xử lý phản hồi”

3.5.10 Quản lý người dùng

Hình 3.5.10 Activity “Quản lý người dùng”

3.5.11 Quản lý tài sản

Hình 3.5.11 Activity “Quản lý tài sản”

3.6. Sơ đồ ERD\_EER

3.7. Thiết kế bảng

**KẾT LUẬN {Cỡ chữ 14**}

{Để 2 dòng trống}

Nội dung kết luận {Font chữ: Time New Roman; in thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

…………………………………………………………………………………………...

**Ghi chú về phần Kết luận**

* Phần Kết luận cần phải nêu được những kết luận chung, khẳng định những kết quả đạt được, những đóng góp, đề xuất và kiến nghị (nếu có);
* Trong phần này, có thể định dạng các điểm/ mục kết luận theo dạng Outline hoặc Numbering hoặc Bullets.

**TÀI LIỆU THAM KHẢO**

{In đậm, cỡ chữ 14}

{Để 2 dòng trống}

{Font chữ: Time New Roman; in thường; cỡ chữ: 13; dãn dòng: 1,3; căn lề: justified}

[1]

[2]

**Ghi chú:** tài liệu tham khảo được liệt kê theo thứ tự trích dẫn trong Báo cáo đồ án tốt nghiệp, đối với từng loại tài liệu thì được ghi thông tin trích dẫn như sau:

1. Tài liệu tham khảo là sách:

Tên của tác giả, năm xuất bản (đặt trong ngoặc đơn). Tên sách (in nghiêng), lần xuất bản, nhà xuất bản, nơi xuất bản.

Ví dụ: Phạm Thị Thu Hà (2014). *Giáo trình Quản lý dự án*, NXB Bách khoa, Hà Nội.

2. Tài liệu tham khảo là tạp chí, bài báo:

Tên tác giả, năm xuất bản (trong ngoặc đơn). Tên bài báo (in nghiêng), Tên tạp chí, tập/quyển, các số trang.

Ví dụ: Nguyễn Danh Nguyên, Phạm Thị Thanh Hồng (2016). *Mô hình sản xuất hiệu suất cao: Đặc điểm và vai trò đối với sự phát triển của nền kinh tế nói chung*, Tạp chí Quản lý kinh tế, Số 75, 73-79.

3. Tài liệu tham khảo từ các nguồn khóa luận, bài luận khác

Tên tác giả, năm xuất bản (đặt trong ngoặc đơn). Tên đề tài luận văn (in nghiêng), bậc học, tên cơ sở đào tạo.

Ví dụ: Nguyễn Đức Núi (2014). *Nghiên cứu xâm nhập mặn nước ngầm khu vực ven biển Hà Tĩnh và đề xuất các giải pháp bảo vệ, sử dụng hợp lý*, Luận văn thạc sỹ, Trường Đại học quốc gia Hà Nội.

4. Tài liệu tham khảo từ nguồn Internet:

Tên tác giả, năm (đặt trong ngoặc đơn). Tên tài liệu (in nghiêng), địa chỉ trang web.

Ví dụ: World Bank (2016). *World Development Indicators Online*, http://publications.worldbank/WDI/.

**PHỤ LỤC**

{In đậm, cỡ chữ 14}

{Font chữ: Time New Roman; in thường; cỡ chữ: 12; dãn dòng: 1,3; căn lề: justified}