

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:
XÂY DỰNG WEBSITE QUẢN LÝ KÝ TÚC XÁ

Người hướng dẫn: **TS. TRƯƠNG NGỌC CHÂU**
Sinh viên thực hiện: **NGÔ VĂN QUỐC**
Mã số sinh viên: **2151220002**
Lớp: **21CT1**

Đà Nẵng, 12/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG WEBSITE QUẢN LÝ KÝ TÚC XÁ

Người hướng dẫn: **TS. TRƯƠNG NGỌC CHÂU**

Sinh viên thực hiện: **NGÔ VĂN QUỐC**

Mã số sinh viên: **2151220002**

Lớp: **21CT1**

Đà Nẵng, 12/2025

TÓM TẮT

Tên đề tài: Xây dựng website Quản lý ký túc xá

Sinh viên thực hiện: Ngô Văn Quốc

Mã số SV: 2151220002 - Lớp: 21CT1

Đề tài tập trung xây dựng một hệ thống website quản lý ký túc xá nhằm hỗ trợ ban quản lý ký túc xá trong việc số hóa quy trình quản lý sinh viên lưu trú. Hệ thống được thiết kế để tự động hóa các nghiệp vụ quan trọng như đăng ký chỗ ở, quản lý sinh viên, quản lý phòng, hợp đồng, hóa đơn, thu phí và thống kê dữ liệu. Việc áp dụng các công nghệ web hiện đại giúp người quản trị thao tác nhanh chóng, chính xác, đồng thời đảm bảo tính bảo mật, ổn định và khả năng mở rộng trong tương lai.

Website được tối ưu hóa trải nghiệm người dùng (UX/UI) cho quản trị viên và cung cấp giao diện đơn giản cho người dùng khác. Bên cạnh đó, hệ thống tích hợp ChatBot hỗ trợ tra cứu thông tin về phòng, quy trình đăng ký và phí dịch vụ, góp phần nâng cao tính tiện lợi và giảm tải khối lượng công việc cho bộ phận quản lý ký túc xá. Nghiên cứu cũng đề xuất các hướng phát triển và bảo trì lâu dài, giúp hệ thống đáp ứng hiệu quả nhu cầu quản lý và nâng cao chất lượng phục vụ sinh viên.

LỜI NÓI ĐẦU

Để hoàn thành được Đồ án tốt nghiệp này, em xin gửi lời cảm ơn chân thành đến quý thầy cô Khoa Công Nghệ Thông Tin – Trường Đại học Kiến Trúc Đà Nẵng, những người đã tận tình giảng dạy và trang bị cho em nền tảng kiến thức quý báu trong suốt những năm học vừa qua. Những kiến thức ấy không chỉ là hành trang quan trọng giúp em hoàn thành đồ án tốt nghiệp mà còn là cơ sở vững chắc để em tự tin bước vào công việc thực tế sau này.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến thầy TS. Trương Ngọc Châu, người đã tận tâm hướng dẫn, hỗ trợ và định hướng cho em trong suốt quá trình thực hiện đề tài. Thầy luôn tạo điều kiện thuận lợi, đưa ra những ý kiến chuyên môn quý báu và giúp em hoàn thiện đề tài một cách tốt nhất.

Trong quá trình thực hiện đồ án, do kiến thức thực tiễn và kinh nghiệm còn hạn chế, chắc chắn không thể tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô để em có cơ hội hoàn thiện bản thân hơn nữa.

Em xin chân thành cảm ơn!

CAM ĐOAN

Em xin cam đoan kết quả đạt được trong đồ án tốt nghiệp là sản phẩm của riêng cá nhân dưới sự hướng dẫn của thầy TS. Trương Ngọc Châu, không sao chép lại của người khác. Trong toàn bộ nội dung của đồ án, những điều được trình bày hoặc là của bản thân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp. Em xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Sinh viên thực hiện

Ngô Văn Quốc

MỤC LỤC

TÓM TẮT

LỜI NÓI ĐẦU	i
CAM ĐOAN.....	ii
MỤC LỤC	iii
DANH SÁCH BẢNG.....	v
DANH SÁCH HÌNH ẢNH	vi
MỞ ĐẦU	1
Chương 1: CƠ SỞ LÝ THUYẾT	3
1.1 Giới thiệu về ngôn ngữ JavaScript	3
1.2 Giới thiệu về ngôn ngữ PHP.....	4
1.3 Giới thiệu về ReactJS	5
1.4 Hệ cơ sở dữ liệu MySQL.....	6
Chương 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG	8
2.1 Xác định yêu cầu	8
2.2 Phân tích yêu cầu.....	8
2.2.1 Chức năng đăng nhập.....	9
2.2.2 Chức năng đăng ký chỗ ở.....	9
2.2.3 Chức năng quản lý sinh viên.....	9
2.2.4 Chức năng quản lý phòng	9
2.2.5 Chức năng quản lý tài chính	10
2.2.6 Chức năng thống kê	10
2.2.7 Chức năng ChatBot.....	11
2.3 Mô hình hóa yêu cầu	11

2.3.1 Chức năng đăng nhập.....	15
2.3.2 Chức năng đăng ký chỗ ở.....	16
2.3.3 Chức năng quản lý sinh viên.....	17
2.3.4 Chức năng quản lý phòng	18
2.3.5 Chức năng quản lý tài chính	19
2.3.6 Chức năng thống kê	20
2.3.7 Chức năng ChatBot.....	21
2.4 Sơ đồ Lớp	22
2.5 Thiết kế cơ sở dữ liệu	22
2.6 Danh mục các bảng cơ sở dữ liệu.....	23
2.7 Thiết kế giao diện	28
2.7.1 Giao diện người dùng.....	28
2.7.2 Giao diện Admin	30
Chương 3: PHÁT TRIỂN WEBSITE	34
3.1 Lập kế hoạch phát triển, phân chia modul, ước lượng thời gian.....	34
3.2 Phát triển frontend	35
3.2.1 Cấu trúc thư mục Frontend	35
3.2.2 Kết nối API	36
3.3 Phát triển backend	39
3.3.1 Cấu trúc thư mục Backend.....	39
3.3.2 Xử lý dữ liệu	41
3.3.3 API	41
3.3.4 Bảo mật	42
KẾT LUẬN	43
TÀI LIỆU THAM KHẢO	45

DANH SÁCH BẢNG

Bảng 2.1 Mô tả các bảng cơ sở dữ liệu	23
Bảng 2.2 Mô tả bảng admin	24
Bảng 2.3 Mô tả bảng student	24
Bảng 2.4 Mô tả bảng room	25
Bảng 2.5 Mô tả bảng contract.....	25
Bảng 2.6 Mô tả bảng invoice.....	25
Bảng 2.7 Mô tả bảng fee.....	26
Bảng 2.8 Mô tả bảng room	26
Bảng 2.9 Mô tả bảng activity_log	26
Bảng 2.10 Mô tả bảng conversation	27
Bảng 2.11 Mô tả bảng chat_message	27
Bảng 2.12 Mô tả bảng faq	27

DANH SÁCH HÌNH ẢNH

Hình 2.1 Biểu đồ Use Case tổng quát.....	11
Hình 2.2 Biểu đồ DFD mức đỉnh	12
Hình 2.3 Biểu đồ DFD mức dưới đỉnh (1)	12
Hình 2.4 Biểu đồ DFD mức dưới đỉnh (2)	13
Hình 2.5 Biểu đồ DFD mức dưới đỉnh (3)	13
Hình 2.6 Biểu đồ DFD mức dưới đỉnh (4)	13
Hình 2.7 Biểu đồ DFD mức dưới đỉnh (5)	14
Hình 2.8 Biểu đồ DFD mức dưới đỉnh (6)	14
Hình 2.9 Biểu đồ DFD mức dưới đỉnh (7)	14
Hình 2.10 Biểu đồ tuần tự Đăng nhập.....	15
Hình 2.11 Biểu đồ tuần tự Đăng ký chỗ ở.....	16
Hình 2.12 Biểu đồ tuần tự Quản lý sinh viên	17
Hình 2.13 Biểu đồ tuần tự Quản lý phòng.....	18
Hình 2.14 Biểu đồ tuần tự Quản lý tài chính.....	19
Hình 2.15 Biểu đồ tuần tự Thống kê	20
Hình 2.16 Biểu đồ tuần tự ChatBot	21
Hình 2.17 Sơ đồ lớp.....	22
Hình 2.18 Sơ đồ Quan hệ Cơ sở dữ liệu.....	23
Hình 2.19 Màn hình Trang chủ	28
Hình 2.20 Màn hình tra cứu phòng	29
Hình 2.21 Màn hình ChatBot	29
Hình 2.22 Màn hình Login	30
Hình 2.23 Màn hình Chính của Admin	31
Hình 2.24 Màn hình Quản lý sinh viên	31
Hình 2.25 Màn hình Quản lý phòng.....	32

Hình 2.26 Màn hình Quản lý tài chính	32
Hình 2.27 Màn hình Quản lý phí.....	33
Hình 2.28 Màn hình Thống kê	33

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Trong bối cảnh chuyển đổi số diễn ra mạnh mẽ trên toàn thế giới, việc ứng dụng công nghệ thông tin vào quản lý ký túc xá sinh viên đang trở thành xu hướng tất yếu. Nhiều nghiên cứu trong và ngoài nước đã chỉ ra rằng các hệ thống quản lý truyền thống dựa trên giấy tờ hoặc xử lý thủ công thường gây ra sai sót, tốn thời gian và khó mở rộng khi số lượng sinh viên tăng cao. Các mô hình quản lý ký túc xá hiện đại tại Hàn Quốc, Singapore hay châu Âu đều áp dụng hệ thống quản lý tập trung, tự động hóa các nghiệp vụ như quản lý phòng, hợp đồng, hóa đơn và giám sát hoạt động của nhân sự. Tại Việt Nam, nhiều trường đại học đã bắt đầu triển khai các giải pháp số hóa, tuy nhiên phần lớn vẫn còn hạn chế về tính tự động hóa, chưa tối ưu trải nghiệm người dùng và thiếu các tính năng hỗ trợ thông minh.

Từ những vấn đề thực tiễn đó, đề tài được thực hiện với mục đích xây dựng một website hỗ trợ quản lý ký túc xá một cách hiệu quả, hiện đại và tiện ích, giúp giảm tải công việc cho ban quản lý, đồng thời nâng cao khả năng tra cứu thông tin và hỗ trợ tự động cho người dùng thông qua hệ thống ChatBot.

2. Mục tiêu đề tài

Đề tài hướng đến việc xây dựng một hệ thống website quản lý ký túc xá đầy đủ nghiệp vụ, thân thiện với người dùng và dễ vận hành. Cụ thể, đề tài đặt mục tiêu phân tích và thiết kế hệ thống dựa trên các yêu cầu quản lý thực tế; xây dựng giao diện người dùng bằng ReactJS; phát triển backend bằng PHP kết nối cơ sở dữ liệu MySQL; và triển khai các chức năng như đăng ký chỗ ở, quản lý sinh viên, phòng ở, hợp đồng, tài chính và thống kê. Ngoài ra, đề tài còn tích hợp ChatBot hỗ trợ người dùng tra cứu thông tin nhanh chóng.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là hệ thống quản lý ký túc xá dành cho sinh viên, bao gồm toàn bộ các nghiệp vụ liên quan đến quản lý thông tin sinh viên, phòng ở, hợp đồng lưu trú, hóa đơn – phí và các hoạt động thống kê phục vụ công tác quản lý. Hệ thống hướng đến mô hình ký túc xá mở, cho phép sinh viên đến từ nhiều trường đại học khác nhau đăng ký ở. Phạm vi nghiên cứu tập trung vào việc xây dựng website với các chức năng dành riêng cho quản trị viên như đăng nhập, quản lý sinh viên, quản

lý phòng, theo dõi hợp đồng, quản lý tài chính và tổng hợp thống kê. Đối với người dùng không có tài khoản, hệ thống chỉ hỗ trợ tra cứu trạng thái phòng trống và sử dụng ChatBot để đặt câu hỏi. Đề tài không bao gồm các tính năng như thanh toán trực tuyến, hệ thống đăng nhập cho sinh viên, ứng dụng di động hay tự động hóa quy trình kiểm tra phòng, nhằm đảm bảo phạm vi phù hợp với thời gian và yêu cầu thực hiện đồ án.

4. Phương pháp nghiên cứu

Về phương pháp nghiên cứu lý thuyết, đề tài tham khảo các tài liệu về thiết kế hệ thống thông tin, kiến trúc phần mềm web, cơ sở dữ liệu quan hệ, cùng các tài liệu chuyên ngành liên quan đến ReactJS, PHP và MySQL. Đồng thời, đề tài tìm hiểu mô hình quản lý ký túc xá trong thực tế để xây dựng nền tảng nghiệp vụ vững chắc.

Ở hướng thực nghiệm, người thực hiện tiến hành khảo sát quy trình quản lý ký túc xá tại các mô hình tương tự, phân tích nhu cầu của ban quản lý và người dùng, từ đó xây dựng các chức năng phù hợp. Quá trình thực nghiệm bao gồm thiết kế giao diện, xây dựng API, kiểm thử chức năng, theo dõi trạng thái dữ liệu và tối ưu hiệu năng hệ thống.

5. Cấu trúc của đồ án tốt nghiệp

Đồ án tốt nghiệp gồm năm chương chính. Chương 1 trình bày các cơ sở lý thuyết liên quan, bao gồm tổng quan về JavaScript, PHP, ReactJS và MySQL. Chương 2 tập trung phân tích yêu cầu, mô hình hóa hệ thống thông qua các biểu đồ như Use Case, DFD, Lớp, tuần tự, đồng thời trình bày thiết kế cơ sở dữ liệu và danh mục các bảng cơ sở dữ liệu được sử dụng. Chương 3 mô tả quá trình cài đặt và phát triển website, bao gồm kế hoạch triển khai, xây dựng frontend bằng ReactJS và phát triển backend bằng PHP kết nối MySQL. Cuối cùng đưa ra các kết luận, trình bày kết quả đạt được, đánh giá hiệu quả hệ thống, phân tích ưu điểm, hạn chế và đưa ra những định hướng phát triển trong tương lai.

Chương 1: CƠ SỞ LÝ THUYẾT

1.1 Giới thiệu về ngôn ngữ JavaScript

Ngôn ngữ JavaScript là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay và đóng vai trò quan trọng trong phát triển web hiện đại. Ban đầu JavaScript được tạo ra vào năm 1995 để xử lý các tương tác đơn giản trên trình duyệt, giúp website trở nên linh hoạt và phản hồi tốt hơn với người dùng. Trải qua nhiều năm phát triển, JavaScript đã mở rộng phạm vi vượt ra ngoài trình duyệt với sự ra đời của các môi trường chạy như Node.js, cho phép lập trình viên sử dụng JavaScript ở cả phía máy khách (client-side) và phía máy chủ (server-side). Ngôn ngữ này có cú pháp linh hoạt, dễ học đối với người mới bắt đầu nhưng vẫn đủ mạnh để xây dựng các hệ thống lớn, đồng thời hỗ trợ mô hình lập trình hướng đối tượng, lập trình hàm và lập trình sự kiện. Nhờ tính đa năng, JavaScript hiện là nền tảng của rất nhiều công nghệ web như AJAX, JSON, Web API và các thư viện hoặc framework mạnh mẽ như React, Angular hay Vue.

Sự phổ biến của JavaScript đến từ khả năng tương tác trực tiếp với DOM, hỗ trợ tạo giao diện người dùng động, kiểm tra dữ liệu đầu vào, xử lý sự kiện và kết nối với API phía server. Ngôn ngữ này còn có hệ sinh thái khổng lồ thông qua NPM (Node Package Manager), giúp lập trình viên dễ dàng cài đặt, tái sử dụng và quản lý các thư viện. JavaScript cũng sở hữu cộng đồng phát triển rộng lớn, tài liệu phong phú và thường xuyên được cập nhật theo chuẩn ECMAScript (ES6, ES7...). Với sự phát triển mạnh mẽ và xu hướng chuyển đổi số hiện nay, JavaScript vẫn giữ vị trí trung tâm trong lập trình web, đảm bảo phù hợp cho cả người mới học và chuyên gia xây dựng các hệ thống quy mô lớn.

