**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

Trường Công nghệ thông tin và Truyền thông

A red and white logo

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

ĐỀ TÀI: THIẾT KẾ HƯỚNG ĐỐI TƯỢNG CHO BÀI TOÁN QUẢN LÝ KÝ TÚC XÁ

Môn: Lập trình hướng đối tượng – IT3103 (Mã lớp: 151964)

Nhóm: 2

|  |  |  |
| --- | --- | --- |
| Giảng viên hướng dẫn | : | Thầy Lê Đức Hậu |
| Họ và tên – MSSV | : | Nghiêm Xuân Diện – 20215007  Trần Đình Dũng – 20215014  Ngọ Doãn Ngọc – 20215103  Phạm Minh Thắng - 20215139 |

**PHÂN CÔNG CÔNG VIỆC VÀ ĐÓNG GÓP**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | MSSV | Đóng góp (%) | Công việc |
| Nghiêm Xuân Diện | 20215007 | 20 | GUI, làm báo cáo |
| Ngọ Doãn Ngọc (Leader) | 20215103 | 40 | Thiết kế erd, Database, Thiết kế usecase, class Diagram, làm package Service |
| Trần Đình Dũng | 20215014 | 20 | Tạo database, DAO, connect Database, làm báo cáo |
| Phạm Minh Thắng | 20215139 | 20 | Làm Slide, model |

Mục lục

[**Lời nói đầu** 3](#_Toc185782064)

[**1. Giới thiệu** 4](#_Toc185782065)

[1.1. Đặt vấn đề 4](#_Toc185782066)

[1.2. Yêu cầu hệ thống 4](#_Toc185782067)

[1.3. Công nghệ sử dụng 5](#_Toc185782068)

[**2. Thiết kế hệ thống** 5](#_Toc185782069)

[2.1. Biểu đồ UML 6](#_Toc185782070)

[2.1.1. Usecase Diagram 6](#_Toc185782071)

[2.1.2. Class Diagram 6](#_Toc185782072)

[2.2. Sơ đồ thực thể liên kết ERD 7](#_Toc185782073)

[2.3. Kiến trúc hệ thống 8](#_Toc185782074)

[**3. Phân tích và mô tả hệ thống** 8](#_Toc185782075)

[3.1. Package **DAO** 8](#_Toc185782076)

[3.2. Package **connectionDB** 9](#_Toc185782077)

[3.3. Package **service** 9](#_Toc185782078)

[3.4. Package **model** 10](#_Toc185782079)

[3.5. Package **gui** 10](#_Toc185782080)

[**4. Kỹ thuật lập trình hướng đối tượng đã áp dụng** 11](#_Toc185782081)

[**5. Demo các chức năng của chương trình** 13](#_Toc185782082)

[**6. Kết luận** 19](#_Toc185782083)

# **Lời nói đầu**

Trong bối cảnh hiện đại, việc quản lý ký túc xá tại các trường học không chỉ dừng lại ở việc cung cấp chỗ ở mà còn đòi hỏi một hệ thống tổ chức thông tin hiệu quả, minh bạch và chính xác. Đặc biệt, với sự gia tăng số lượng sinh viên cùng các yêu cầu về chất lượng dịch vụ ngày càng cao, việc ứng dụng công nghệ vào quản lý ký túc xá trở thành một nhu cầu thiết yếu.

Đề tài *“Thiết kế hướng đối tượng cho bài toán Quản lý ký túc xá”* được thực hiện nhằm mục tiêu xây dựng một mô hình hệ thống quản lý thông tin ký túc xá cơ bản nhưng toàn diện. Thông qua việc áp dụng các nguyên lý thiết kế hướng đối tượng, bài báo cáo không chỉ hướng đến việc giải quyết các vấn đề hiện tại mà còn đảm bảo tính linh hoạt, khả năng mở rộng và dễ dàng bảo trì hệ thống trong tương lai.

Bài báo cáo bao gồm các phần từ phân tích bài toán, xây dựng mô hình lớp cho đến thiết kế các phương thức quản lý cụ thể. Trong đó, các thực thể chính như sinh viên, phòng, hợp đồng, và các loại phí được phân tích chi tiết, đảm bảo phản ánh sát thực tế hoạt động của ký túc xá.

