BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA: CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**TIỂU LUẬN**

**XÂY DỰNG CHƯƠNG TRÌNH QUẢN LÝ DANH SÁCH HOÁ ĐƠN TIỀN ĐIỆN CỦA KHÁCH HÀNG.**

**MÔN:** **THIẾT KẾ VÀ XÂY DỰNG PHẦN MỀM**

Ngành: **KỸ THUẬT PHẦN MỀM**

Giảng viên hướng dẫn: **Th.S LÊ HUỲNH PHƯỚC**

Sinh viên thực hiện: **TRƯƠNG NGỌC HÀO**

MSSV: **22140336**

Lớp: **221402**

*TP. Hồ Chí Minh, tháng 11, năm 2024*

**Khoa: CÔNG NGHỆ THÔNG TIN**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: Thiết kế và xây dựng phần mềm**

**1. Họ và tên sinh viên: Trương Ngọc Hào**

**2. Tên đề tài: Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng.**

**3. Nhận xét:**

*a) Những kết quả đạt được:*

………………………………………………………………………………………….

………………………………………………………………………………………….

………………………………………………………………………………………….………………………………………………………………………………………….

*b)* *Những hạn chế:*

………………………………………………………………………………………….………………………………………………………………………………………….………………………………………………………………………………………….………………………………………………………………………………………….

**4. Điểm đánh giá** *(theo thang điểm 10, làm tròn đến 0.5):*

Sinh viên: Trương Ngọc Hào MSSV: 22140336

Điểm số: ……….…… Điểm chữ: ……………………………………………………..

|  |  |
| --- | --- |
|  | *TP. HCM, ngày … tháng … năm 20……*  **Giảng viên chấm thi**  *(Ký và ghi rõ họ tên)* |
|  |  |

**LỜI CAM ĐOAN**

Em xin cam đoan đề tài tiểu luận: **“Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng.”** do Trương Ngọc Hào tìm hiểu và thực hiện.

Em đã kiểm tra dữ liệu theo quy định hiện hành.

Kết quả bài làm của đề tài ***“*Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng.*”*** là trung thực và không sao chép từ bất kì bài tập của các cá nhân khác.

*TP. HCM, …. tháng 11 năm 2024.*

**Sinh viên cam đoan**

*(Ký và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến cô **Lê Huỳnh Phước.** Trong quá trình học tập và tìm hiểu môn “Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng. .”, em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tâm huyết và tận tình của cô. Cô đã giúp em tích lũy thêm nhiều kiến thức về môn học này để có thể hoàn thành được bài tiểu luận về đề tài ***“*Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng.*”.***

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, em kính mong nhận được những lời góp ý của cô để bài tiểu luận của em ngày càng hoàn thiện hơn.

Em xin chân thành cảm ơn!

**DANH MỤC HÌNH**

Hình 2.1: Sơ đồ Use Case cho ứng dụng Quản lý Công việc Cá nhân...........................12

Hình 3.2: Sơ đồ Class Diagram cho ứng dụng Quản lý Công việc Cá nhân..................16

Hình 3.3: Giao diện Đăng Nhập.....................................................................................18

Hình 3.4: Giao diện đăng kí...........................................................................................19

Hình 3.5: Giao diện Danh Sách Công Việc....................................................................20

Hình 3.6: Giao diện Thêm/Sửa Công Việc.....................................................................21

Hình 3.7: Giao diện Thống Kê Công Việc......................................................................22

Hình 3.8: Giao diện Cài Đặt Tài Khoản.........................................................................23

**MỤC LỤC**

[CHƯƠNG 1: TỔNG QUAN VỀ MÔN HỌC 1](#_Toc183253229)

[1.1. Giới thiệu về môn học 1](#_Toc183253230)

[1.1.1. Định nghĩa và mục tiêu của thiết kế phần mềm 2](#_Toc183253231)

[1.1.2. Vai trò của kiến trúc phần mềm trong dự án phần mềm 2](#_Toc183253232)

[1.1.3. Tổng quan về Clean Architecture và tầm quan trọng của nó 3](#_Toc183253233)

[1.2. Các khái niệm cơ bản liên quan đến Clean Architecture 3](#_Toc183253234)

[1.2.1. Kiến trúc phần mềm là gì? 3](#_Toc183253235)

[1.2.2. Nguyên tắc SOLID và sự liên quan đến Clean Architecture 4](#_Toc183253236)

[1.2.3. Các tầng trong Clean Architecture (Entities, Use Cases, Interface Adapters, Frameworks & Drivers) 5](#_Toc183253237)

[1.3. Mô tả bài toán 6](#_Toc183253238)

[1.3.1. Giới thiệu bài toán 6](#_Toc183253239)

[1.3.2. Quy tắc tính thành tiền 6](#_Toc183253240)

[1.3.3. Yêu cầu chức năng của hệ thống 7](#_Toc183253241)

[1.3.4. Mục tiêu của đề tài 7](#_Toc183253242)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG 9](#_Toc183253243)

[2.1. Cơ sở dữ liệu và các bảng dữ liệu 9](#_Toc183253244)

[2.1.1. Bảng invoice 9](#_Toc183253245)

[2.1.2. Bảng doiTuongKH 9](#_Toc183253246)

[2.1.3. Bảng account 10](#_Toc183253247)

[2.1.4. Dữ liệu mẫu 10](#_Toc183253248)

[2.1.5. Mô tả quan hệ giữa các bảng 12](#_Toc183253249)

[2.2. Các khái niệm liên quan đến đề tài 12](#_Toc183253250)

[2.2.1. Đặc điểm hóa đơn điện năng 12](#_Toc183253251)

[2.2.2. Tính toán hóa đơn 13](#_Toc183253252)

[2.3. Danh sách Use Case 14](#_Toc183253253)