Ví dụ code JavaScript:

```
function tinhTong(a, b) {  
    return a + b;  
}  
  
console.log("Tổng bằng: " + tinhTong(1, 2));
```

Lợi ích của JavaScript:

- + Hoạt động trực tiếp trên trình duyệt, không cần biên dịch.
- + Giúp website trở nên tương tác và thân thiện hơn.

- + Hỗ trợ đa nền tảng: web, mobile, desktop (Electron), server (Node.js).
- + Cộng đồng lớn, tài nguyên phong phú, thư viện đa dạng.

1.2 Giới thiệu về ngôn ngữ PHP

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản phía server được sử dụng rộng rãi trong phát triển web. Ra mắt từ năm 1994, PHP nhanh chóng trở thành lựa chọn phổ biến nhờ khả năng nhúng trực tiếp vào HTML và xử lý hiệu quả các tác vụ trên máy chủ như quản lý dữ liệu, xác thực người dùng, xử lý form, và tạo trang web động. PHP hoạt động dựa trên mô hình request – response: trình duyệt gửi yêu cầu (request) đến server, PHP xử lý nội dung cần thiết và trả về kết quả dạng HTML cho người dùng. Nhờ khả năng tích hợp đơn giản với nhiều hệ quản trị cơ sở dữ liệu, đặc biệt là MySQL, PHP được sử dụng rộng rãi trong các website thương mại điện tử, blog, hệ thống quản lý nội dung và nhiều nền tảng web khác.

Một trong những lý do giúp PHP duy trì sự phổ biến lâu dài là do tính dễ học, cú pháp tương đối đơn giản và có tài liệu phong phú. Hệ sinh thái PHP cũng rất lớn với các framework mạnh mẽ như Laravel, CodeIgniter, Symfony, giúp tối giản quy trình phát triển và tăng độ an toàn cho ứng dụng web. Nhiều hệ thống nổi tiếng trên thế giới như Facebook, Wikipedia, WordPress, Moodle... cũng từng xây dựng trên nền tảng PHP, cho thấy độ ổn định và khả năng mở rộng của ngôn ngữ này. PHP có khả năng xử lý hiệu quả các tác vụ liên quan đến bảo mật như mã hóa, xác thực, phân quyền và hỗ trợ kết nối API linh hoạt.

Ví dụ code PHP: <?php

```
function tinhHieu($a, $b) {  
    return $a - $b;  
}  
  
echo "Hiệu bằng: " . tinhHieu(3, 2);  
?>
```

Lợi ích của PHP:

- + Dễ học, dễ triển khai, hỗ trợ đa nền tảng.
- + Tương thích tốt với MySQL và các hệ CSDL khác.
- + Có nhiều framework mạnh mẽ giúp phát triển nhanh hơn.
- + Chi phí triển khai thấp, phù hợp cho website từ nhỏ đến lớn.

1.3 Giới thiệu về ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở do Facebook phát triển nhằm xây dựng giao diện người dùng (UI) theo hướng component. React tập trung vào việc tối ưu UI và mang đến hiệu năng cao thông qua cơ chế Virtual DOM, giúp hạn chế việc cập nhật trực tiếp lên DOM thật và giảm thời gian render. React sử dụng mô hình component hóa, nơi mỗi phần của giao diện được chia thành các thành phần nhỏ, độc lập và dễ tái sử dụng. Điều này giúp các ứng dụng React dễ bảo trì, dễ mở rộng và tổ chức mã nguồn rõ ràng. React cũng hỗ trợ JSX – cú pháp mở rộng cho phép viết HTML trong JavaScript, giúp lập trình viên tiếp cận nhanh hơn và tạo ra UI trực quan.

Ngoài khả năng tái sử dụng component, React còn có cơ chế quản lý trạng thái (state management) mạnh mẽ. Đối với các ứng dụng lớn có nhiều trạng thái phức tạp, lập trình viên có thể kết hợp với thư viện Redux hoặc Context API để kiểm soát luồng dữ liệu hiệu quả. React cũng có thể sử dụng để phát triển ứng dụng mobile thông qua React Native, một framework cho phép xây dựng ứng dụng đa nền tảng Android – iOS từ cùng một codebase. Nhờ đó, React trở thành lựa chọn hàng đầu trong việc phát triển giao diện web hiện đại, đặc biệt đối với các dự án yêu cầu tốc độ cao, UI động và khả năng mở rộng lớn.

Ví dụ code ReactJS:

```
import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);
```

Lợi ích của ReactJS:

- + Tốc độ nhanh nhờ Virtual DOM.
- + Tổ chức mã theo component, dễ mở rộng và bảo trì.
- + Hỗ trợ JSX giúp viết UI thuận tiện.
- + Cộng đồng lớn, thư viện hỗ trợ phong phú.

Khi nào nên sử dụng ReactJS?

- + Khi xây dựng website có giao diện phức tạp, nhiều thành phần thay đổi liên tục.
- + Khi muốn tái sử dụng component và phát triển dự án lớn.
- + Khi cần mở rộng sang mobile với React Native.
- + Khi cần hiệu năng giao diện cao và tối ưu trải nghiệm người dùng.

1.4 Hệ cơ sở dữ liệu MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến nhất trên thế giới, được sử dụng rộng rãi trong nhiều hệ thống web, ứng dụng doanh nghiệp và các nền tảng quản lý dữ liệu lớn. MySQL hoạt động dựa trên mô hình bảng (table) và hỗ trợ truy vấn bằng ngôn ngữ SQL, cho phép lưu trữ, tìm kiếm, cập nhật và quản lý dữ liệu một cách hiệu quả. Với hiệu năng cao, độ ổn định tốt và khả năng xử lý khối lượng dữ liệu lớn, MySQL thường được sử dụng trong các dự án từ nhỏ đến rất lớn như thương mại điện tử, quản trị nội dung, hệ thống ngân hàng, hoặc các nền tảng đòi hỏi tính sẵn sàng cao. MySQL cũng hỗ trợ nhiều thư viện kết nối cho các ngôn ngữ lập trình như PHP, Java, Python.

MySQL cung cấp nhiều tính năng quan trọng như ràng buộc toàn vẹn dữ liệu, khóa ngoại, trigger, stored procedure, view và các khả năng bảo mật như quyền truy cập, mã hóa và ghi log. Hệ quản trị MySQL cũng có khả năng sao lưu dễ dàng và hỗ trợ nhân bản (replication), giúp tăng tính ổn định và đảm bảo an toàn dữ liệu. Một trong những lý do khiến MySQL được ưa chuộng là vì khả năng tích hợp tốt với PHP và Apache trong mô hình LAMP (Linux – Apache – MySQL – PHP), vốn được sử dụng trong hàng triệu website trên thế giới.

Ví dụ truy vấn MySQL:

```
SELECT * FROM STUDENT WHERE room_id = 101;
```

Lợi ích của MySQL:

- + Hiệu năng cao, ổn định, dễ triển khai.

- + Hỗ trợ SQL chuẩn, phù hợp cho các hệ thống quản lý dữ liệu lớn.
- + Miễn phí, tài liệu đầy đủ, cộng đồng đông đảo.
- + Kết nối mạnh mẽ với PHP, Java, Python.

Ứng dụng thực tế của MySQL:

- + Website doanh nghiệp, thương mại điện tử.
- + Hệ thống quản lý sinh viên, nhân sự, tài chính.
- + Các nền tảng CMS như WordPress, Joomla.
- + Ứng dụng lưu trữ dữ liệu lớn.

Một số công cụ hỗ trợ MySQL:

- + phpMyAdmin: Quản lý MySQL qua giao diện web.
- + MySQL Workbench: Công cụ trực quan cho thiết kế CSDL.
- + XAMPP / WAMP: Môi trường chạy PHP – MySQL nhanh chóng.
- + HeidiSQL: Công cụ nhẹ, thao tác MySQL nhanh chóng.

Chương 2: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

2.1 Xác định yêu cầu

Để xây dựng hệ thống quản lý ký túc xá hiện đại, thân thiện và đáp ứng đúng nhu cầu thực tế, bước xác định yêu cầu cần tiến hành phân tích toàn diện các quy trình vận hành của ký túc xá: tiếp nhận sinh viên, phân phòng, quản lý hợp đồng lưu trú, lập và thu các khoản phí, quản lý sự cố và hỗ trợ sinh hoạt cho sinh viên. Mục tiêu là đảm bảo hệ thống mô phỏng chính xác nghiệp vụ, đồng thời tối ưu hóa thao tác cho ban quản lý và cung cấp giao diện đơn giản, dễ sử dụng cho người dùng bên ngoài như sinh viên và phụ huynh.

Kết quả khảo sát cho thấy hệ thống cần hỗ trợ các nghiệp vụ chính: quản lý sinh viên (hồ sơ, trạng thái, lịch sử), quản lý phòng (khu nhà, số phòng, sức chứa, tình trạng), quản lý hợp đồng (tạo, gia hạn, kết thúc), quản lý tài chính (loại phí, tạo hóa đơn, xác nhận thanh toán) và thống kê báo cáo. Một yêu cầu quan trọng là mọi thao tác can thiệp dữ liệu chính (thêm/sửa/xóa sinh viên, đăng ký chỗ ở, tạo hợp đồng, xác nhận thanh toán, quản lý phòng) chỉ do Admin thực hiện thông qua tài khoản quản trị để bảo đảm tính an toàn và toàn vẹn dữ liệu. Người dùng không có tài khoản (guest) chỉ được phép truy cập Trang chủ để xem danh sách phòng và trạng thái phòng (còn trống / đã đầy / bảo trì) và sử dụng ChatBot để tra cứu thông tin (FAQ, quy trình, giá phòng, liên hệ).

Hệ thống cần kiểm tra tự động các ràng buộc nghiệp vụ: không cho đăng ký vào phòng đã đầy, ngăn chặn trùng lặp thông tin (số điện thoại, email), không cho xóa sinh viên khi còn hợp đồng hiệu lực (trường hợp có hợp đồng hiệu lực phải lưu lý do do Admin nhập vào). Về phi chức năng, hệ thống phải đảm bảo bảo hiệu năng (phân trang, tối ưu truy vấn cho thống kê), khả năng mở rộng và responsive để truy cập trên thiết bị di động. Ngoài ra, ChatBot cần cung cấp gợi ý câu hỏi và lưu lịch sử hội thoại phục vụ tra cứu, nhưng không được phép thực hiện hành động thay đổi dữ liệu (nếu có yêu cầu thay đổi, ChatBot sẽ tạo ticket hoặc chuyển tới Admin).

Toàn bộ yêu cầu này là cơ sở để phân tích chi tiết các chức năng, thiết kế kiến trúc hệ thống và xây dựng các API/flow nghiệp vụ ở các phần tiếp theo.

2.2 Phân tích yêu cầu

2.2.1 Chức năng đăng nhập

a. Admin

- Đăng nhập bằng tài khoản quản trị để truy cập giao diện quản lý.
- Sau khi xác thực, Admin có quyền sử dụng toàn bộ chức năng quản lý: đăng ký chỗ ở, quản lý sinh viên, phòng, hợp đồng, hóa đơn, thu phí và thống kê.
- Admin có quyền thêm, sửa, xóa dữ liệu nghiệp vụ theo quyền hạn.

b. Người dùng bình thường (Khách)

- Không có tài khoản đăng nhập; chỉ xem Trang chủ và sử dụng ChatBot để tra cứu.

2.2.2 Chức năng đăng ký chỗ ở

- Admin chọn phòng còn trống, mở form đăng ký, nhập thông tin sinh viên (họ tên, ngày sinh, giới tính, phone, email, trường, khoa, ...), hệ thống kiểm tra ràng buộc (phòng còn chỗ, không trùng phone/email). Nếu hợp lệ, hệ thống tạo hồ sơ sinh viên, tạo hợp đồng, cập nhật số người ở của phòng và sinh hóa đơn liên quan.

2.2.3 Chức năng quản lý sinh viên

- Hiện thị danh sách sinh viên với tổng số lượng, hỗ trợ tìm kiếm theo tên, phòng, trường, khoa, tình trạng tốt nghiệp, trạng thái hợp đồng và phân trang.
- Xem chi tiết hồ sơ và hợp đồng; cập nhật thông tin cá nhân (không được phép thay đổi các thông tin hợp đồng đang hiệu lực).
- Nếu còn hợp đồng hiệu lực, hệ thống yêu cầu nhập lý do rồi xóa và lưu lịch sử thao tác.
- Có thể xem Lịch sử quản lý sinh viên, đồng thời xuất Excel cho cả danh sách sinh viên hoặc lịch sử.

2.2.4 Chức năng quản lý phòng

- Hiện thị danh sách phòng, số lượng sinh viên hiện tại, sức chứa, tình trạng phòng.
- Bộ lọc tìm kiếm theo số phòng, khu nhà, trạng thái; phân trang danh sách phòng.
- Thêm, chỉnh sửa thông tin phòng; khóa phòng khi bảo trì.

- Chỉ được xóa phòng khi không còn người ở và không có hợp đồng nào còn hiệu lực.
- Có thể xem Lịch sử quản lý phòng.

2.2.5 Chức năng quản lý tài chính

- Hiện thị danh sách sinh viên cùng các thông tin thanh toán như số tiền nợ, trạng thái thanh toán,...
- Bộ lọc tìm kiếm theo tên, phòng, trạng thái; phân trang danh sách sinh viên.
- Quản lý các loại phí, có thể tạo, sửa, xóa phí dịch vụ.
- Admin xác nhận thanh toán; tạo, hiển thị phiếu thu PDF; hệ thống lưu lịch sử thanh toán.
- Có thể xem Lịch sử quản lý tài chính, đồng thời xuất Excel cho cả danh sách sinh viên và lịch sử.

2.2.6 Chức năng thống kê

Hệ thống cung cấp một trang Thống kê dành cho Admin với các tính năng:

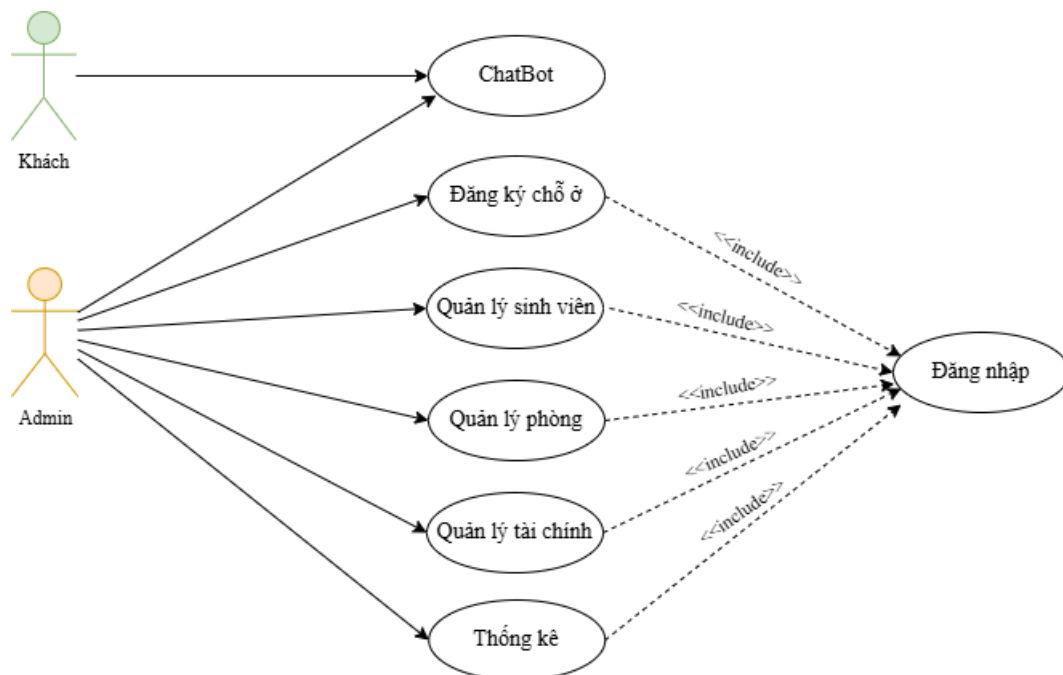
- Bộ lọc theo năm: Người quản trị chọn năm cần xem
- Chỉ số tổng hợp (Insight cards):
 - + Tổng số phòng và phân bố phòng theo trạng thái (Còn trống / Đã đầy / Bảo trì).
 - + Số sinh viên đang lưu trú trong năm (dựa trên hợp đồng hiệu lực).
 - + Số hợp đồng mới, bị hủy, còn hiệu lực, hết hạn trong năm.
 - + Doanh thu đã thu được trong năm, doanh thu trung bình/tháng, tăng trưởng ròng (đăng ký – hủy) và tỷ lệ hủy.
- Biểu đồ trực quan:
 - + Biểu đồ cột phân bố sinh viên theo khu nhà.
 - + Biểu đồ Pie trạng thái phòng.
 - + Biểu đồ cột đăng ký theo 12 tháng.
 - + Biểu đồ đường (hoặc cột) hủy hợp đồng theo 12 tháng.
 - + Biểu đồ đường doanh thu theo 12 tháng

2.2.7 Chức năng ChatBot

- Trả lời tự động các câu hỏi về: giá phòng, quy định ký túc xá, quy trình đăng ký, thời gian đóng phí...
- Có danh sách các câu hỏi gợi ý từ bảng FAQ.
- Luôn tạo phiên trò chuyện mới khi reload trang của khách, luôn giữ lịch sử cuộc trò chuyện của tài khoản admin.