Chúng em hy vọng rằng bài báo cáo này sẽ không chỉ cung cấp những kiến thức nền tảng về thiết kế hệ thống mà còn là tài liệu tham khảo hữu ích cho các nhà quản lý và các cá nhân, tổ chức quan tâm đến việc hiện đại hóa công tác quản lý ký túc xá.

Trân trọng cảm ơn!

# **1. Giới thiệu**

## 1.1. Đặt vấn đề

Ký túc xá là môi trường sinh sống và học tập quan trọng đối với sinh viên, đặc biệt là những sinh viên xa nhà. Việc quản lý ký túc xá không chỉ đơn thuần là sắp xếp chỗ ở mà còn phải đảm bảo các khía cạnh như quản lý thông tin cá nhân của sinh viên, theo dõi hợp đồng, xử lý phí sinh hoạt, và duy trì cơ sở vật chất. Với quy mô ngày càng lớn và tính phức tạp trong vận hành, các phương pháp quản lý truyền thống dựa trên giấy tờ hoặc bảng tính không còn đáp ứng được yêu cầu thực tế.

Bên cạnh đó, sự phát triển của công nghệ thông tin đã mang đến những công cụ và phương pháp giúp tối ưu hóa công tác quản lý. Việc thiết kế một hệ thống quản lý ký túc xá theo hướng đối tượng không chỉ giúp tổ chức và lưu trữ thông tin một cách có hệ thống mà còn tạo điều kiện để tự động hóa các quy trình, tăng hiệu quả làm việc và giảm thiểu sai sót trong quá trình quản lý.

Tuy nhiên, để xây dựng một hệ thống quản lý ký túc xá hiệu quả, việc phân tích bài toán và áp dụng các nguyên lý thiết kế hướng đối tượng là yếu tố cốt lõi. Việc mô hình hóa các thực thể và mối quan hệ giữa chúng trong hệ thống là bước đầu tiên để đảm bảo rằng hệ thống có thể đáp ứng các nhu cầu thực tế, đồng thời giữ được tính linh hoạt để mở rộng hoặc nâng cấp trong tương lai.

Với mục tiêu đó, đề tài *“Thiết kế hướng đối tượng cho bài toán Quản lý ký túc xá”* được thực hiện, tập trung giải quyết các vấn đề cơ bản như quản lý thông tin sinh viên, phòng ở, hợp đồng, và các loại phí. Qua đó, đề tài không chỉ cung cấp giải pháp kỹ thuật mà còn góp phần nâng cao hiệu quả quản lý ký túc xá trong thực tế.

## 1.2. Yêu cầu hệ thống

Yêu cầu hệ thống: Xây dựng một hệ thống quản lý kí túc xá với các chức năng hữu ích, tiện lợi cho người dùng

## 1.3. Công nghệ sử dụng

Cơ sở dữ liệu: **MySQL**

App: Ngôn ngữ lập trình **JAVA, JAVASWING**

# **2. Thiết kế hệ thống**

Xây dựng hệ thống quản lý ký túc xá nơi người quản lý ký túc xá có thể quản lý sinh viên, các phòng, hợp đồng thuê phòng, các chi phí phát sinh, ...

**Các thực thể chính:**

* **Sinh viên**: Thông tin cá nhân (họ tên, ngày sinh, giới tính, quê quán,...), thông tin liên hệ, thông tin về phòng ở, thông tin về hợp đồng,...
* **Phòng**: Thông tin về phòng (số phòng, loại phòng, số giường, giá phòng,...), danh sách sinh viên ở.
* **Hợp đồng**: Thông tin về hợp đồng (mã hợp đồng, ngày bắt đầu, ngày kết thúc, giá phòng, hình thức thanh toán,...).
* **Phí**: Thông tin về các loại phí (phí điện, phí nước, phí vệ sinh,...).