[2.3.1. Danh sách Use Case chính 14](#_Toc183253254)

[2.4. Sơ đồ tuần tự (Sequence Diagram) 24](#_Toc183253255)

[2.4.1. Đăng nhập 24](#_Toc183253256)

[2.4.2. Đăng ký 25](#_Toc183253257)

[2.4.3. Quên mật khẩu 27](#_Toc183253258)

[2.4.4. Xóa hóa đơn tiền điện 30](#_Toc183253259)

[2.4.5. Sửa hóa đơn tiền điện 32](#_Toc183253260)

[2.4.6. Xuất toàn bộ hóa đơn 34](#_Toc183253261)

[2.4.7. Tìm kiếm hóa đơn 35](#_Toc183253262)

[2.4.8. Tính trung bình thành tiền khách hàng nước ngoài 36](#_Toc183253263)

[2.4.9. Tổng khách hàng từng loại của hóa đơn tiền điện 38](#_Toc183253264)

[2.4.10. Xuất hóa đơn tiền điện theo tháng 39](#_Toc183253265)

[2.5. Sơ đồ Lớp (Class Diagram) 40](#_Toc183253266)

[2.5.1. Sơ đồ Lớp cho chức năng Đăng nhập 41](#_Toc183253267)

[2.5.2. Sơ đồ Lớp cho chức năng Đăng ký 42](#_Toc183253268)

[2.5.3. Sơ đồ Lớp cho chức năng Quên mật khẩu 43](#_Toc183253269)

[2.5.4. Sơ đồ Lớp tổng quan cho ứng dụng Quản lý hóa đơn tiền điện 44](#_Toc183253270)

[2.5.5. Sơ đồ Lớp cho chức năng Thêm hóa đơn 46](#_Toc183253271)

[2.5.6. Sơ đồ Lớp cho chức năng Xóa hóa đơn 48](#_Toc183253272)

[2.5.7. Sơ đồ Lớp cho chức năng Sửa hóa đơn 50](#_Toc183253273)

[2.5.8. Sơ đồ Lớp cho chức năng Xuất toàn bộ hóa đơn 52](#_Toc183253274)

[2.5.9. Sơ đồ Lớp cho chức năng Tìm kiếm hóa đơn 53](#_Toc183253275)

[2.5.10. Sơ đồ Lớp cho chức năng Tính trung bình thành tiền khách hàng nước ngoài 55](#_Toc183253276)

[2.5.11. Sơ đồ Lớp cho chức năng Tổng số khách hàng từng loại của hóa đơn tiền điện 57](#_Toc183253277)

[2.5.12. Sơ đồ Lớp cho chức năng Xuất hóa đơn tiền điện theo tháng 59](#_Toc183253278)

[CHƯƠNG 3: KIỂM THỬ CHƯƠNG TRÌNH 61](#_Toc183253279)

[3.1. Kiểm thử chương trình 61](#_Toc183253280)

[3.1.1. Đăng nhập 61](#_Toc183253281)

[3.1.2. Đăng ký 62](#_Toc183253282)

[3.1.3. Quên mật khẩu 63](#_Toc183253283)

[3.1.4. Quản lý hóa đơn tiền điện 64](#_Toc183253284)

[3.1.5. Thêm hóa đơn tiền điện 65](#_Toc183253285)

[3.1.6. Xóa hóa đơn tiền điện 66](#_Toc183253286)

[3.1.7. Sửa hóa đơn tiền điện 67](#_Toc183253287)

[3.1.8. Xuất hóa đơn tiền điện theo tháng 68](#_Toc183253288)

[3.1.9. Tìm kiếm hóa đơn tiền điện 69](#_Toc183253289)

[3.1.10. Tính trung bình thành tiền nước ngoài 70](#_Toc183253290)

[3.1.11. Tính tổng khách hàng theo loại 70](#_Toc183253291)

[CHƯƠNG 4: TỔNG KẾT 71](#_Toc183253292)

[4.1. Kết quả đạt được 71](#_Toc183253293)

[4.2. Đánh giá ưu điểm và khuyết điểm 72](#_Toc183253294)

[4.3. Hướng phát triển và mở rộng đề tài 73](#_Toc183253295)

[TÀI LIỆU THAM KHẢO 75](#_Toc183253296)

# CHƯƠNG 1: TỔNG QUAN VỀ MÔN HỌC

## 1.1. Giới thiệu về môn học

Trong lĩnh vực phát triển phần mềm, thiết kế phần mềm đóng một vai trò trung tâm, là cầu nối giữa việc phân tích yêu cầu và hiện thực hóa sản phẩm phần mềm. Đây không chỉ đơn thuần là quá trình tạo ra mã nguồn mà còn là việc xác định cấu trúc tổng thể của hệ thống, thiết kế các thành phần, và cách chúng tương tác để đáp ứng được các yêu cầu nghiệp vụ và phi chức năng. Một thiết kế phần mềm hiệu quả là nền tảng để xây dựng các hệ thống phần mềm chất lượng cao, dễ bảo trì, mở rộng, và đáp ứng được những thay đổi trong tương lai.

Thiết kế phần mềm không chỉ giải quyết các vấn đề về kỹ thuật mà còn đảm bảo rằng phần mềm có thể hoạt động ổn định trong môi trường thực tế. Các quyết định thiết kế liên quan đến kiến trúc phần mềm, mô hình hóa dữ liệu, và cơ chế giao tiếp giữa các thành phần đều có ảnh hưởng lớn đến hiệu năng, khả năng mở rộng, và tính linh hoạt của hệ thống. Bên cạnh đó, thiết kế phần mềm cũng hướng đến việc tối ưu hóa thời gian phát triển và chi phí bảo trì thông qua các nguyên tắc thiết kế như SOLID, DRY (Don't Repeat Yourself), và KISS (Keep It Simple, Stupid).