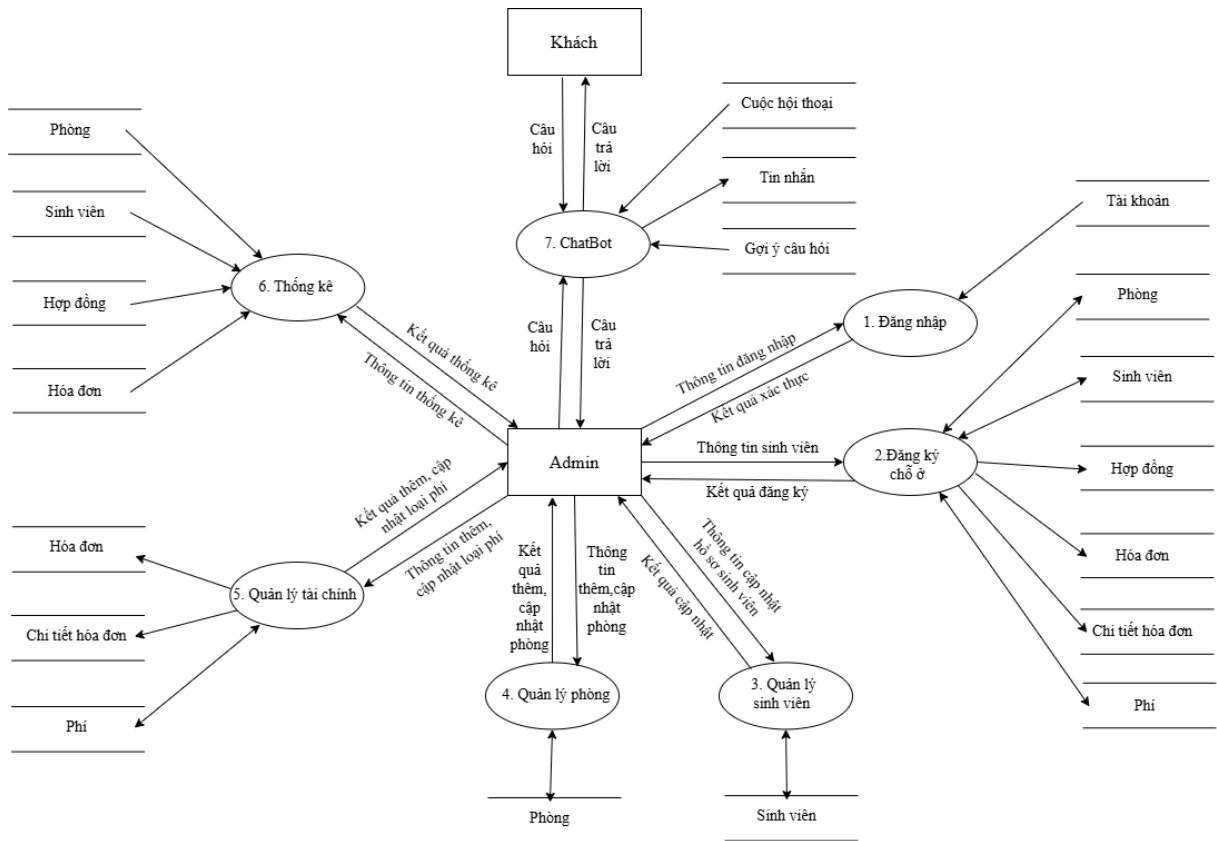
2.3 Mô hình hóa yêu cầu

- Use Case tổng quát:



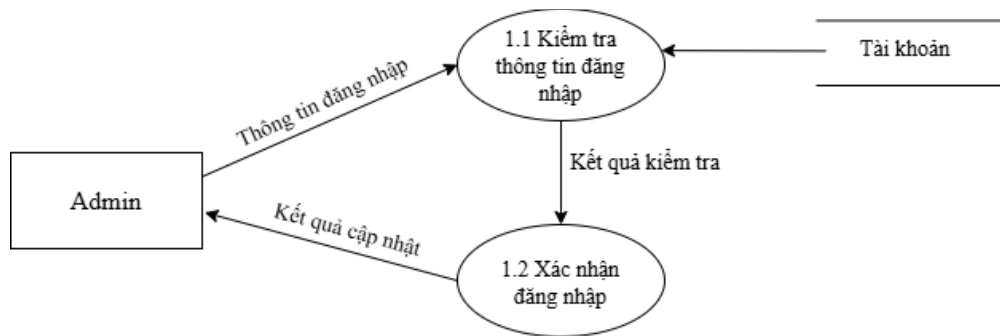
Hình 2.1 Biểu đồ Use Case tổng quát

- DFD mức đỉnh:



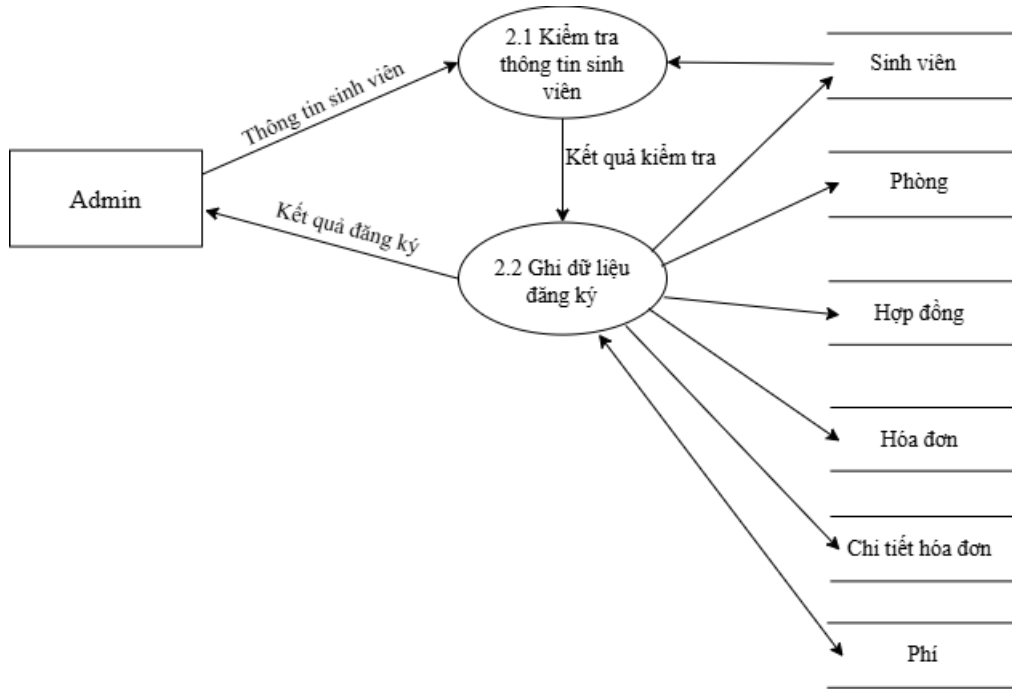
Hình 2.2 Biểu đồ DFD mức đỉnh

- Biểu đồ DFD mức dưới đỉnh (1):



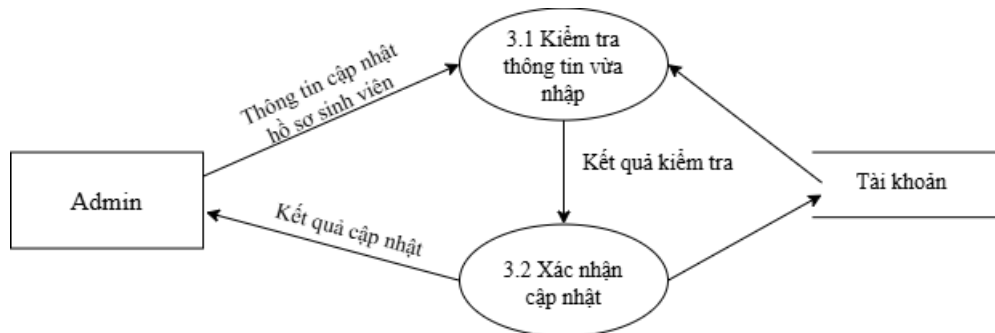
Hình 2.3 Biểu đồ DFD mức dưới đỉnh (1)

- Biểu đồ DFD mức dưới đỉnh (2):



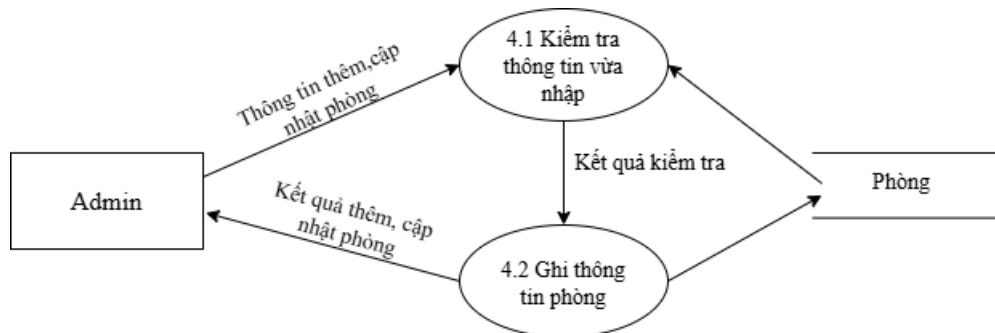
Hình 2.4 Biểu đồ DFD mức dưới đỉnh (2)

- Biểu đồ DFD mức dưới đỉnh (3):



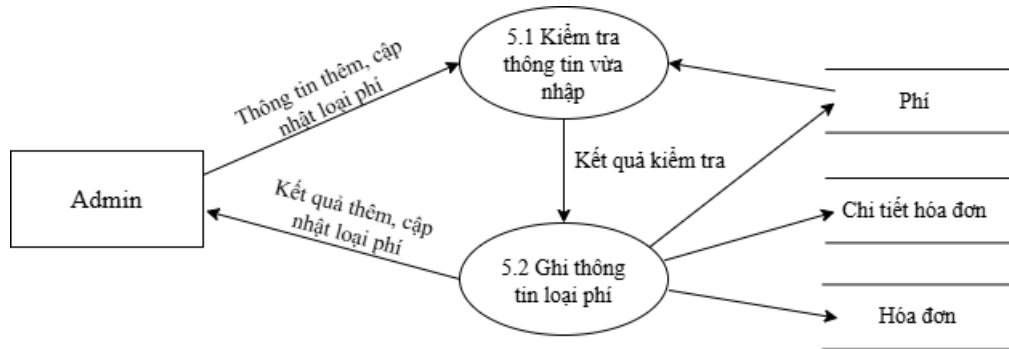
Hình 2.5 Biểu đồ DFD mức dưới đỉnh (3)

- Biểu đồ DFD mức dưới đỉnh (4):



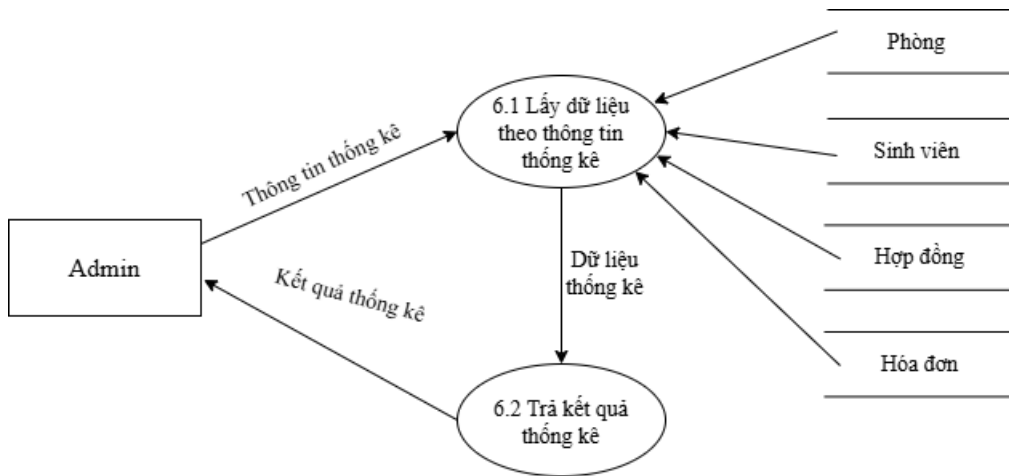
Hình 2.6 Biểu đồ DFD mức dưới đỉnh (4)

- Biểu đồ DFD mức dưới đỉnh (5):



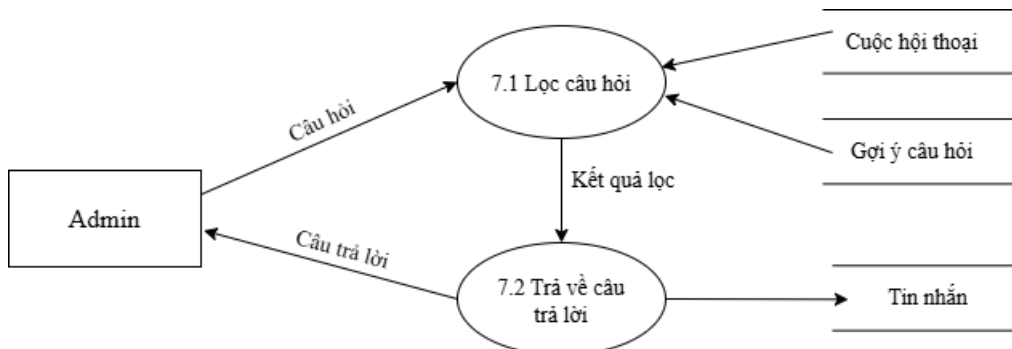
Hình 2.7 Biểu đồ DFD mức dưới đỉnh (5)

- Biểu đồ DFD mức dưới đỉnh (6):