**Quan hệ giữa các thực thể:**

* Một người quản lý có thể quản lý nhiều sinh viên, nhiều phòng, hợp đồng, phí
* Một sinh viên có thể được quản lý bởi một hoặc nhiều người quản lý
* Mỗi sinh viên thuộc về 1 phòng và chỉ có 1 hợp đồng.
* Một hợp đồng liên kết 1 sinh viên và 1 phòng
* Một sinh viên có nhiều bản ghi phí
* Một phòng có nhiều sinh viên

## 2.1. Biểu đồ UML

### 2.1.1. Usecase Diagram



### 2.1.2. Class Diagram

Ảnh có chứa văn bản, chữ viết tay, biểu đồ, Hình chữ nhật

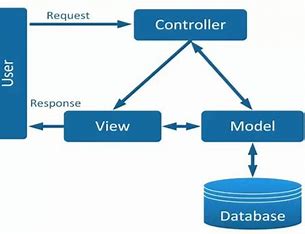
Mô tả được tạo tự động

## 2.2. Sơ đồ thực thể liên kết ERD

Ảnh có chứa hình vẽ, bản phác thảo, Nghệ thuật vẽ nét đơn, màu trắng

Mô tả được tạo tự động

## 2.3. Kiến trúc hệ thống

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

# **3. Phân tích và mô tả hệ thống**

## 3.1. Package **DAO**

* **StudentDAO**
  + Quản lý dữ liệu sinh viên.
  + Thực hiện thêm, sửa, xóa sinh viên.
  + Tìm kiếm sinh viên theo tên hoặc mã.
  + Lấy danh sách toàn bộ sinh viên.
* **RoomDAO**
  + Quản lý dữ liệu phòng ký túc xá.
  + Lấy thông tin phòng theo số phòng.
  + Quản lý danh sách tất cả phòng.
  + Cập nhật trạng thái phòng (trống hoặc đang sử dụng).
* **ContractDAO**
  + Quản lý dữ liệu hợp đồng thuê phòng.
  + Tạo mới hợp đồng.
  + Lấy danh sách hợp đồng hiện tại.
* **FeeDAO**
  + Quản lý dữ liệu các khoản phí phát sinh.
  + Thêm mới, cập nhật, xóa các khoản phí.
  + Tìm kiếm các khoản phí theo loại phí.
  + Lấy danh sách toàn bộ phí.
* **Searchable (interface)**
  + Định nghĩa phương thức tìm kiếm chung.
  + Được triển khai bởi các lớp DAO cụ thể để hỗ trợ tìm kiếm dữ liệu theo tên.

## 3.2. Package **connectionDB**

* **DBConnection**
  + Cung cấp kết nối với cơ sở dữ liệu
  + Các thuộc tính:
    - URL: Đường dẫn đến cơ sở dữ liệu.
    - USER: Tên người dùng cơ sở dữ liệu.
    - PASSWORD: Mật khẩu cơ sở dữ liệu.
  + Phương thức:
    - getConnection(): Tạo kết nối tới cơ sở dữ liệu và trả về 1 đối tượng Connection.

## 3.3. Package **service**

* Chứa các lớp xử lý logic nghiệp vụ của hệ thống, thực hiện thao tác với DAO và cung cấp dữ liệu cho GUI.
* **StudentService**
  + Xử lý các yêu cầu liên quan đến sinh viên.
  + Gọi các phương thức trong StudentDAO để thêm, cập nhật, xóa hoặc tìm kiếm thông tin sinh viên.
* **RoomService**
  + Quản lý thông tin phòng.
  + Các chức năng bao gồm thêm mới, cập nhật thông tin hoặc xóa phòng thông qua RoomDAO.
* **FeeService**
  + Xử lý nghiệp vụ liên quan đến các loại phí.
  + Bao gồm thêm phí, cập nhật, xóa phí, và tìm kiếm phí.
* **ContractService**
  + Quản lý các hợp đồng thuê phòng.
  + Gọi ContractDAO để thực hiện tạo mới hợp đồng hoặc lấy danh sách hợp đồng hiện tại.