Môn học "Thiết kế và Xây dựng Phần mềm" được xây dựng nhằm trang bị cho sinh viên những kiến thức cơ bản và nâng cao về thiết kế phần mềm. Mục tiêu của môn học không chỉ dừng lại ở việc cung cấp các lý thuyết nền tảng mà còn hướng đến việc giúp sinh viên hiểu sâu sắc cách áp dụng các phương pháp và công cụ thiết kế hiện đại vào thực tiễn. Đặc biệt, môn học nhấn mạnh vào việc giới thiệu các mô hình kiến trúc phần mềm phổ biến, chẳng hạn như Clean Architecture, Microservices, và Layered Architecture, giúp sinh viên nắm bắt được cách thức tổ chức và triển khai các thành phần trong một hệ thống lớn.

Thông qua môn học này, sinh viên sẽ được tiếp cận với quy trình thiết kế phần mềm từ việc phân tích yêu cầu, xác định các chức năng cần thiết, đến việc hiện thực hóa chúng thành một hệ thống hoàn chỉnh. Đồng thời, môn học cũng nhấn mạnh tầm quan trọng của việc kiểm thử thiết kế để đảm bảo rằng hệ thống đáp ứng được các tiêu chuẩn về chất lượng, hiệu suất, và bảo mật. Với các kiến thức và kỹ năng thu nhận được, sinh viên sẽ có nền tảng vững chắc để tham gia vào các dự án phát triển phần mềm thực tế, giải quyết những thách thức trong việc thiết kế và xây dựng các hệ thống phức tạp, hiện đại.

### 1.1.1. Định nghĩa và mục tiêu của thiết kế phần mềm

Thiết kế phần mềm là quá trình xác định cấu trúc, thành phần, giao diện và dữ liệu của một hệ thống phần mềm để đáp ứng các yêu cầu cụ thể. Mục tiêu chính của thiết kế phần mềm bao gồm:

* **Đáp ứng yêu cầu người dùng**: Đảm bảo rằng phần mềm được thiết kế để giải quyết các vấn đề và nhu cầu của người dùng cuối.
* **Tối ưu hóa hiệu suất**: Thiết kế hệ thống sao cho phần mềm hoạt động hiệu quả, nhanh chóng và sử dụng tài nguyên một cách hợp lý.
* **Dễ bảo trì và mở rộng**: Tạo ra một cấu trúc phần mềm dễ dàng để sửa chữa, nâng cấp và thêm các chức năng mới mà không làm gián đoạn hệ thống hiện tại.
* **Tính nhất quán và rõ ràng**: Đảm bảo rằng mọi thành phần của phần mềm được thiết kế một cách nhất quán, dễ hiểu và dễ quản lý.

### 1.1.2. Vai trò của kiến trúc phần mềm trong dự án phần mềm

Kiến trúc phần mềm là cấu trúc tổng thể của một hệ thống phần mềm, bao gồm các thành phần chính, mối quan hệ giữa chúng và các nguyên tắc hướng dẫn thiết kế và phát triển. Vai trò của kiến trúc phần mềm trong dự án phần mềm bao gồm:

* **Định hướng phát triển**: Kiến trúc phần mềm cung cấp một khung tổng thể để định hướng quá trình phát triển, giúp các nhà phát triển hiểu rõ cấu trúc và mục tiêu của hệ thống.
* **Quản lý phức tạp**: Giúp chia nhỏ hệ thống thành các thành phần dễ quản lý, giảm thiểu sự phức tạp và tăng tính linh hoạt trong việc phát triển.
* **Đảm bảo tính nhất quán**: Đảm bảo rằng tất cả các thành phần của hệ thống được thiết kế và triển khai một cách nhất quán, giảm thiểu lỗi và xung đột.
* **Hỗ trợ bảo trì và mở rộng**: Một kiến trúc tốt giúp dễ dàng bảo trì và mở rộng hệ thống khi cần thiết, giảm thiểu thời gian và chi phí cho các thay đổi sau này.

### 1.1.3. Tổng quan về Clean Architecture và tầm quan trọng của nó

Clean Architecture là một phong cách kiến trúc phần mềm được Robert C. Martin giới thiệu nhằm mục đích tạo ra các hệ thống phần mềm dễ bảo trì, dễ mở rộng và dễ kiểm thử. Clean Architecture tuân thủ các nguyên tắc SOLID và phân tách rõ ràng các thành phần của hệ thống theo các tầng khác nhau. Tầm quan trọng của Clean Architecture nằm ở khả năng:

* **Tách biệt các tầng**: Giúp phân tách rõ ràng giữa các thành phần nghiệp vụ (business logic) và các thành phần giao diện người dùng (UI), cơ sở dữ liệu, hoặc các dịch vụ bên ngoài khác.
* **Giảm sự phụ thuộc**: Các tầng cao hơn không phụ thuộc vào các tầng thấp hơn, đảm bảo rằng các thay đổi trong một tầng không ảnh hưởng đến các tầng khác.
* **Dễ kiểm thử**: Bằng cách tách biệt các thành phần, Clean Architecture giúp dễ dàng viết các bài kiểm thử đơn vị (unit tests) cho từng thành phần riêng lẻ.
* **Tính mở rộng và bảo trì**: Hệ thống được thiết kế theo Clean Architecture dễ dàng mở rộng và bảo trì, đáp ứng nhanh chóng các yêu cầu thay đổi hoặc thêm chức năng mới.