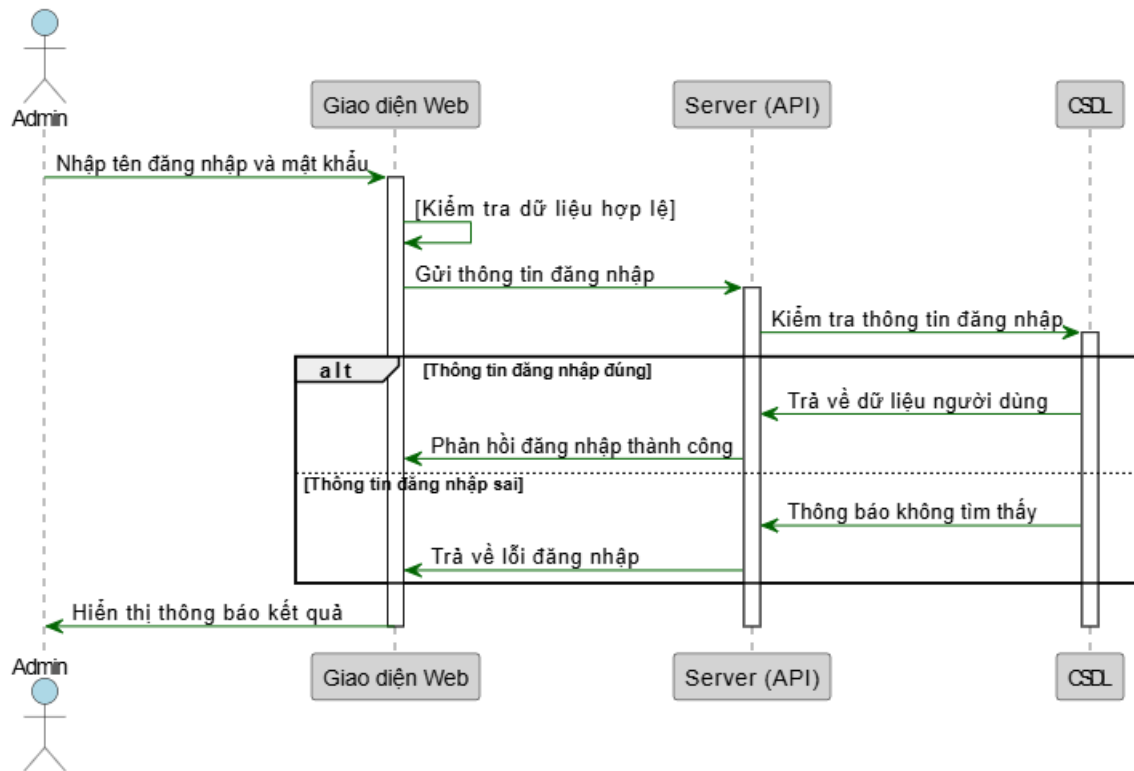
Hình 2.8 Biểu đồ DFD mức dưới đỉnh (6)

- Biểu đồ DFD mức dưới đỉnh (7):



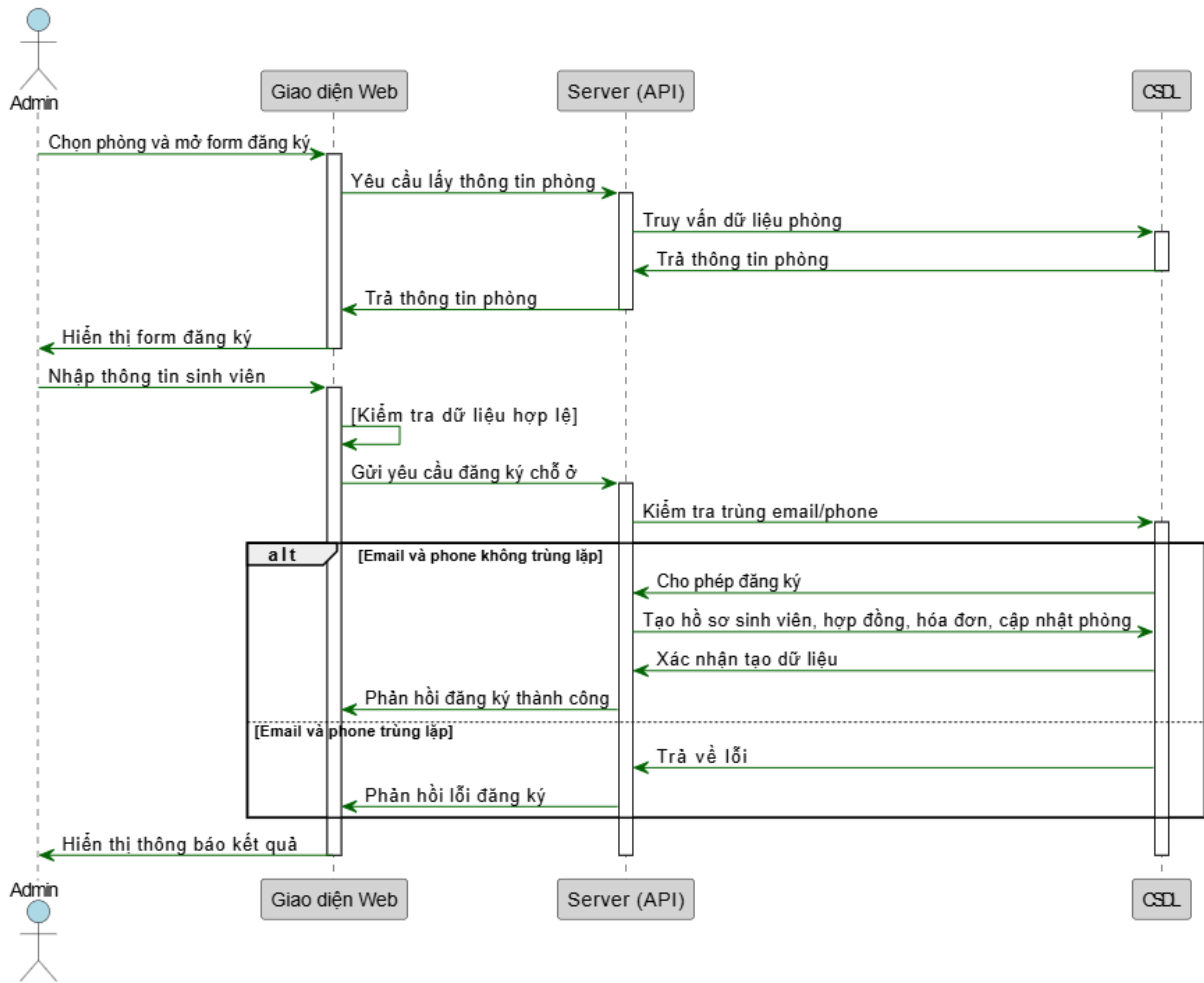
Hình 2.9 Biểu đồ DFD mức dưới đỉnh (7)

2.3.1 Chức năng đăng nhập



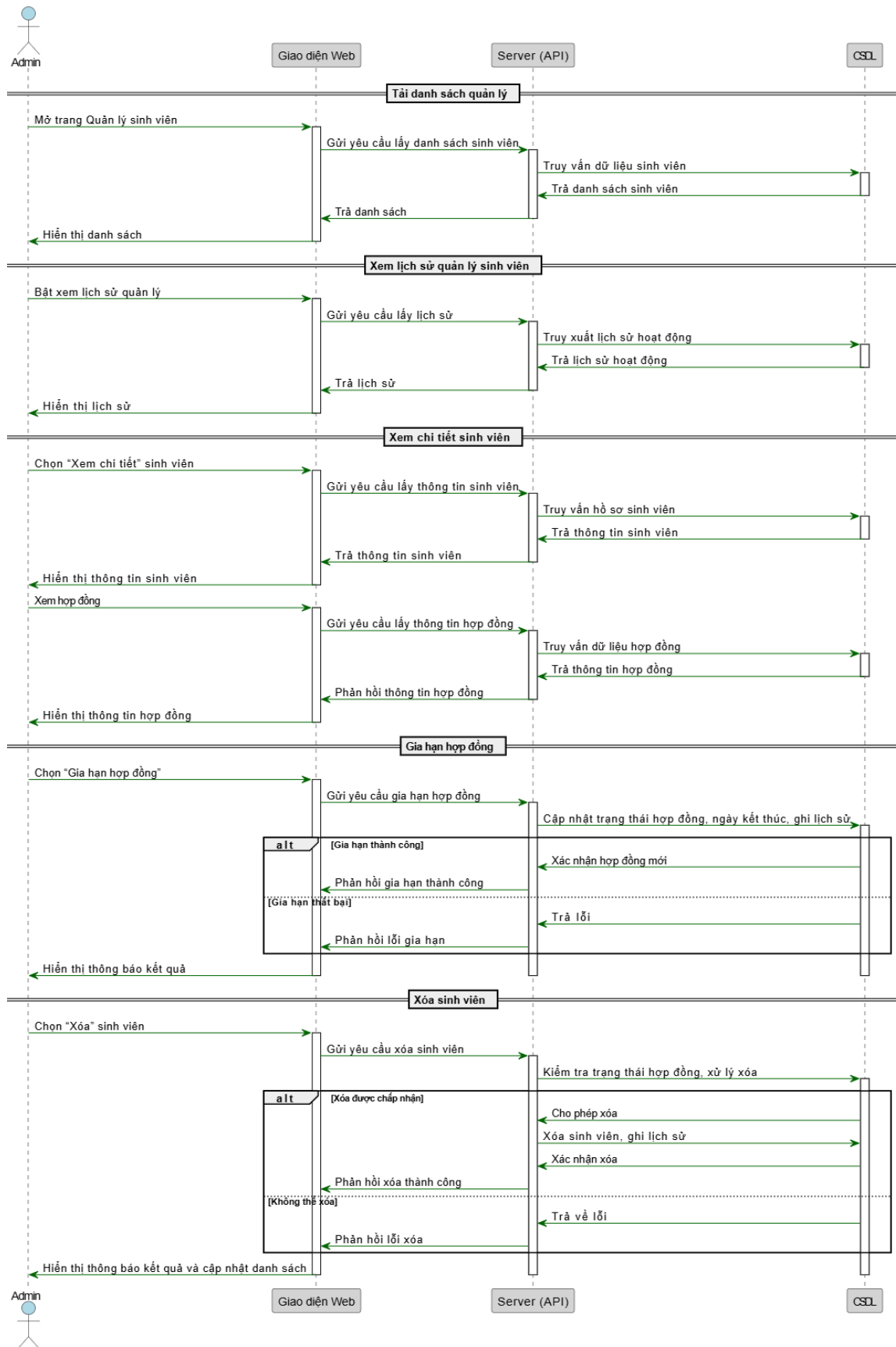
Hình 2.10 Biểu đồ tuần tự Đăng nhập

2.3.2 Chức năng đăng ký chỗ ở



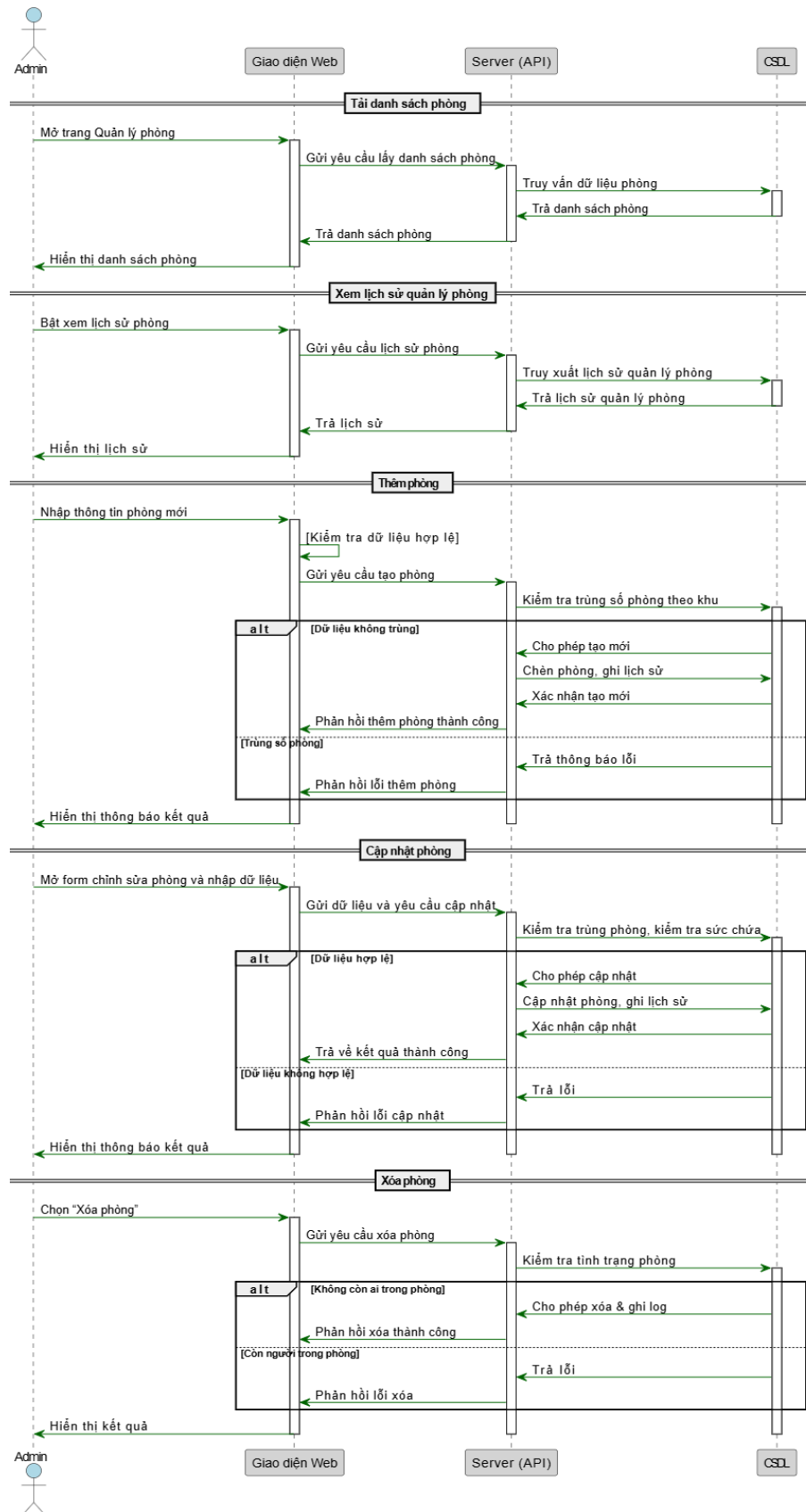
Hình 2.11 Biểu đồ tuần tự Đăng ký chỗ ở

2.3.3 Chức năng quản lý sinh viên



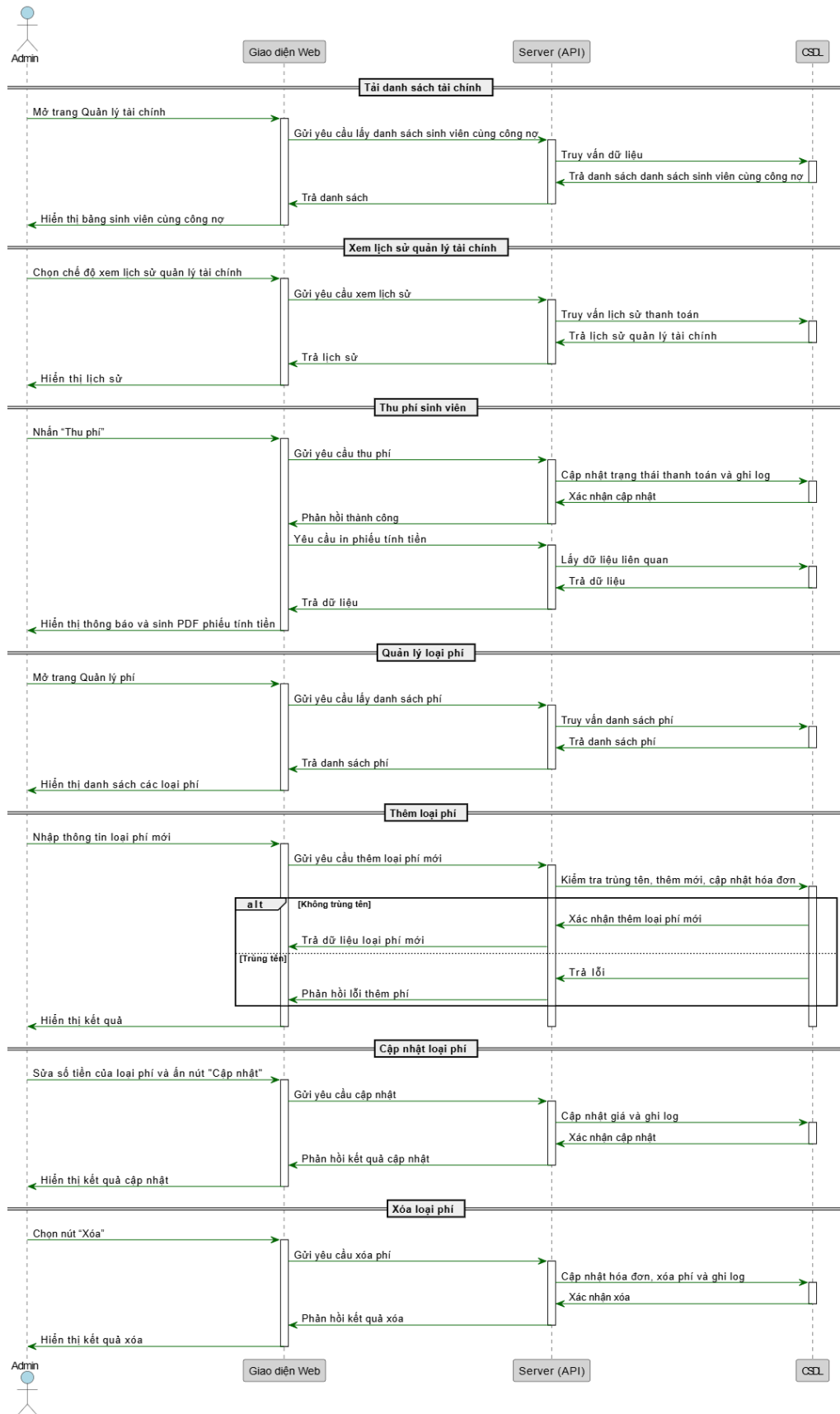
Hình 2.12 Biểu đồ tuần tự Quản lý sinh viên