## 3.4. Package **model**

* Chứa các lớp biểu diễn các thực thể trong hệ thống.
* **Student**
  + Đại diện cho thông tin sinh viên.
  + Các thuộc tính: mã sinh viên, tên đầy đủ, ngày sinh, giới tính, số điện thoại, email.
  + Phương thức: GetterAndSetter() để truy xuất và chỉnh sửa dữ liệu.
* **Room**
  + Đại diện thông tin phòng.
  + Các thuộc tính: mã phòng, số phòng, loại phòng, số giường, giá phòng, trạng thái (có đang được sử dụng hay không).
  + Kế thừa thêm các lớp con:
    - ARoom: Phòng có thêm các chi phí phụ (additionalFee).
    - BRoom: Loại phòng cơ bản không có thêm phụ phí.
* **Contract**
  + Đại diện cho hợp đồng thuê phòng.
  + Các thuộc tính: mã hợp đồng, ngày bắt đầu, ngày kết thúc, giá phòng, hình thức thanh toán.
  + Phương thức: GetterAndSetter() để truy cập và chỉnh sửa dữ liệu.
* **Fee**
  + Đại diện các loại phí phát sinh.
  + Các thuộc tính: mã phí, loại phí (FeeType: Điện, Nước, Vệ sinh), số tiền, phương thức thanh toán.
  + Phương thức: GetterAndSetter().
* **FeeType (Enumeration)**
  + Liệt kê các loại phí trong hệ thống.
    - WATER (Phí nước).
    - ELECTRICITY (Phí điện).
    - CLEANING (Phí vệ sinh).

## 3.5. Package **gui**

* Giao diện người dùng để quản lý các thành phần trong hệ thống.
* **MainFrame**
  + Giao diện chính, chứa các bảng điều khiển (panel) khác như StudentPanel, RoomPanel, FeePanel, và ContractPanel.
  + Các thuộc tính: JTabbedPane, JMenuBar, JMenu.
* **StudentPanel**
  + Giao diện quản lý sinh viên.
  + Chứa các nút và chức năng:
    - Thêm, xóa, sửa thông tin sinh viên.
    - Tìm kiếm sinh viên theo tên.
* **RoomPanel**
  + Giao diện quản lý phòng.
  + Chứa các bảng hiển thị thông tin phòng, chức năng thêm, sửa, xóa phòng.
* **FeePanel**
  + Giao diện quản lý phí.
  + Cung cấp chức năng thêm phí, sửa phí, xóa phí, và tìm kiếm phí.
* **ContractPanel**
  + Giao diện quản lý hợp đồng thuê phòng.
  + Hiển thị danh sách hợp đồng, cho phép thêm, sửa hoặc xóa hợp đồng.
* **StatisticPanel**
  + Hiển thị các thống kê liên quan đến phòng, sinh viên, hợp đồng hoặc phí.
  + Bao gồm bảng dữ liệu (JTable) và các nút làm mới, tải dữ liệu thống kê.

# **4. Kỹ thuật lập trình hướng đối tượng đã áp dụng**

**Kế thừa**

Lớp ARoom và BRoom được kế thừa từ lớp Room.

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, màu trắng

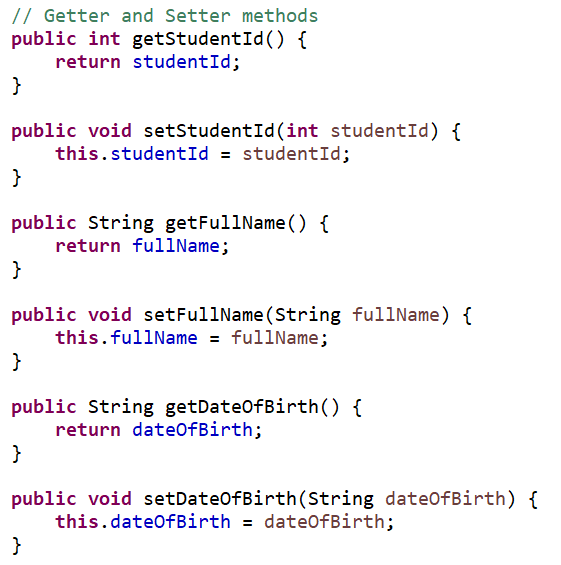
Mô tả được tạo tự động



**Lợi ích:** Tính kế thừa giúp tăng khả năng tái sử dụng. Khi một lớp kế thừa hoặc dẫn xuất một lớp khác, thì nó có thể truy cập tất cả các chức năng của lớp mà nó kế thừa. Giúp lớp “ARoom”, “BRoom” được hưởng tất cả các thuộc tính và phương thức public của lớp “Room”.