## 1.2. Các khái niệm cơ bản liên quan đến Clean Architecture

### 1.2.1. Kiến trúc phần mềm là gì?

Kiến trúc phần mềm là cấu trúc tổng thể của một hệ thống phần mềm, bao gồm các thành phần chính, mối quan hệ giữa chúng và các nguyên tắc hướng dẫn thiết kế và phát triển. Kiến trúc phần mềm không chỉ định nghĩa cách các thành phần được tổ chức mà còn xác định cách chúng tương tác với nhau để đạt được các mục tiêu chức năng và phi chức năng của hệ thống. Một kiến trúc phần mềm tốt sẽ đảm bảo rằng hệ thống có thể đáp ứng các yêu cầu về hiệu suất, bảo mật, khả năng mở rộng và dễ bảo trì.

### 1.2.2. Nguyên tắc SOLID và sự liên quan đến Clean Architecture

Nguyên tắc SOLID là một tập hợp các nguyên tắc thiết kế hướng đối tượng, được Robert C. Martin đề xuất để tạo ra các hệ thống phần mềm dễ bảo trì và mở rộng. SOLID bao gồm:

* **S - Single Responsibility Principle (SRP)**: Mỗi lớp chỉ nên có một lý do duy nhất để thay đổi, tức là chỉ chịu trách nhiệm về một nhiệm vụ cụ thể.
* **O - Open/Closed Principle (OCP)**: Các lớp nên mở để mở rộng nhưng đóng để sửa đổi. Điều này có nghĩa là bạn có thể thêm các chức năng mới mà không cần thay đổi mã nguồn hiện tại.
* **L - Liskov Substitution Principle (LSP)**: Các lớp con nên có thể thay thế cho lớp cha mà không làm hỏng tính đúng đắn của chương trình.
* **I - Interface Segregation Principle (ISP)**: Khách hàng không nên bị ép buộc phải phụ thuộc vào các giao diện mà họ không sử dụng.
* **D - Dependency Inversion Principle (DIP)**: Các module cấp cao không nên phụ thuộc vào các module cấp thấp. Cả hai nên phụ thuộc vào các abstraction (giao diện).

**Sự liên quan đến Clean Architecture:**

Clean Architecture tuân thủ các nguyên tắc SOLID để đảm bảo rằng hệ thống phần mềm được thiết kế một cách linh hoạt và dễ bảo trì. Bằng cách áp dụng SOLID, Clean Architecture giúp phân tách các tầng của hệ thống, giảm sự phụ thuộc giữa các thành phần và tăng tính mở rộng của hệ thống. Các nguyên tắc này đảm bảo rằng mỗi thành phần của hệ thống chỉ chịu trách nhiệm về một nhiệm vụ cụ thể, dễ dàng mở rộng mà không ảnh hưởng đến các thành phần khác và tạo điều kiện thuận lợi cho việc kiểm thử và bảo trì hệ thống.

### 1.2.3. Các tầng trong Clean Architecture (Entities, Use Cases, Interface Adapters, Frameworks & Drivers)

Clean Architecture được chia thành các tầng chính, mỗi tầng có vai trò và trách nhiệm riêng biệt, đảm bảo sự tách biệt rõ ràng giữa các thành phần của hệ thống. Các tầng này bao gồm:

1. **Entities:**
   * **Vai trò**: Chứa các đối tượng nghiệp vụ (business objects) và các quy tắc nghiệp vụ cốt lõi của hệ thống.
   * **Trách nhiệm**: Định nghĩa các thực thể chính và các thuộc tính, phương thức liên quan đến chúng. Entities không phụ thuộc vào bất kỳ tầng nào khác, đảm bảo tính độc lập và tái sử dụng cao.
2. **Use Cases:**
   * **Vai trò**: Chứa các quy trình nghiệp vụ cụ thể mà hệ thống thực hiện để đáp ứng yêu cầu của người dùng.
   * **Trách nhiệm**: Định nghĩa các trường hợp sử dụng (use cases) và các quy trình liên quan. Use Cases tương tác với Entities để thực hiện các nhiệm vụ nghiệp vụ.
3. **Interface Adapters:**
   * **Vai trò**: Chứa các adapter để chuyển đổi dữ liệu giữa các tầng bên ngoài và các tầng nội bộ.
   * **Trách nhiệm**: Bao gồm các controller, presenter và gateway để giao tiếp với các tầng bên ngoài như giao diện người dùng, cơ sở dữ liệu hoặc các dịch vụ bên ngoài. Interface Adapters giúp duy trì sự tách biệt giữa các tầng và đảm bảo rằng các tầng nội bộ không phụ thuộc trực tiếp vào các tầng bên ngoài.
4. **Frameworks & Drivers:**
   * **Vai trò**: Chứa các công nghệ và framework cụ thể mà hệ thống sử dụng để triển khai các chức năng.
   * **Trách nhiệm**: Bao gồm các công cụ như giao diện người dùng (UI), cơ sở dữ liệu, web frameworks và các dịch vụ bên ngoài khác. Tầng này là nơi các thành phần cụ thể được triển khai, nhưng không nên ảnh hưởng đến các tầng bên trong.

## 1.3. Mô tả bài toán

### 1.3.1. Giới thiệu bài toán

Đề tài tập trung vào việc xây dựng một chương trình quản lý danh sách hóa đơn tiền điện của khách hàng. Hệ thống cần quản lý thông tin của hai loại khách hàng chính:

* **Khách hàng Việt Nam**: Bao gồm mã khách hàng, họ tên, ngày ra hóa đơn (ngày, tháng, năm), đối tượng khách hàng (sinh hoạt, kinh doanh, sản xuất), số lượng (số KW tiêu thụ), đơn giá và định mức.
* **Khách hàng nước ngoài**: Bao gồm mã khách hàng, họ tên, ngày ra hóa đơn (ngày, tháng, năm), quốc tịch, số lượng và đơn giá.