2.3.4 Chức năng quản lý phòng



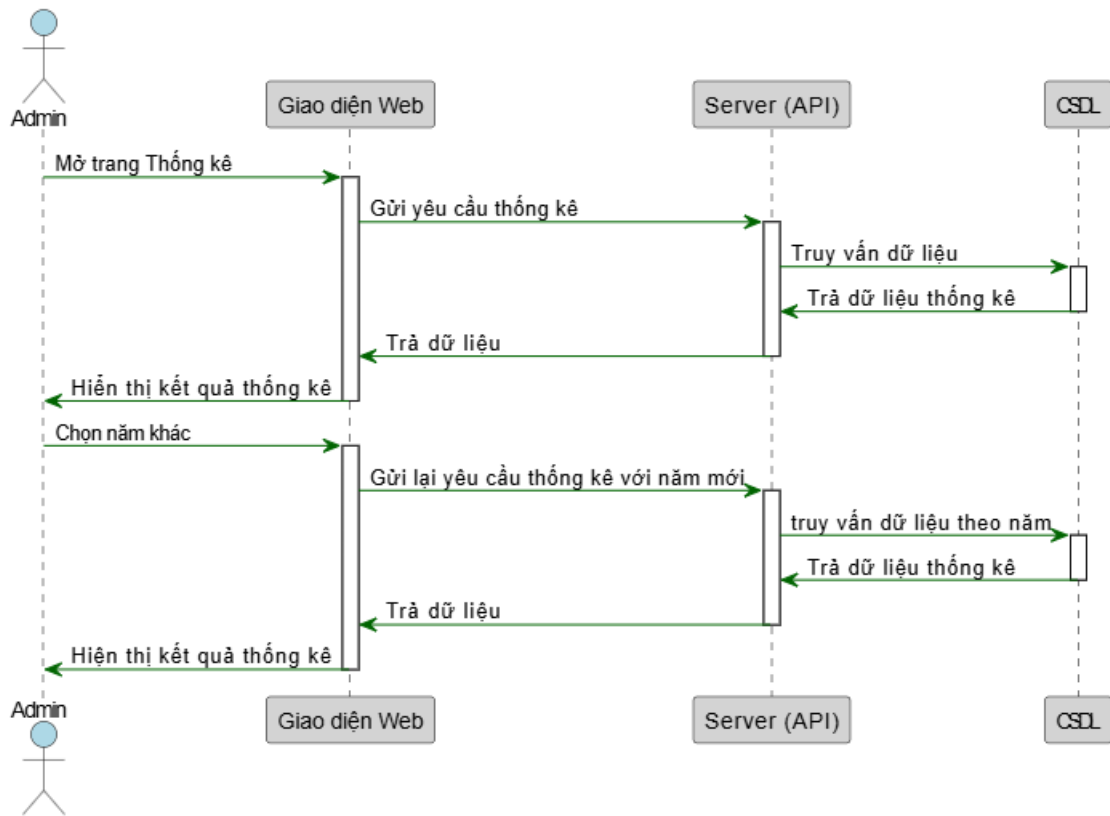
Hình 2.13 Biểu đồ tuần tự Quản lý phòng

2.3.5 Chức năng quản lý tài chính



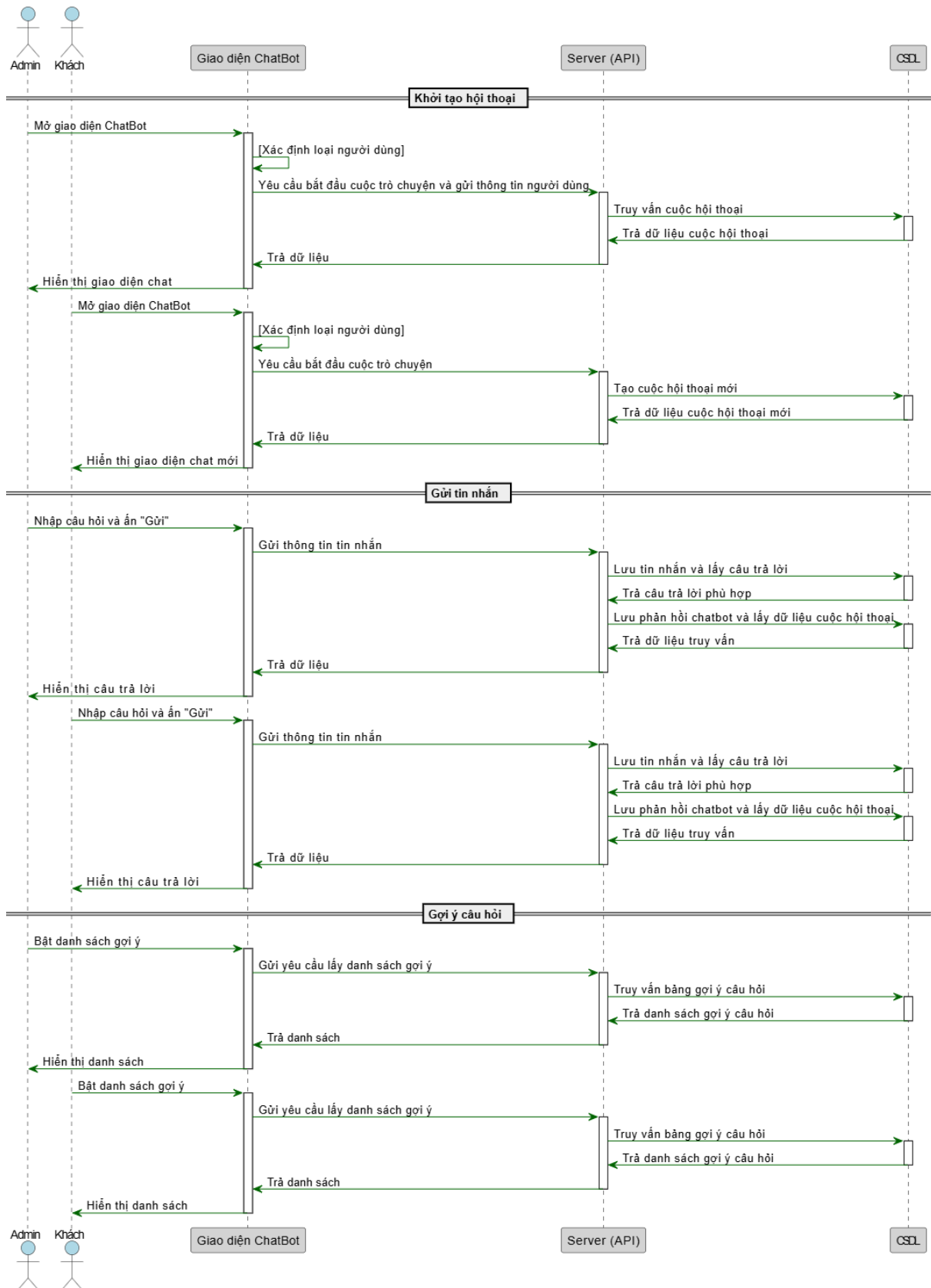
Hình 2.14 Biểu đồ tuần tự Quản lý tài chính

2.3.6 Chức năng thống kê



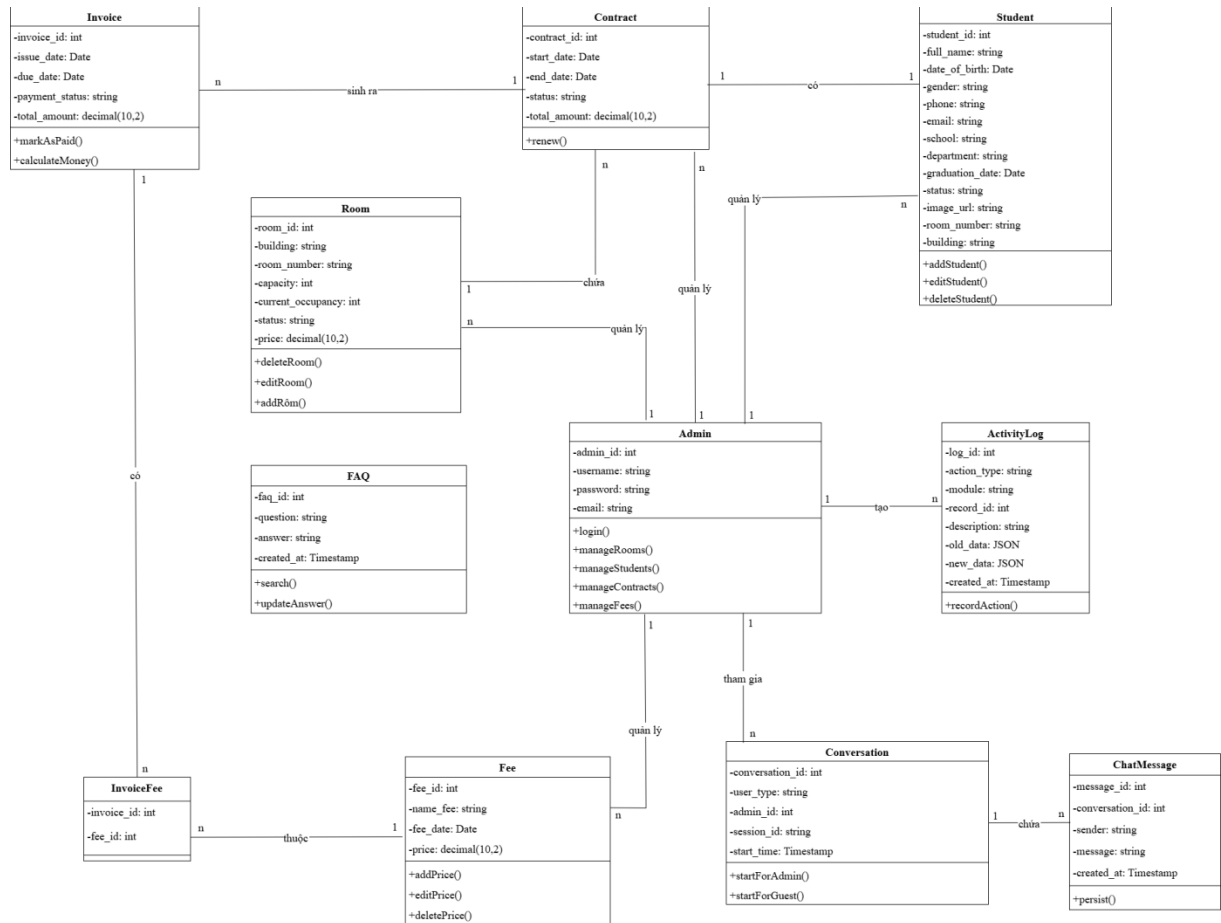
Hình 2.15 Biểu đồ tuần tự Thống kê

2.3.7 Chức năng ChatBot



Hình 2.16 Biểu đồ tuần tự ChatBot

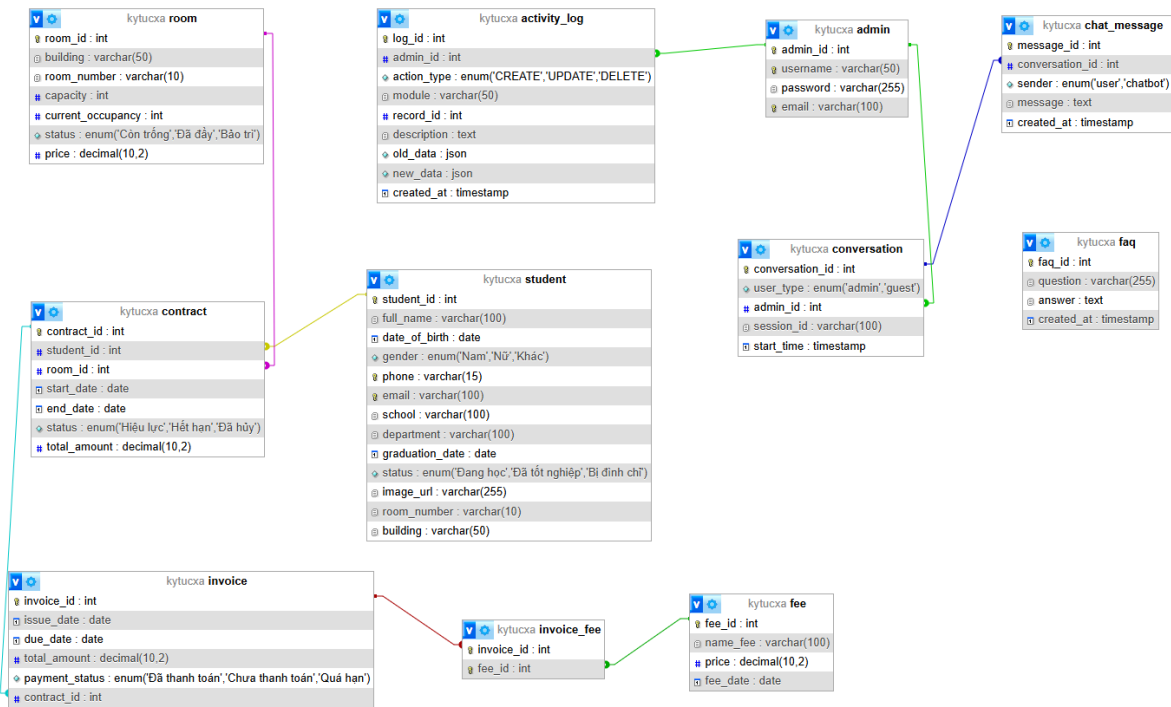
2.4 Sơ đồ Lớp



Hình 2.17 Sơ đồ lớp

2.5 Thiết kế cơ sở dữ liệu

Sơ đồ Quan hệ Cơ sở dữ liệu:



Hình 2.18 Sơ đồ Quan hệ Cơ sở dữ liệu

Sơ đồ này được tạo ra từ công cụ phpMyAdmin Designer, mô tả cấu trúc vật lý chi tiết của cơ sở dữ liệu sau quá trình chuyển đổi từ mô hình thực thể liên kết (ERD) sang mô hình quan hệ.

Sơ đồ thể hiện các thành phần chính bao gồm các bảng (admin, student, room, contract, invoice, fee, invoice_fee, activity_log, conversation, chat_message, faq), các cột (thuộc tính), kiểu dữ liệu cụ thể, và các ràng buộc toàn vẹn dữ liệu. Các đường nối giữa các bảng biểu thị mối quan hệ logic thông qua việc sử dụng Khóa chính (Primary Key - PK) và Khóa ngoại (Foreign Key - FK), đảm bảo dữ liệu được lưu trữ một cách nhất quán và hiệu quả.