**Đóng gói**

Thuộc tính của các lớp “Student”, “Room”, “Fee”, “Contract”, … được đóng gói bằng cách sử dụng các phương thức getter và setter để truy cập và cập nhật giá trị của chúng. Nó giúp ngăn chặn việc truy cập trực tiếp vào các thành phần bên trong của đối tượng.



**Đa hình**

Các lớp “RoomDAO”, “StudentDAO”, “FeeDAO” triển khai interface “Search”

Ảnh có chứa văn bản, Phông chữ, màu trắng, hàng

Mô tả được tạo tự độngẢnh có chứa văn bản, Phông chữ, hàng

Mô tả được tạo tự độngẢnh có chứa văn bản, Phông chữ, hàng, màu trắng

Mô tả được tạo tự động

Phương thức searchByName là một hợp đồng chung, buộc tất cả các lớp triển khai phải cung cấp phiên bản cụ thể của nó. “RoomDAO”, “FeeDAO” và “StudentDAO” đều triển khai Search, nhưng mỗi lớp cung cấp logic cụ thể cho phương thức searchByName để phù hợp với ngữ cảnh của lớp đó.

**Lợi ích:** Nếu ta cần thêm một lớp khác, ví dụ ContractDAO, để tìm kiếm hợp đồng, bạn chỉ cần triển khai Search và viết logic riêng mà không cần sửa đổi các lớp hiện có. Đồng thời, Code làm việc với interface (Search) thay vì lớp cụ thể, giúp dễ dàng thay đổi và bảo trì.

**Trừu tượng**



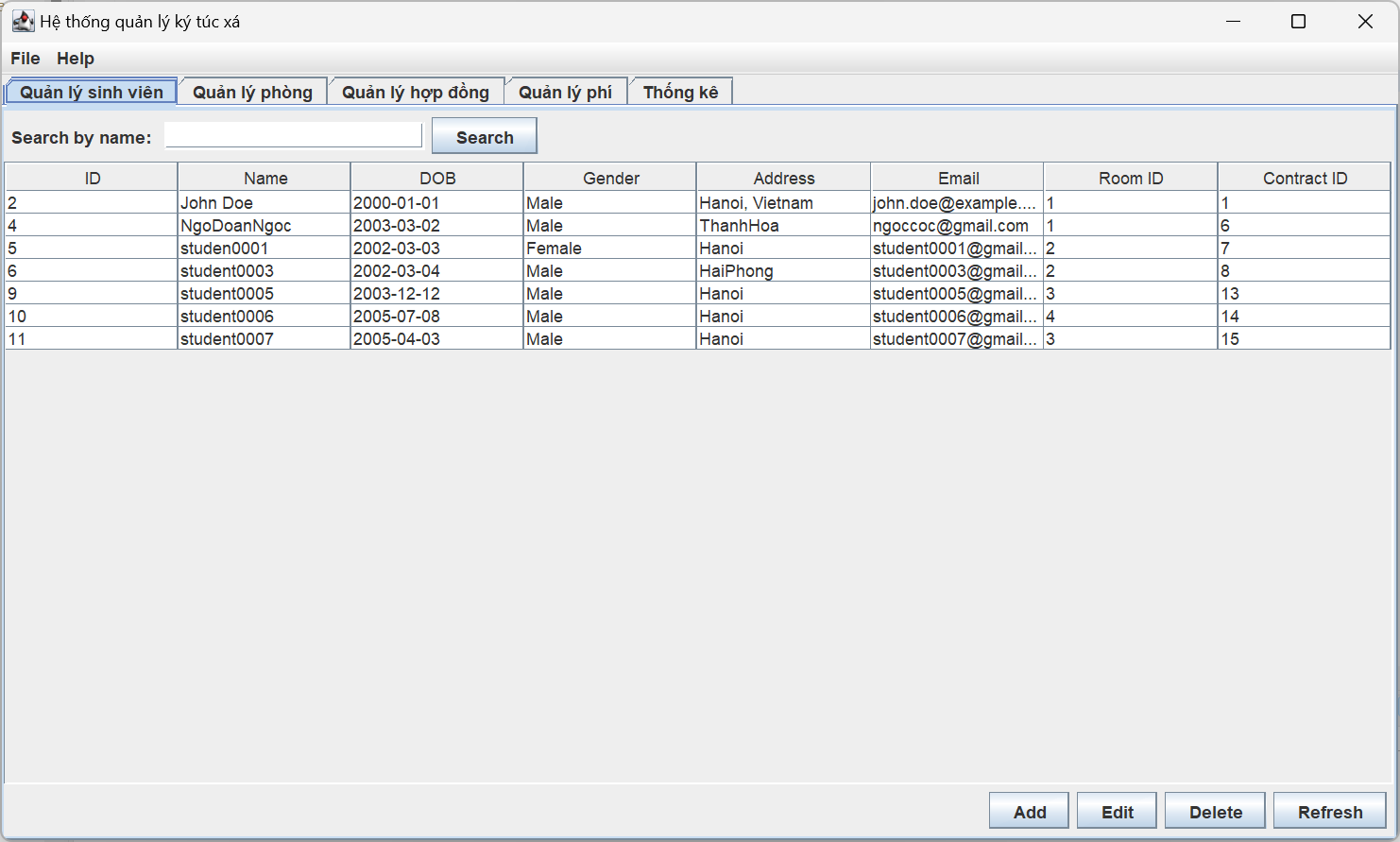
Interface Search định nghĩa một phương thức chung: searchByName(String name), mà không cung cấp chi tiết cài đặt.