### 1.3.2. Quy tắc tính thành tiền

* **Khách hàng Việt Nam**:
  + **Nếu số lượng ≤ định mức**:
    - Thành tiền = Số lượng \* Đơn giá.
  + **Nếu số lượng > định mức**:
    - Thành tiền = (Định mức \* Đơn giá) + (Số lượng vượt định mức \* Đơn giá \* 2.5).
    - Trong đó, Số lượng vượt định mức = Số lượng - Định mức.
* **Khách hàng nước ngoài**:
  + Thành tiền = Số lượng \* Đơn giá.

### 1.3.3. Yêu cầu chức năng của hệ thống

Hệ thống cần đáp ứng các chức năng sau:

* **Quản lý hóa đơn**:
  + **Thêm hóa đơn**: Cho phép thêm mới hóa đơn cho khách hàng.
  + **Xóa hóa đơn**: Xóa hóa đơn không còn hiệu lực hoặc bị sai.
  + **Sửa hóa đơn**: Chỉnh sửa thông tin hóa đơn khi cần thiết.
* **Thống kê và báo cáo**:
  + **Tính tổng số lượng cho từng loại khách hàng**: Tính tổng số KW tiêu thụ cho khách hàng Việt Nam và khách hàng nước ngoài.
  + **Tính trung bình thành tiền của khách hàng nước ngoài**: Tính giá trị trung bình của thành tiền đối với khách hàng nước ngoài.
  + **Xuất các hóa đơn trong một tháng cụ thể**: Liệt kê các hóa đơn được phát hành trong một tháng nào đó.
* **Tìm kiếm hóa đơn**: Cho phép tìm kiếm hóa đơn dựa trên các tiêu chí như mã khách hàng, tên khách hàng, ngày ra hóa đơn, v.v.

### 1.3.4. Mục tiêu của đề tài

* **Xây dựng một hệ thống quản lý hóa đơn điện năng hiệu quả**, đáp ứng đầy đủ các yêu cầu nghiệp vụ đã đề ra.
* **Áp dụng các nguyên tắc và mô hình thiết kế phần mềm hiện đại**, đặc biệt là Clean Architecture, để tạo ra một hệ thống dễ bảo trì, mở rộng và kiểm thử.
* **Tích hợp các khái niệm về thiết kế phần mềm** vào thực tế, giúp củng cố kiến thức và kỹ năng đã học trong môn học.

# CHƯƠNG 2: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

## 2.1. Cơ sở dữ liệu và các bảng dữ liệu

Để đáp ứng các yêu cầu chức năng của hệ thống quản lý hóa đơn tiền điện, cơ sở dữ liệu được thiết kế với các bảng chính sau:

### 2.1.1. Bảng invoice

* **Mục đích**: Lưu trữ thông tin về hóa đơn của khách hàng, bao gồm cả khách hàng Việt Nam và nước ngoài.
* **Cấu trúc bảng**:

|  |  |  |
| --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả |
| maKH | INT(11) | Mã khách hàng (Primary Key, Auto Increment) |
| tenKH | VARCHAR(255) | Họ tên khách hàng |
| ngayHD | DATE | Ngày ra hóa đơn |
| soLuong | INT(11) | Số lượng điện tiêu thụ (KW) |
| donGia | INT(11) | Đơn giá |
| quocTich | VARCHAR(255) | Quốc tịch (để trống nếu là khách hàng Việt Nam) |
| doiTuongKH | VARCHAR(255) | Đối tượng khách hàng (Sinh hoạt, Kinh doanh, Sản xuất) |
| dinhMuc | INT(11) | Định mức tiêu thụ (0 nếu là khách hàng nước ngoài) |

**Ghi chú**: Bảng invoice kết hợp thông tin của cả khách hàng Việt Nam và nước ngoài, với các trường quocTich, doiTuongKH, và dinhMuc để phân biệt.

### 2.1.2. Bảng doiTuongKH

* **Mục đích**: Lưu trữ các loại đối tượng khách hàng Việt Nam.
* **Cấu trúc bảng**:

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| maDT | INT(11) | Mã đối tượng (Primary Key, Auto Increment) |
| doiTuong | VARCHAR(255) | Loại đối tượng (Sinh hoạt, Kinh doanh, Sản xuất) |

**Ghi chú**: Bảng này giúp quản lý các loại đối tượng khách hàng và có thể liên kết với bảng invoice thông qua trường doiTuongKH.

### 2.1.3. Bảng account

* **Mục đích**: Quản lý thông tin người dùng và phân quyền truy cập hệ thống.
* **Cấu trúc bảng**:

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| username | VARCHAR(255) | Tên đăng nhập (Primary Key) |
| password | VARCHAR(255) | Mật khẩu |
| email | VARCHAR(255) | Địa chỉ email |
| isAdmin | BIT | Quyền quản trị (1: Admin, 0: Người dùng thường) |

**Ghi chú**: Trường isAdmin giúp phân biệt giữa quản trị viên và người dùng thông thường.

### 2.1.4. Dữ liệu mẫu

Dưới đây là một số dữ liệu mẫu được thêm vào các bảng để phục vụ quá trình phát triển và kiểm thử:

* **Bảng invoice**:

A screenshot of a computer

Description automatically generated

* Bảng doiTuongKH:

A screenshot of a computer

Description automatically generated

* Bảng account:

A screenshot of a computer

Description automatically generated

### 2.1.5. Mô tả quan hệ giữa các bảng

Các bảng trong cơ sở dữ liệu có mối quan hệ như sau:

* **Bảng invoice liên kết với bảng doiTuongKH** thông qua trường doiTuongKH để xác định loại đối tượng khách hàng Việt Nam. Điều này giúp hệ thống dễ dàng quản lý và phân loại khách hàng dựa trên mục đích sử dụng điện năng.
* **Bảng account quản lý thông tin người dùng hệ thống**, bao gồm cả quản trị viên và người dùng thông thường. Trường isAdmin trong bảng này xác định quyền truy cập và các chức năng mà người dùng có thể thực hiện trên hệ thống.