2.6 Danh mục các bảng cơ sở dữ liệu

Bảng 2.1 Mô tả các bảng cơ sở dữ liệu

TT	Tên bảng	Mô tả
1	admin	Lưu thông tin tài khoản của người quản lý ký túc xá
2	student	Lưu thông tin cá nhân của sinh viên đang/đã ở ký túc xá
3	room	Lưu thông tin các phòng ký túc xá
4	contract	Lưu thông tin hợp đồng thuê phòng
5	invoice	Lưu hóa đơn thanh toán của sinh viên

6	fee	Lưu danh sách các loại phí chung
7	invoice_fee	Lưu nhiều loại phí vào một hóa đơn
8	activity_log	Lưu lịch sử thao tác của admin
9	conversation	Lưu thông tin cuộc trò chuyện ChatBot
10	chat_message	Lưu tin nhắn trong từng cuộc hội thoại
11	faq	Lưu các câu hỏi – trả lời gợi ý để Chatbot ưu tiên phản hồi

Bảng Admin

Bảng 2.2 Mô tả bảng admin

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
admin_id	int	Không	PK	Mã admin
username	varchar(50)	Không	UNIQUE	Tên đăng nhập
password	varchar (255)	Không		Mật khẩu
email	varchar (100)	Không	UNIQUE	Email của admin

Bảng Student

Bảng 2.3 Mô tả bảng student

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
student_id	Int auto_increment	Không	PK	Mã sinh viên
full_name	varchar (100)	Không		Họ tên
date_of_birth	date	Không		Ngày sinh
gender	enum('Nam','Nữ','Khác')	Không		Giới tính
phone	varchar (15)	Không	UNIQUE	Số điện thoại
email	varchar (100)	Không	UNIQUE	Email
school	varchar (100)	Không		Trường
department	varchar (100)	Không		Khoa
graduation_date	date	Có		Ngày tốt nghiệp
status	enum ('Đang học','Đã tốt nghiệp','Bị đình chỉ')	Không		Tình trạng

image_url	varchar (255)	Có		Ảnh sinh viên
building	varchar (50)	Không		Khu nhà
room_number	varchar (10)	Không		Số phòng

Bảng Room

Bảng 2.4 Mô tả bảng room

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
room_id	Int auto_increment	Không	PK	Mã phòng
building	varchar (50)	Không		Khu nhà
room_number	varchar (10)	Không		Số phòng
capacity	int	Không		Sức chứa
current_occupancy	int	Không		Số người đang ở
status	enum ('Còn trống', 'Đã đầy', 'Bảo trì')	Không		Trạng thái
price	decimal (10,2)	Không		Giá phòng

Bảng Contract

Bảng 2.5 Mô tả bảng contract

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
contract_id	Int auto_increment	Không	PK	Mã hợp đồng
student_id	int	Không	FK	Mã sinh viên
room_id	int	Không	FK	Mã phòng
start_date	date	Không		Ngày bắt đầu
end_date	date	Không		Ngày kết thúc
status	enum ('Hiệu lực', 'Hết hạn', 'Đã hủy')	Không		Trạng thái hợp đồng
total_amount	decimal(10,2)	Không		Tổng tiền

Bảng Invoice

Bảng 2.6 Mô tả bảng invoice

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
invoice_id	Int auto_increment	Không	PK	Mã hóa đơn
issue_date	date	Không		Ngày xuất hóa đơn

due_date	date	Không		Hạn thanh toán
total_amount	decimal (10,2)	Không		Tổng tiền
payment_status	enum ('Đã thanh toán','Chưa thanh toán','Quá hạn')	Không		Trạng thái thanh toán
contract_id	int	Không	FK	Tham chiếu bảng contract

Bảng Fee

Bảng 2.7 Mô tả bảng fee

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
fee_id	Int auto_increment	Không	PK	Mã phí
name_fee	varchar (100)	Không		Tên phí
price	decimal (10,2)	Không		Giá
fee_date	date	Không		Ngày tạo phí

Bảng Invoice_fee

Bảng 2.8 Mô tả bảng room

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
invoice_id	int	Không	PK,FK	Mã hóa đơn
fee_id	int	Không	PK,FK	Mã phí

Bảng Activity_log

Bảng 2.9 Mô tả bảng activity_log

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
log_id	Int auto_increment	Không	PK	Mã log
admin_id	int	Không	FK	Mã admin
action_type	enum ('CREATE','UPDATE','DELETE')	Không		Loại hành động
module	varchar (50)	Không		Tên module
record_id	int	Không		ID bản ghi tác động

description	text	Có		Mô tả
old_data	json	Có		Dữ liệu cũ
new_data	json	Có		Dữ liệu mới
created_at	timestamp	Không		Thời gian tạo

Bảng Conversation

Bảng 2.10 Mô tả bảng conversation

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
conversation_id	Int auto_increment	Không	PK	Mã hội thoại
user_type	enum ('admin', 'guest')	Không		Loại người dùng
admin_id	int	Có	FK	Mã admin
session_id	varchar (100)	Có		ID phiên hội thoại
start_time	timestamp	Không		Thời gian bắt đầu

Bảng Chat_message

Bảng 2.11 Mô tả bảng chat_message

Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
message_id	Int auto_increment	Không	PK	Mã tin nhắn
conversation_id	int	Không	FK	Tham chiếu hội thoại
sender	enum ('user', 'chatbot')	Không		Người gửi
message	text	Không		Tin nhắn
created_at	timestamp	Không		Thời gian tạo

Bảng FAQ

Bảng 2.12 Mô tả bảng faq

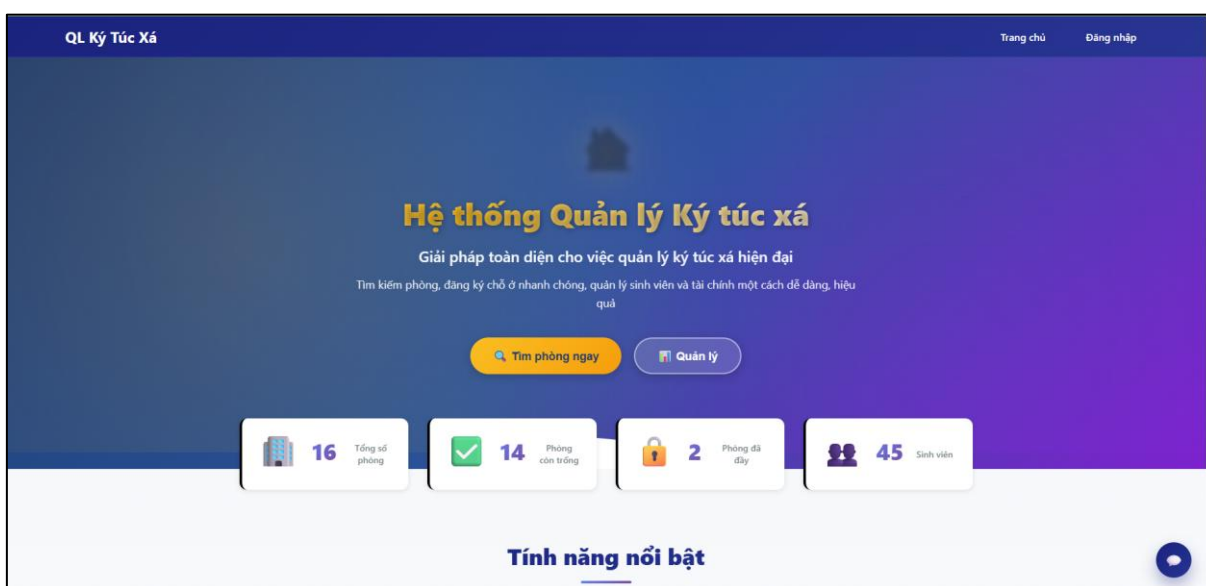
Tên trường	Kiểu dữ liệu	Null	Key	Ghi chú
------------	--------------	------	-----	---------

faq_id	Int auto_increment	Không	PK	Mã FAQ
question	varchar (255)	Không		Câu hỏi
answer	text	Không		Câu trả lời
created_at	timestamp	Không		Ngày tạo

2.7 Thiết kế giao diện

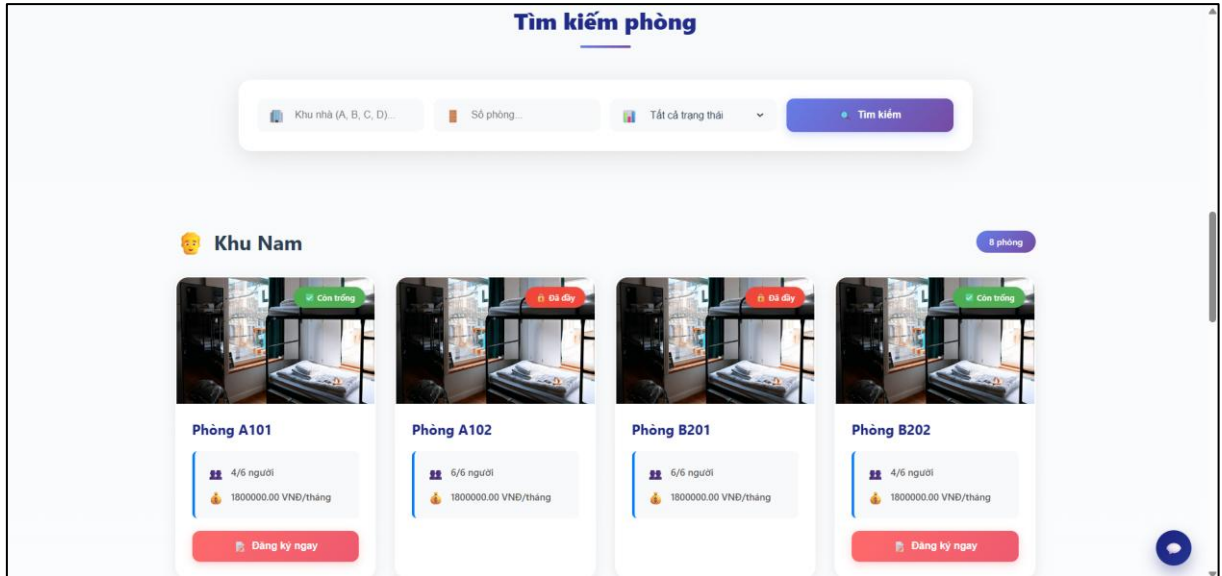
2.7.1 Giao diện người dùng

Trang chủ của hệ thống mở đầu bằng banner hero nổi bật giới thiệu giải pháp quản lý ký túc xá, kèm nút hành động rõ ràng. Ngay bên dưới là cụm thống kê trực quan về số phòng, tình trạng chỗ ở và tổng sinh viên, tiếp nối bởi lưới “Tính năng nổi bật” tóm tắt các tiện ích chính. Toàn bộ bố cục được bọc trong Navbar/Footer thống nhất và tối ưu cho cả máy tính lẫn thiết bị di động..



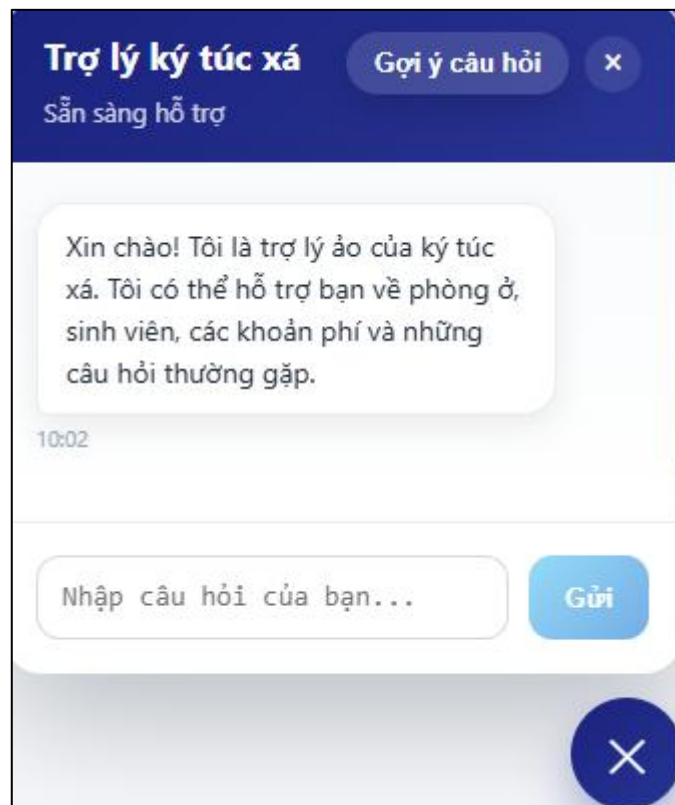
Hình 2.19 Màn hình Trang chủ

Màn hình tra cứu phòng nằm trong Trang chủ, người dùng có thể tìm phòng nhanh qua bộ lọc theo khu, số phòng, trạng thái, rồi xem danh sách phòng, phòng được phân tách khu nam nữ riêng với hình ảnh, sức chứa, giá thuê và nút Đăng ký ngay.



Hình 2.20 Màn hình tra cứu phòng

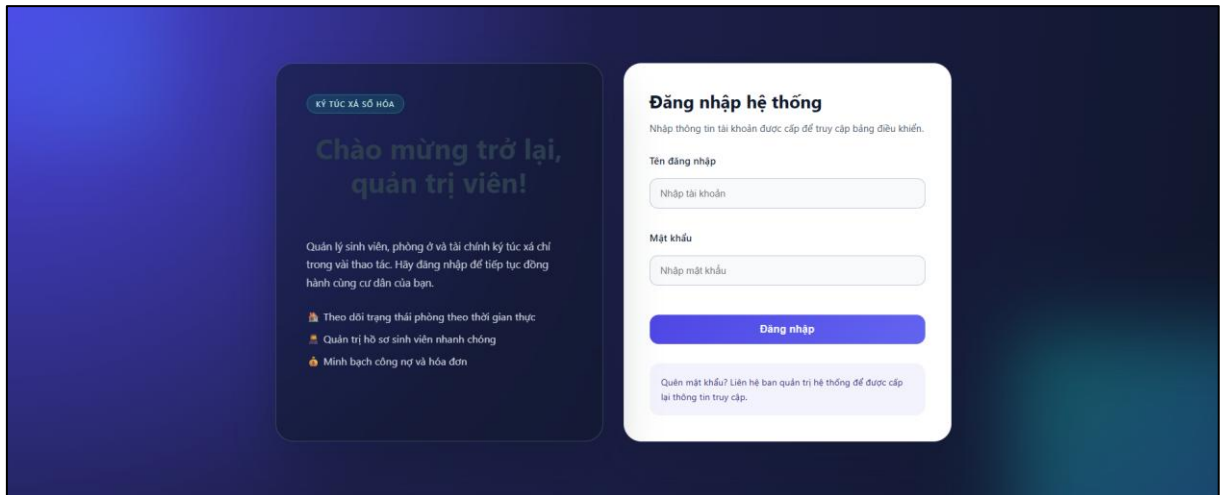
Giao diện ChatBot xuất hiện dưới dạng nút bong bóng ở góc màn hình; khi mở ra sẽ hiển thị khung hội thoại bo tròn hiện đại với tiêu đề “Trợ lý ký túc xá”, trạng thái hoạt động và nút xem gợi ý câu hỏi. Khu vực tin nhắn tách màu rõ ràng giữa người dùng và bot, có thời gian gửi, kèm khả năng xem nhanh các gợi ý FAQ để gửi một chạm. Ô nhập tin được thiết kế dạng textarea gọn gàng cùng nút “Gửi” nổi bật.