Các lớp triển khai (RoomDAO, StudentDAO) phải tự định nghĩa cách thực hiện cụ thể của phương thức này, nhưng chúng được che giấu chi tiết thực hiện đối với người dùng.

**Lợi ích:** Người dùng chỉ cần biết rằng họ có thể tìm kiếm đối tượng theo tên thông qua searchByName, mà không cần quan tâm đến cách dữ liệu được tìm kiếm (truy vấn SQL, danh sách trong bộ nhớ, v.v.).

# **5. Demo các chức năng của chương trình**

Giao diện chương trình

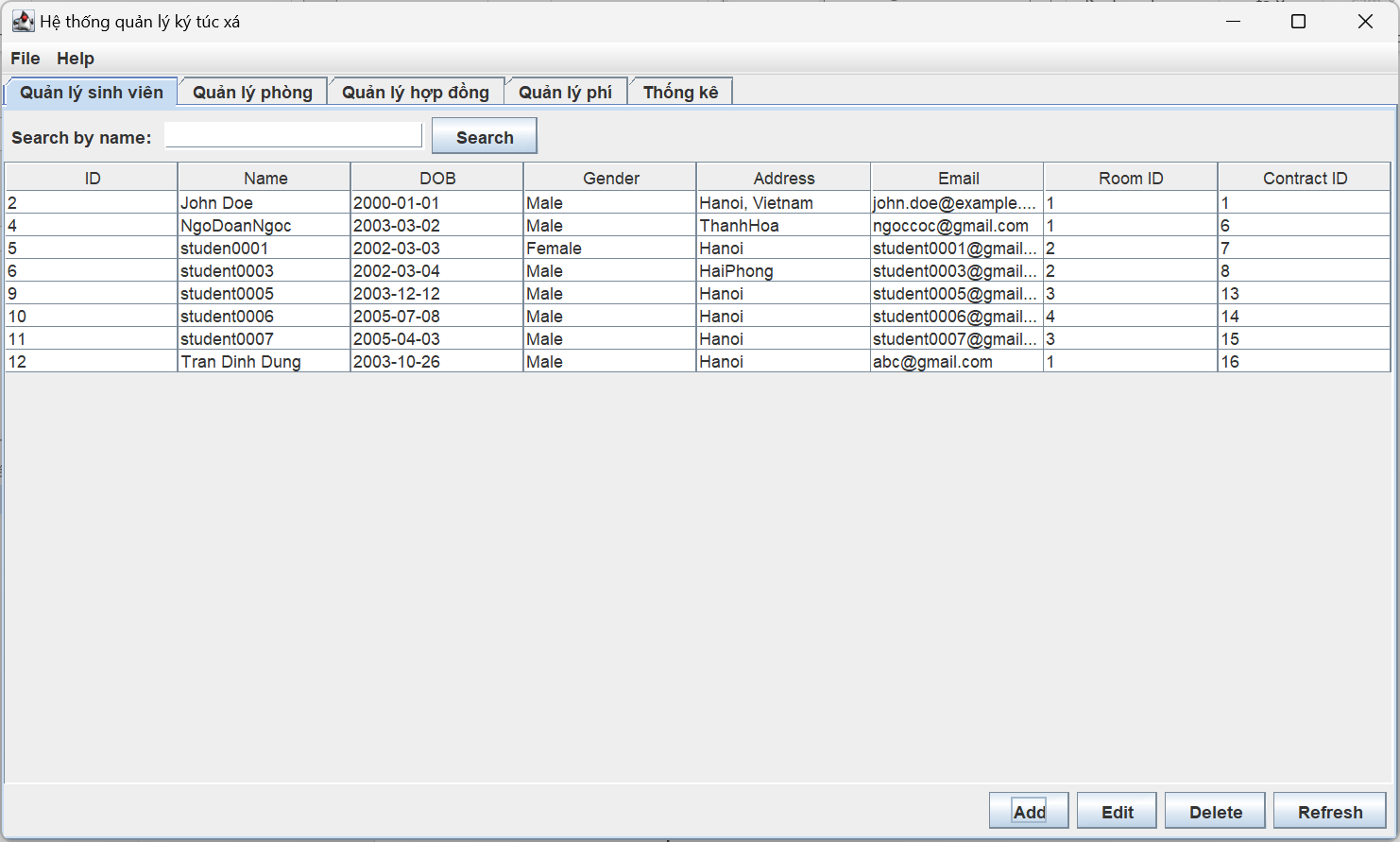


- Chức năng thêm sinh viên:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Sinh viên được thêm



- Chức năng sửa thông tin:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

Thông tin đã được sửa

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động

- chức năng xóa sinh viên:

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động

Sinh viên đã được xóa

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, số

Mô tả được tạo tự động

- Chức năng quản lý phòng:

Thêm phòng:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Xóa phòng:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Sửa phòng:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

- chức năng quản lý hợp đồng:

Ảnh có chứa văn bản, ảnh chụp màn hình, số, màn hình

Mô tả được tạo tự động

- chức năng quản lý phí:

Thêm phí:  
Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Cập nhật phí:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