## 2.2. Các khái niệm liên quan đến đề tài

### 2.2.1. Đặc điểm hóa đơn điện năng

**Hóa đơn điện năng** là tài liệu ghi nhận thông tin về lượng điện tiêu thụ của khách hàng trong một khoảng thời gian nhất định. Đối với khách hàng Việt Nam, hóa đơn điện năng thường bao gồm các thông tin sau:

* **Mã khách hàng (maKH):** Mã định danh duy nhất của khách hàng.
* **Họ tên (tenKH):** Tên đầy đủ của khách hàng.
* **Ngày ra hóa đơn (ngayHD):** Ngày, tháng, năm mà hóa đơn được phát hành.
* **Đối tượng khách hàng (doiTuongKH):** Phân loại khách hàng dựa trên mục đích sử dụng điện, bao gồm:
  + **Sinh hoạt:** Khách hàng sử dụng điện cho mục đích cá nhân, gia đình.
  + **Kinh doanh:** Khách hàng sử dụng điện cho các hoạt động kinh doanh.
  + **Sản xuất:** Khách hàng sử dụng điện cho các hoạt động sản xuất công nghiệp.
* **Số lượng điện tiêu thụ (soLuong):** Lượng điện đã tiêu thụ trong kỳ hóa đơn, được tính bằng kilowatt-giờ (kWh).
* **Đơn giá (donGia):** Giá tiền cho mỗi kilowatt-giờ điện năng.
* **Định mức (dinhMuc):** Mức điện năng tiêu thụ tối đa cho phép trong kỳ hóa đơn, áp dụng cho khách hàng Việt Nam.
* **Thành tiền (thanhTien):** Tổng số tiền cần thanh toán, được tính dựa trên số lượng điện tiêu thụ và đơn giá.
* **Quốc tịch (quocTich):** Thông tin về quốc tịch của khách hàng (chỉ áp dụng cho khách hàng nước ngoài).

### 2.2.2. Tính toán hóa đơn

**Tính toán hóa đơn điện năng** là quá trình xác định số tiền mà khách hàng cần phải thanh toán dựa trên lượng điện tiêu thụ và các yếu tố khác như đơn giá và định mức tiêu thụ. Cách tính toán có thể khác nhau tùy vào loại khách hàng và quy định của công ty điện lực. Dưới đây là các công thức tính toán hóa đơn cho khách hàng Việt Nam và nước ngoài:

* **Khách hàng Việt Nam:**
  + **Nếu số lượng điện tiêu thụ ≤ định mức:**  
    Thành tiền = Số lượng \* Đơn giá
  + **Nếu số lượng điện tiêu thụ > định mức:**  
    Thành tiền = Định mức \* Đơn giá + (Số lượng - Định mức) \* Đơn giá \* 2.5

Điều này nghĩa là nếu khách hàng tiêu thụ điện vượt quá định mức, họ sẽ phải trả mức giá cao hơn cho phần điện tiêu thụ vượt định mức.

* **Khách hàng nước ngoài:**  
  Thành tiền = Số lượng \* Đơn giá

Với khách hàng nước ngoài, việc tính toán thành tiền đơn giản hơn vì không áp dụng định mức tiêu thụ và mức giá tăng cho phần điện vượt định mức.

Ngoài ra, hóa đơn cũng có thể bao gồm các khoản phí phụ thu khác như thuế VAT, phí bảo trì, phí dịch vụ và các khoản phí khác tùy theo quy định của công ty điện lực.

## 2.3. Danh sách Use Case

Trong phần này, chúng tôi sẽ trình bày chi tiết các Use Case của hệ thống quản lý hóa đơn tiền điện. Mỗi Use Case bao gồm mô tả, đầu vào, đầu ra và quy trình chính, giúp hiểu rõ hơn về chức năng và cách thức hoạt động của hệ thống.

A black background with white ovals

Description automatically generated

*Hình 2.3.1: Sơ đồ Use Case tổng thể của hệ thống.*

### 2.3.1. Danh sách Use Case chính

**1. Quản lý người dùng**

**1.1. Đăng ký**

* **Mô tả**: Cho phép người dùng tạo tài khoản mới trong hệ thống.
* **Input**:
  + Tên tài khoản
  + Email
  + Mật khẩu
* **Output**:
  + Thông báo đăng ký thành công hoặc thất bại.
* **Quy trình chính**:
  + Người dùng nhập thông tin đăng ký.
  + Hệ thống kiểm tra tính hợp lệ của dữ liệu:
    - Các trường không được để trống.
    - Mật khẩu phải có độ dài tối thiểu 8 ký tự.
    - Email phải đúng định dạng.
    - **Nếu không hợp lệ**:
      * Thông báo lỗi và yêu cầu nhập lại.
  + Hệ thống kiểm tra tên tài khoản:
    - **Nếu tên tài khoản đã tồn tại**:
      * Thông báo lỗi và yêu cầu nhập tên khác.
    - **Nếu tên tài khoản chưa tồn tại**:
      * Lưu thông tin tài khoản vào cơ sở dữ liệu.
      * Thông báo đăng ký thành công.

**1.2. Đăng nhập**

* **Mô tả**: Cho phép người dùng xác thực để truy cập vào hệ thống.
* **Input**:
  + Tên tài khoản
  + Mật khẩu
* **Output**:
  + Chuyển đến giao diện quản lý nếu đăng nhập thành công.
  + Thông báo lỗi nếu đăng nhập thất bại.
* **Quy trình chính**:
  + Người dùng nhập tên tài khoản và mật khẩu.
  + Hệ thống kiểm tra tính hợp lệ:
    - Các trường không được để trống.
    - Mật khẩu phải có độ dài tối thiểu 8 ký tự.
    - **Nếu không hợp lệ**:
      * Thông báo lỗi và yêu cầu nhập lại.
  + Hệ thống xác thực thông tin:
    - **Nếu thông tin không chính xác**:
      * Thông báo lỗi và yêu cầu nhập lại.
    - **Nếu thông tin chính xác**:
      * Chuyển đến giao diện quản lý.