Hình 2.21 Màn hình ChatBot

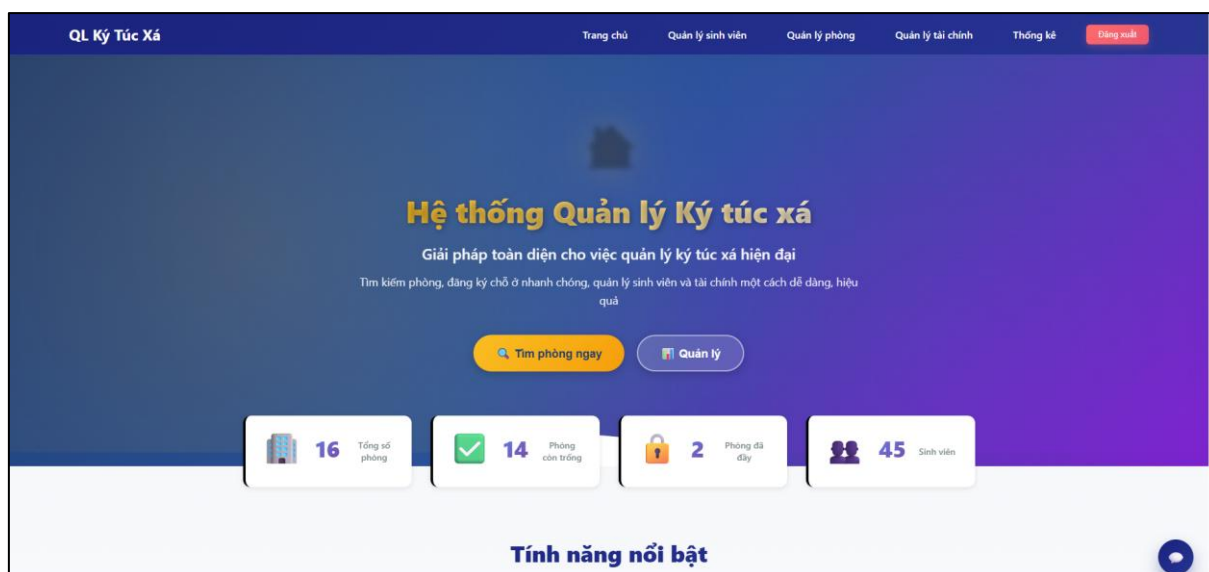
2.7.2 Giao diện Admin

Trang Đăng nhập được thiết kế với nền gradient hiện đại và các khối glassmorphism tạo chiều sâu, chia thành hai cột: bên trái giới thiệu ngắn gọn lợi ích quản trị ký túc xá, bên phải là thẻ đăng nhập màu trắng với tiêu đề rõ ràng, ô nhập tài khoản/mật khẩu kèm gợi ý placeholder. Nút “Đăng nhập” nổi bật bằng hiệu ứng gradient đa sắc, bên dưới là thông tin hỗ trợ khi quên mật khẩu.



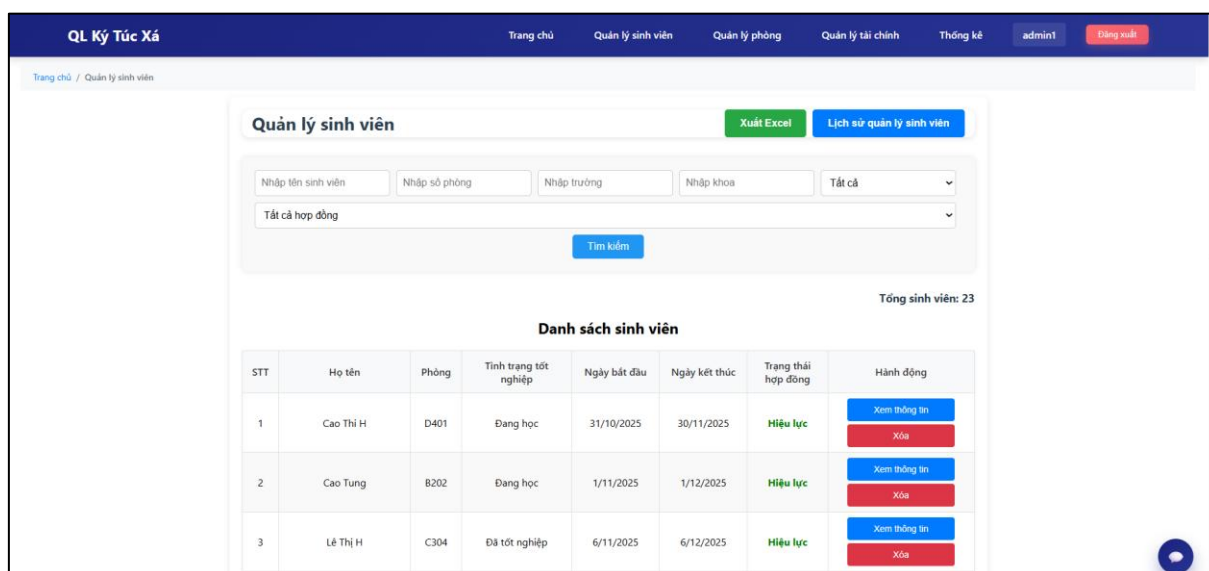
Hình 2.22 Màn hình Login

Khi Admin đăng nhập thành công, Trang chủ sẽ hiển thị thêm các chức năng quản lý ở trên thanh Navbar và Admin được thực hiện các chức năng đó.



Hình 2.23 Màn hình Chính của Admin

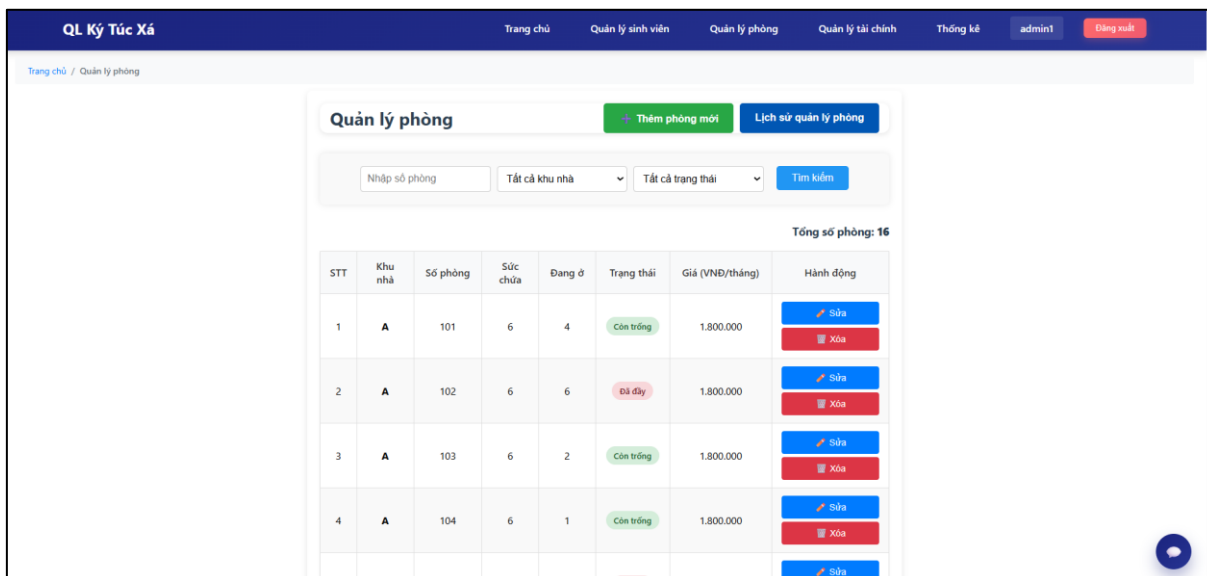
Trang Quản lý sinh viên cung cấp bảng điều khiển rõ ràng với bộ lọc đa tiêu chí (tên, phòng, trường, khoa, trạng thái học tập và trạng thái hợp đồng) để tìm kiếm nhanh. Danh sách hiển thị đầy đủ thông tin thời gian, phòng ở, trạng thái hợp đồng cùng các nút xem chi tiết, gia hạn hoặc xóa. Admin có thể chuyển sang tab lịch sử thao tác để theo dõi các cập nhật trước đây và xuất toàn bộ dữ liệu ra Excel chỉ với một cú nhấp chuột.



Hình 2.24 Màn hình Quản lý sinh viên

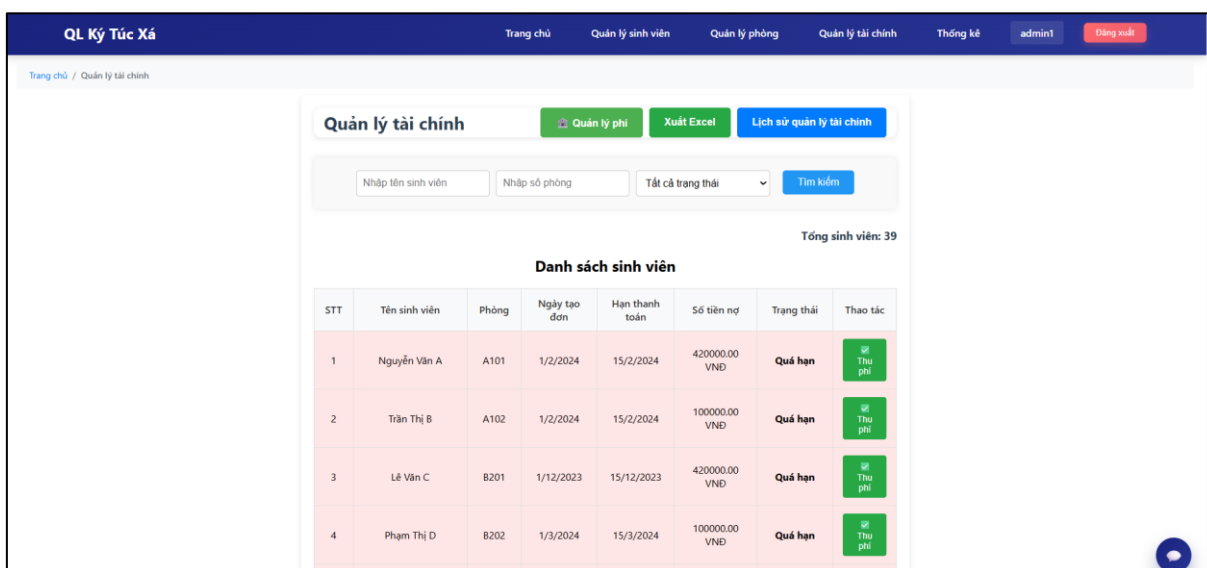
Trang Quản lý phòng cung cấp bảng dữ liệu trực quan với bộ lọc theo khu, trạng thái và số phòng, giúp admin nhanh chóng tìm kiếm và theo dõi sức chứa từng phòng. Các nút thao tác cho phép thêm mới, chỉnh sửa hoặc xóa phòng ngay trong bảng, kèm

hiển thị lịch sử thay đổi để kiểm soát hoạt động. Toàn bộ giao diện được bố trí gọn gàng, hỗ trợ phân trang và hiển thị thông báo khi thao tác thành công hay gặp lỗi.



Hình 2.25 Màn hình Quản lý phòng

Trang Quản lý tài chính tập trung mọi thông tin thu phí theo giao diện bảng rõ ràng: admin có thể lọc sinh viên theo tên, phòng, trạng thái, xem hạn thanh toán, số tiền nợ và thực hiện thu phí chỉ bằng một nút bấm. Khi cần, hệ thống cho phép mở phiếu tính tiền dạng PDF, xuất danh sách ra Excel và chuyển sang tab lịch sử để theo dõi các thao tác thu chi đã thực hiện. Ở trang này, Admin có thể chuyển sang trang Quản lý phí bằng cách nhấn nút Quản lý phí.



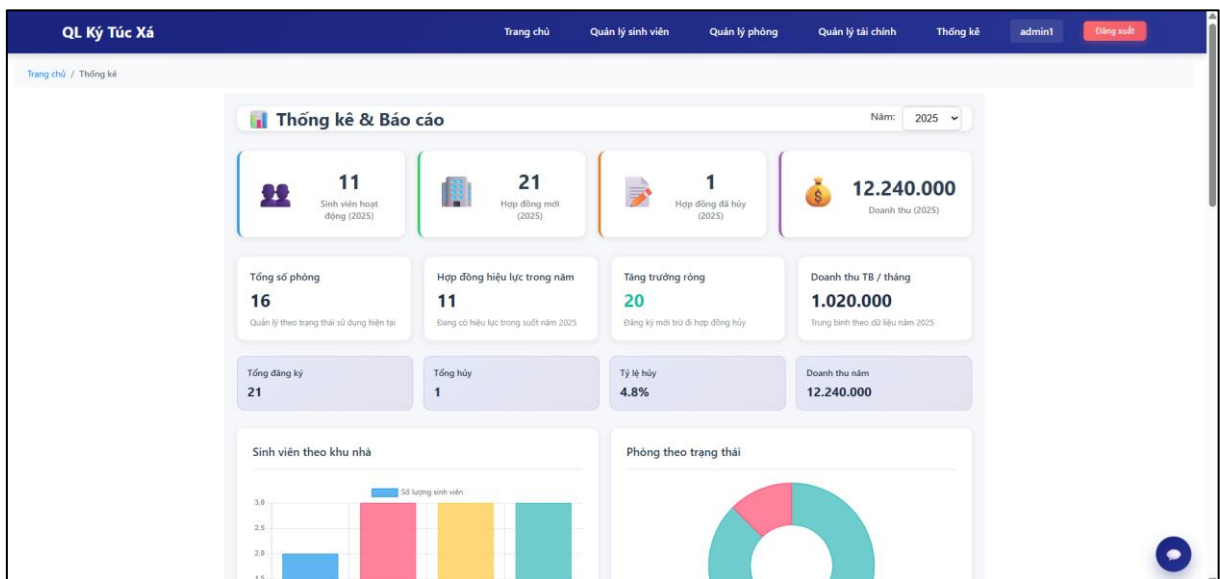
Hình 2.26 Màn hình Quản lý tài chính

Trang Quản lý phí cho phép cấu hình danh sách các khoản thu của ký túc xá trong một bảng trực quan: bạn có thể thêm mới, chỉnh sửa hoặc xóa phí. Mỗi thao tác được ghi nhận vào lịch sử, giúp theo dõi thay đổi và đảm bảo tính minh bạch khi các khoản phí được áp dụng vào hóa đơn sinh viên.

STT	Loại phí	Số tiền	Ngày tạo	Thao tác
-	Loại phí	Số tiền	-	<button>Thêm</button>
1	Tiền wifi	80000	06/12/2025	<button>Cập nhật</button> <button>Xóa</button>
2	Tiền ăn	100000	04/05/2025	<button>Cập nhật</button> <button>Xóa</button>
3	Tiền camera	30000	04/05/2025	<button>Cập nhật</button> <button>Xóa</button>
4	Tiền vòi nước	20000	02/05/2025	<button>Cập nhật</button> <button>Xóa</button>
5	Tiền sửa cửa	50000	02/05/2025	<button>Cập nhật</button> <button>Xóa</button>
6	Tiền cọc	100000	02/05/2025	<button>Cập nhật</button> <button>Xóa</button>

Hình 2.27 Màn hình Quản lý phí

Trang Quản lý thống kê hiển thị các biểu đồ và số liệu trực quan theo từng năm về tình trạng phòng, số lượng sinh viên và doanh thu, giúp ban quản lý nắm bắt xu hướng ký túc xá trong nháy mắt. Admin có thể chọn năm cần xem, hệ thống sẽ tự động cập nhật đồ thị và bảng số liệu.