Xóa phí:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

- Thống kê:

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

# **6. Kết luận**

Qua quá trình nghiên cứu và thực hiện, đề tài *“Thiết kế hướng đối tượng cho bài toán Quản lý ký túc xá”* đã hoàn thành các mục tiêu đề ra, bao gồm phân tích bài toán, xây dựng mô hình lớp, và thiết kế các phương thức quản lý cụ thể. Hệ thống quản lý ký túc xá được xây dựng dựa trên nguyên lý thiết kế hướng đối tượng không chỉ giải quyết được các yêu cầu cơ bản như quản lý sinh viên, phòng ở, hợp đồng và phí, mà còn đảm bảo tính linh hoạt, dễ mở rộng và bảo trì.

Mô hình đã mô tả chi tiết các thực thể chính (Sinh viên, Phòng, Hợp đồng, Phí) cùng các mối quan hệ giữa chúng, cung cấp một nền tảng vững chắc để triển khai thực tế. Hệ thống này không chỉ giúp giảm thiểu khối lượng công việc quản lý thủ công mà còn tăng độ chính xác và hiệu quả trong việc xử lý dữ liệu, đồng thời cải thiện trải nghiệm cho cả người quản lý và sinh viên.

Tuy nhiên, hệ thống mới chỉ dừng lại ở mức mô hình cơ bản và chưa xét đến các yếu tố phức tạp hơn như tích hợp thanh toán trực tuyến, quản lý phản hồi từ sinh viên, hoặc tối ưu hóa quy trình ra quyết định. Đây sẽ là những hướng nghiên cứu và phát triển trong tương lai nhằm hoàn thiện hơn nữa hệ thống quản lý ký túc xá.

Chúng tôi hy vọng rằng kết quả từ đề tài sẽ không chỉ là một giải pháp kỹ thuật mà còn góp phần vào việc nâng cao chất lượng và hiệu quả quản lý ký túc xá trong thực tiễn.