**1.3. Quên mật khẩu**

* **Mô tả**: Hỗ trợ người dùng đặt lại mật khẩu khi quên.
* **Input**:
  + Tên tài khoản
  + Email
* **Output**:
  + Thông báo đặt lại mật khẩu thành công hoặc thất bại.
* **Quy trình chính**:
  + Người dùng nhập tên tài khoản và email.
  + Hệ thống kiểm tra tính hợp lệ:
    - Các trường không được để trống.
    - Email phải đúng định dạng.
    - **Nếu không hợp lệ**:
      * Thông báo lỗi và yêu cầu nhập lại.
  + Hệ thống kiểm tra thông tin tài khoản:
    - **Nếu không khớp**:
      * Thông báo lỗi và yêu cầu kiểm tra lại.
    - **Nếu khớp**:
      * Tạo mật khẩu mới.
      * Gửi mật khẩu mới đến email người dùng.
      * Thông báo đặt lại mật khẩu thành công.

**1.4. Đăng xuất**

* **Mô tả**: Cho phép người dùng thoát khỏi hệ thống.
* **Input**:
  + Yêu cầu đăng xuất.
* **Output**:
  + Chuyển đến giao diện đăng nhập.
* **Quy trình chính**:
  + Người dùng chọn chức năng đăng xuất.
  + Hệ thống hủy phiên làm việc hiện tại.
  + Chuyển đến giao diện đăng nhập.

**2. Quản lý hóa đơn**

**2.1. Thêm hóa đơn**

* **Mô tả**: Cho phép thêm mới một hóa đơn tiền điện vào hệ thống.
* **Input**:
  + Họ tên khách hàng
  + Ngày ra hóa đơn
  + Đối tượng khách hàng (Sinh hoạt, Kinh doanh, Sản xuất)
  + Số lượng điện tiêu thụ
  + Đơn giá
  + Quốc tịch (nếu là khách hàng nước ngoài)
  + Định mức (nếu là khách hàng Việt Nam)
* **Output**:
  + Thông báo thêm hóa đơn thành công hoặc thất bại.
* **Quy trình chính**:
  + Người dùng nhập thông tin hóa đơn.
  + Hệ thống kiểm tra tính hợp lệ:
    - Các trường bắt buộc không được để trống.
    - Số lượng và đơn giá phải là số dương.
    - Ngày ra hóa đơn phải đúng định dạng.
    - **Nếu không hợp lệ**:
      * Thông báo lỗi và yêu cầu nhập lại.
  + Hệ thống lưu hóa đơn vào cơ sở dữ liệu.
  + Thông báo thêm hóa đơn thành công.

A black screen with white text

Description automatically generated

*Hình 2.1: Usecase thêm hoá đơn tiền điện*

**2.2. Sửa hóa đơn**

* **Mô tả**: Cho phép chỉnh sửa thông tin của một hóa đơn hiện có.
* **Input**:
  + Mã hóa đơn
  + Thông tin cần chỉnh sửa
* **Output**:
  + Thông báo sửa hóa đơn thành công hoặc thất bại.
* **Quy trình chính**:
  + Người dùng chọn hóa đơn cần sửa.
  + Hệ thống hiển thị thông tin hiện tại của hóa đơn.
  + Người dùng chỉnh sửa thông tin.
  + Hệ thống kiểm tra tính hợp lệ:
    - **Nếu không hợp lệ**:
      * Thông báo lỗi và yêu cầu chỉnh sửa lại.
  + Hệ thống cập nhật thông tin trong cơ sở dữ liệu.
  + Thông báo sửa hóa đơn thành công.

A black background with a black square

Description automatically generated with medium confidence

*Hình 2.2: Sửa hoá đơn tiền điện*

**2.3. Xóa hóa đơn tiền điện**

* **Mô tả**: Cho phép xóa một hóa đơn khỏi hệ thống.
* **Input**:
  + Mã hóa đơn
* **Output**:
  + Thông báo xóa hóa đơn thành công hoặc thất bại.
* **Quy trình chính**:
  + Người dùng chọn hóa đơn cần xóa.
  + Hệ thống yêu cầu xác nhận việc xóa.
  + **Nếu xác nhận**:
    - Hệ thống xóa hóa đơn khỏi cơ sở dữ liệu.
    - Thông báo xóa hóa đơn thành công.
  + **Nếu hủy bỏ**:
    - Quá trình xóa bị hủy.

A screen shot of a phone

Description automatically generated

*Hình 2.3: Xoá hoá đơn tiền điện*

**2.4. Tìm kiếm hóa đơn**

* **Mô tả**: Cho phép tìm kiếm hóa đơn dựa trên các tiêu chí.
* **Input**:
  + Tiêu chí tìm kiếm (Mã khách hàng, Tên khách hàng, Ngày ra hóa đơn, v.v.)
  + Thông tin tìm kiếm
* **Output**:
  + Danh sách hóa đơn phù hợp hoặc thông báo không tìm thấy.
* **Quy trình chính**:
  + Người dùng nhập tiêu chí và thông tin tìm kiếm.
  + Hệ thống kiểm tra tính hợp lệ:
    - **Nếu không hợp lệ**:
      * Thông báo lỗi và yêu cầu nhập lại.
  + Hệ thống truy vấn cơ sở dữ liệu.
  + **Nếu tìm thấy**:
    - Hiển thị danh sách hóa đơn.
  + **Nếu không tìm thấy**:
    - Thông báo không tìm thấy hóa đơn phù hợp.