Hình 2.28 Màn hình Thống kê

Chương 3: PHÁT TRIỂN WEBSITE

3.1 Lập kế hoạch phát triển, phân chia modul, ước lượng thời gian

Bảng 3.1 Bảng kế hoạch thời gian phát triển

STT	Module	Công việc	Thời gian ước lượng
1	Cơ sở dữ liệu	Thiết kế, xây dựng CSDL	1 tuần
2	Giao diện người dùng	Thiết kế UI/UX tổng thể, tối ưu trải nghiệm người dùng	3 ngày
3	Đăng nhập	Xây dựng giao diện và chức năng Đăng nhập cho Admin	3 ngày
4	Tìm kiếm phòng	Hiển thị danh sách phòng, tìm kiếm và lọc thông tin phòng	3 ngày
5	Quản lý phòng	Xây dựng chức năng quản lý phòng, tìm kiếm, lọc danh sách phòng, lịch sử quản lý phòng	3 ngày
6	Đăng ký chỗ ở	Xây dựng form đăng ký, tạo hồ sơ sinh viên, tạo hợp đồng, tạo hóa đơn	1 tuần
7	Quản lý sinh viên	Xây dựng danh sách, lọc danh sách sinh viên, xem thông tin sinh viên, cập nhật, xóa sinh viên, gia hạn hợp đồng, lịch sử quản lý sinh viên	1 tuần
8	Quản lý tài chính	Xây dựng chức năng Quản lý phí, hiển thị danh sách thanh toán, xác nhận thu tiền, tạo PDF phiếu thu, lịch sử giao dịch	1 tuần
9	Thống kê	Xây dựng Dashboard tổng hợp và các biểu đồ: phòng, sinh viên, hợp đồng, doanh thu theo tháng	3 ngày
10	ChatBot	Tích hợp ChatBot, xử lý câu hỏi	3 ngày

		FAQ, lưu lịch sử trò chuyện admin	
11	Kiểm thử	Kiểm tra tính năng, sửa lỗi, tối ưu hiệu suất hệ thống	1 tuần

→ Tổng thời gian dự kiến: 53 ngày

3.2 Phát triển frontend

3.2.1 Cấu trúc thư mục Frontend

demo/

```

| — node_modules/      # Chứa các thư viện và dependencies được cài đặt
                           qua npm
| — package.json        # File cấu hình project và danh sách dependencies
| — package-lock.json   # File khóa phiên bản của các dependencies
| — tailwind.config.js  # Cấu hình cho Tailwind CSS
| — public/            # Chứa các file static được phục vụ trực tiếp
|   | — index.html      # File HTML chính của ứng dụng
|   | — robots.txt      # File cấu hình cho công cụ tìm kiếm
|   | — images/         # Thư mục chứa các hình ảnh tĩnh
| — src/               # Thư mục nguồn chính của project
|   | — App.js          # Component chính của ứng dụng
|   | — index.js        # File khởi động ứng dụng React
|   | — index.css       # File CSS chung cho toàn bộ ứng dụng
|   | — reportWebVitals.js # File hỗ trợ đo lường hiệu suất ứng dụng
|   | — setupTests.js   # Cấu hình và thiết lập cho testing
|   | — components/     # Chứa các component giao diện
|   |   | — Breadcrumb.js # Component hiển thị đường dẫn điều hướng
|   |   | — ChatBotWidget.js # Component chatbot hỗ trợ người dùng
|   |   | — ContractDetailPage.js # Trang quản lý chi tiết hợp đồng
|   |   | — EditStudentPage.js # Trang chỉnh sửa thông tin sinh viên
|   |   | — FeeManagementPage.js # Trang quản lý các loại phí

```

			FinanceManagementPage.js	# Trang quản lý tài chính
			Footer.js	# Footer của trang web
			HomePage.js	# Trang chủ của ứng dụng
			LoginPage.js	# Trang đăng nhập
			Navbar.js	# Thanh menu điều hướng
			PrivateRoute.js	# Component quản lý quyền truy cập
			RegistrationPage.js	# Trang đăng ký phòng ở
			RoomManagementPage.js	# Trang quản lý phòng ở
			StatisticsPage.js	# Trang thống kê dữ liệu
			StudentDetailPage.js	# Trang hiển thị chi tiết sinh viên
			StudentManagementPage.js	# Trang quản lý danh sách sinh viên
			css/	# Chứa các file CSS riêng biệt cho từng component
			utils/	# Chứa các file tiện ích
			pdfHelper.js	# Các hàm hỗ trợ tạo file PDF

3.2.2 Kết nối API

- Dự án Quản lý ký túc xá kết nối với backend thông qua các API PHP đặt tại đường dẫn /QL_KTX/api/. Việc gọi API được thực hiện chủ yếu bằng thư viện axios trong các component React.

- Kết nối API trong dự án được thực hiện thông qua các endpoint sau:

+ API Quản lý sinh viên:

GET	/api/getStudents.php	#Lấy danh sách sinh viên.
GET	/api/getStudentDetail.php?studentId=...	#Lấy chi tiết sinh viên
GET	/api/getStudentHistory.php?studentId=...	#Lấy lịch sử sinh viên.
POST	/api/updateStudent.php	#Cập nhật thông tin sinh viên.
POST	/api/deleteStudent.php	#Xóa sinh viên.
POST	/api/registerStudent.php	#Đăng ký sinh viên mới.
GET	/api/getContractDetail.php?studentId=...	#Lấy chi tiết hợp đồng
POST	/api/renewContract.php	#Gia hạn hợp đồng.

+ API Quản lý phòng:

GET /api/getRooms.php #Lấy danh sách phòng.

GET /api/getRoomById.php?roomId=... #Lấy thông tin phòng theo ID.

GET /api/searchRooms.php?building=...&status=... #Tìm kiếm phòng

+ API Quản lý tài chính:

GET /api/getAllStudentsFinance.php #Lấy thông tin tài chính sinh viên.

GET /api/getPaymentHistory.php?studentId=... #Lấy lịch sử quản lý tài chính

GET /api/getFees.php #Lấy danh sách các loại phí.

POST /api/confirmPayment.php #Xác nhận thanh toán phí.

POST /api/addFee.php #Thêm loại phí mới.

POST /api/updateFee.php #Cập nhật thông tin phí.

POST /api/deleteFee.php #Xóa loại phí.

+ API Quản lý hệ thống:

POST /api/login.php #Đăng nhập

GET /api/updateDatabase.php #Cập nhật cơ sở dữ liệu

- Cách kết nối API:

1. Sử dụng axios để thực hiện các request:

axios.get("http://localhost:8080/QL_KTX/api/getStudents.php")

axios.post("http://localhost:8080/QL_KTX/api/confirmPayment.php")

2. Xử lý response:

Sử dụng .then() để xử lý dữ liệu trả về

Sử dụng .catch() để xử lý lỗi

Kiểm tra response.data.success để xác nhận kết quả

3. Trả về dữ liệu:

Dữ liệu được trả về dưới dạng JSON

Bao gồm các trường:

success: Trạng thái thành công

data: Dữ liệu trả về

message: Thông báo

4. Xử lý lỗi:

Hiển thị thông báo lỗi và thực hiện các hành động phù hợp (ví dụ: reload dữ liệu, chuyển hướng)

5. Bảo mật:

Khi đăng nhập thành công, thông tin người dùng được lưu vào localStorage để xác thực các request tiếp theo.

Các route quan trọng sử dụng component PrivateRoute để kiểm tra quyền truy cập (dựa vào trạng thái đăng nhập).

- Ví dụ thực tế trong dự án:

+ Khi khởi động ứng dụng, App.js gọi API cập nhật database:

```
useEffect(() => {  
  axios.get("http://localhost:8080/QL_KTX/api/updateDatabase.php")  
    .then(response => {  
      if (response.data.success) {  
        console.log("Updated database:", response.data.updates);  
      }  
    })  
    .catch(error => {  
      console.error("Error updating database:", error);  
    });  
}, []);
```

+ Khi xác nhận thanh toán:

```
axios.post("http://localhost:8080/QL_KTX/api/confirmPayment.php", {  
  studentId, amount })  
  .then(response => {  
    if (response.data.success) {
```

```
// Hiển thị thông báo thành công
} else {
    // Hiển thị thông báo lỗi
}
})
.catch(error => {
    // Xử lý lỗi mạng hoặc server
});
```

3.3 Phát triển backend

3.3.1 Cấu trúc thư mục Backend

```
api/
| — .htaccess          # Cấu hình rewrite URL và CORS cho API
| — dbconnect.php      # Kết nối cơ sở dữ liệu MySQL sử dụng PDO
| — index.php          # Router đơn giản cho các request API
| — login.php          # Xử lý đăng nhập admin
| — registerStudent.php # Đăng ký sinh viên mới
| — updateStudent.php  # Cập nhật thông tin sinh viên
| — deleteStudent.php  # Xóa sinh viên
| — getStudents.php    # Lấy danh sách sinh viên (có phân trang, lọc)
| — getStudentDetail.php # Lấy chi tiết sinh viên
| — getStudentHistory.php # Lấy lịch sử hoạt động của sinh viên
| — renewContract.php  # Gia hạn hợp đồng cho sinh viên
| — getContractDetail.php # Lấy chi tiết hợp đồng của sinh viên
| — getStatistics.php  # Lấy thống kê tổng quan (phòng, sinh viên, hợp
đồng, doanh thu)
| — getRegistrationsByMonth.php # Thống kê số lượng đăng ký phòng theo
tháng
```

- |— getCancellationsByMonth.php # Thống kê số lượng hủy hợp đồng theo tháng
- |— getRevenueByMonth.php # Thống kê doanh thu theo tháng
- |— getAllStudentsFinance.php # Lấy thông tin tài chính của sinh viên
- |— getPaymentHistory.php # Lấy lịch sử thanh toán/phí
- |— getPaymentReceipt.php # Lấy thông tin phiếu thu chi tiết
- |— confirmPayment.php # Xác nhận thanh toán hóa đơn
- |— getFees.php # Lấy danh sách các loại phí
- |— addFee.php # Thêm loại phí mới
- |— updateFee.php # Cập nhật thông tin phí
- |— deleteFee.php # Xóa loại phí
- |— getRooms.php # Lấy danh sách phòng (lọc, tìm kiếm)
- |— getRoomById.php # Lấy thông tin phòng theo ID
- |— getRoomDetail.php # Lấy chi tiết phòng và danh sách sinh viên trong phòng
- |— getRoomsManagement.php # Lấy danh sách phòng (phân trang, lọc cho quản lý)
- |— addRoom.php # Thêm phòng mới
- |— updateRoom.php # Cập nhật thông tin phòng
- |— deleteRoom.php # Xóa phòng
- |— getRoomHistory.php # Lấy lịch sử hoạt động của phòng
- |— updateDatabase.php # Cập nhật trạng thái hệ thống (hợp đồng, hóa đơn, sinh viên)
- |— searchRooms.php # Tìm kiếm phòng theo tiêu chí
- |— getFaqSuggestions.php # Lấy danh sách câu hỏi thường gặp cho chatbot
- |— startConversation.php # Khởi tạo hội thoại mới với chatbot
- |— sendMessage.php # Gửi tin nhắn và nhận phản hồi từ chatbot

3.3.2 Xử lý dữ liệu

- Kết nối cơ sở dữ liệu:

Toàn bộ các file PHP sử dụng chung file dbconnect.php để kết nối tới MySQL thông qua PDO, đảm bảo an toàn và dễ quản lý kết nối.

- Nhận và kiểm tra dữ liệu:

Dữ liệu được nhận qua \$_GET, \$_POST, hoặc đọc trực tiếp từ php://input (dạng JSON).

- Xử lý nghiệp vụ:

Thực hiện các thao tác CRUD (Create, Read, Update, Delete) với các bảng như: sinh viên, phòng, hợp đồng, phí, hóa đơn...

Một số thao tác phức tạp sử dụng transaction (beginTransaction, commit, rollBack) để đảm bảo tính toàn vẹn dữ liệu.

- Trả về kết quả:

Kết quả trả về luôn ở dạng JSON, gồm các trường: success, data (nếu có), message (nếu có).

Khi có lỗi, trả về thông báo lỗi rõ ràng cho frontend.

3.3.3 API

- Mỗi file PHP tương ứng với một endpoint API, thực hiện một chức năng cụ thể.

- Cấu trúc endpoint:

GET: Lấy dữ liệu

POST: Tạo mới

PUT/PATCH: Cập nhật

DELETE: Xóa

- Ví dụ endpoint:

GET /api/getStudents.php?page=1&limit=10

POST /api/updateStudent.php (dữ liệu gửi dạng JSON hoặc form-data)

POST /api/login.php (gửi username, password)

3.3.4 Bảo mật

- Kết nối an toàn: Sử dụng PDO với prepared statement để chống SQL Injection.
- CORS: File .htaccess và header trong các file PHP cho phép CORS, giúp frontend và backend giao tiếp an toàn.
- Xử lý lỗi: Toàn bộ thao tác nhạy cảm đều được bọc trong try-catch, rollback khi có lỗi, trả về thông báo lỗi rõ ràng.

KẾT LUẬN

1. Kết quả đạt được

- Sau quá trình nghiên cứu, phân tích và triển khai, đề tài “Xây dựng website Quản lý ký túc xá” đã hoàn thành hầu hết các mục tiêu đề ra ban đầu. Hệ thống đã được xây dựng với kiến trúc rõ ràng, giao diện dễ sử dụng và đáp ứng được các yêu cầu nghiệp vụ quan trọng của công tác quản lý ký túc xá. Cụ thể:

- + Hoàn thiện chức năng Đăng ký chỗ ở, đảm bảo quy trình kiểm tra dữ liệu, tạo hồ sơ sinh viên, tạo hợp đồng và cập nhật số lượng người ở trong phòng một cách tự động.
- + Xây dựng đầy đủ các module Quản lý sinh viên, phòng, tài chính, hỗ trợ CRUD, tìm kiếm, lọc, phân trang và đảm bảo tính ràng buộc dữ liệu.
- + Chức năng Thống kê đã minh họa trực quan dữ liệu theo năm, giúp người quản trị nắm được các chỉ số quan trọng như số hợp đồng, doanh thu, số lượng phòng, trạng thái phòng,...
- + Tích hợp ChatBot, hỗ trợ người dùng tra cứu nhanh thông tin, kết hợp dữ liệu gợi ý câu hỏi từ bảng FAQ.
- + Thiết kế cơ sở dữ liệu quan hệ MySQL tối ưu, đảm bảo tính toàn vẹn và dễ mở rộng.
- + Xây dựng được giao diện website thân thiện, đẹp mắt với khả năng responsive, giúp người dùng thao tác dễ dàng, thuận tiện trên mọi thiết bị.

2. Hạn chế

- Dù đã hoàn thành các mục tiêu đề ra, vẫn còn một số điểm cần cải thiện:
- + Chưa tích hợp xác thực hai lớp (Two-Factor Authentication) để tăng cường bảo mật cho tài khoản quản trị.
- + Hệ thống ChatBot mới chỉ dừng ở mức trả lời dựa trên dữ liệu FAQ, chưa áp dụng trí tuệ nhân tạo để hiểu ngữ nghĩa sâu hoặc xử lý hội thoại phức tạp.
- + Hệ thống hiện chưa có module gửi thông báo qua email hoặc SMS như nhắc hạn hợp đồng, nhắc thanh toán.
- + Chưa triển khai cơ chế phân quyền chi tiết theo mức độ (ví dụ: quản trị viên cấp cao, nhân viên ký túc xá...).

3. Hướng phát triển

- Để hệ thống hoàn thiện, hiện đại và phục vụ tốt hơn nhu cầu thực tế, trong tương lai đề tài có thể được mở rộng theo các hướng sau:

- + Tích hợp xác thực hai lớp (Two-Factor Authentication) để tăng cường bảo mật cho tài khoản quản trị.
- + Tích hợp công nghệ AI vào ChatBot, giúp chatbot hiểu ngữ cảnh, hỗ trợ người dùng tự nhiên và thông minh hơn.
- + Bổ sung hệ thống thông báo tự động qua email hoặc SMS để nhắc hạn hợp đồng, nhắc thanh toán, thông báo lịch bảo trì phòng...
- + Nâng cấp phân quyền người dùng, chia nhỏ vai trò quản trị theo từng nghiệp vụ (quản lý phòng, tài chính, hợp đồng...).

TÀI LIỆU THAM KHẢO

1. Tài liệu ReactJS: <https://vi.legacy.reactjs.org/docs/getting-started.html>
2. Tài liệu React Router: <https://reactrouter.com/home>
3. Tài liệu JavaScript: <https://javascript.info/document>
4. Tài liệu PHP:
https://users.soict.hust.edu.vn/trungtt/uploads/slides/LTWeb_Bai05.pdf
5. Tài liệu MySQL: <https://dev.mysql.com/doc/>