A black background with a black square

Description automatically generated with medium confidence

*Hình 2.4: Usecase tìm kiếm hoá đơn tiền điện*

**2.5. Xuất hóa đơn theo tháng**

* **Mô tả**: Cho phép xuất danh sách hóa đơn trong một tháng cụ thể.
* **Input**:
  + Tháng cần xuất (từ 1 đến 12)
* **Output**:
  + Danh sách hóa đơn trong tháng hoặc thông báo không có hóa đơn.
* **Quy trình chính**:
  + Người dùng chọn tháng cần xuất.
  + Hệ thống kiểm tra tính hợp lệ:
    - **Nếu không hợp lệ**:
      * Thông báo lỗi và yêu cầu nhập lại.
  + Hệ thống truy vấn cơ sở dữ liệu.
  + **Nếu có hóa đơn**:
    - Hiển thị danh sách hoặc cho phép xuất ra file.
  + **Nếu không có hóa đơn**:
    - Thông báo không có hóa đơn trong tháng đã chọn.

A black background with a black square

Description automatically generated with medium confidence

*Hình 2.5: Xuất hoá đơn tiền điện theo tháng*

**2.6. Tính tổng khách hàng từng loại**

* **Mô tả**: Thống kê tổng số khách hàng theo loại (Việt Nam, Nước ngoài).
* **Input**:
  + Loại khách hàng cần thống kê
* **Output**:
  + Số lượng khách hàng tương ứng
* **Quy trình chính**:
  + Người dùng chọn loại khách hàng.
  + Hệ thống truy vấn cơ sở dữ liệu để đếm số lượng khách hàng.
  + Hiển thị kết quả thống kê.

A black background with a black square

Description automatically generated with medium confidence

*Hình 2.6: Tính tổng khách hàng từng loại*

**2.7. Tính trung bình thành tiền khách hàng nước ngoài**

* **Mô tả**: Tính giá trị trung bình của thành tiền đối với khách hàng nước ngoài.
* **Input**:
  + Không có (tính cho tất cả khách hàng nước ngoài)
* **Output**:
  + Giá trị trung bình thành tiền
* **Quy trình chính**:
  + Hệ thống truy vấn cơ sở dữ liệu lấy hóa đơn của khách hàng nước ngoài.
  + Tính tổng thành tiền và số lượng hóa đơn.
  + Tính giá trị trung bình.
  + Hiển thị kết quả.

A black screen with white text

Description automatically generated

*Hình 2.7: Usecase tính trung bình thành tiền khách hàng nước ngoài*

## 2.4. Sơ đồ tuần tự (Sequence Diagram)

Sơ đồ tuần tự mô tả cách các thành phần trong hệ thống tương tác với nhau theo trình tự thời gian để thực hiện một chức năng cụ thể. Trong phần này, chúng tôi trình bày sơ đồ tuần tự cho các luồng nghiệp vụ chính của hệ thống quản lý hóa đơn tiền điện, tập trung vào chức năng Đăng nhập.

### 2.4.1. Đăng nhập

**Mô tả luồng nghiệp vụ:**

1. Người dùng mở ứng dụng và chọn chức năng **Đăng nhập**.
2. Giao diện người dùng (UI) hiển thị form đăng nhập.
3. Người dùng nhập tên đăng nhập và mật khẩu, sau đó nhấn nút **Đăng nhập**.
4. UI gửi thông tin đăng nhập đến **Bộ điều khiển (Controller)**.
5. Controller chuyển thông tin đến **Lớp nghiệp vụ (Business Logic Layer)**.
6. Business Logic Layer gọi **Dịch vụ xác thực (Authentication Service)** để xác thực người dùng.
7. Authentication Service truy vấn **Cơ sở dữ liệu (Database)** để kiểm tra thông tin đăng nhập.
8. Database trả về kết quả xác thực cho Authentication Service.
9. Authentication Service trả kết quả cho Business Logic Layer.
10. Business Logic Layer thông báo kết quả cho Controller.
11. Controller cập nhật UI với kết quả đăng nhập:
    * Nếu thành công: Chuyển người dùng đến giao diện chính.
    * Nếu thất bại: Hiển thị thông báo lỗi và yêu cầu nhập lại.

**Sơ đồ tuần tự:** A white screen shot of a computer screen

Description automatically generated

### 2.4.2. Đăng ký

Mô tả luồng nghiệp vụ:

1. Người dùng mở ứng dụng và chọn chức năng Đăng ký.
2. Giao diện người dùng (UI) hiển thị form đăng ký.
3. Người dùng nhập thông tin đăng ký (Tên đăng nhập, Email, Mật khẩu) và nhấn nút Đăng ký.
4. UI gửi dữ liệu đăng ký đến Bộ điều khiển (Controller).
5. Controller chuyển dữ liệu đến Lớp nghiệp vụ (Business Logic Layer).
6. Business Logic Layer gọi Dịch vụ xác thực đăng ký (Registration Service) để kiểm tra thông tin người dùng.
7. Registration Service thực hiện:
   * Kiểm tra tính hợp lệ của dữ liệu:
     + Kiểm tra trường bắt buộc (Tên đăng nhập, Email, Mật khẩu).
     + Kiểm tra định dạng email.
     + Mật khẩu phải có độ dài tối thiểu.
     + Nếu không hợp lệ, trả về lỗi.
   * Kiểm tra tài khoản đã tồn tại trong cơ sở dữ liệu:
     + Nếu tài khoản đã tồn tại, trả về lỗi.
     + Nếu tài khoản chưa tồn tại, tiếp tục.
8. Dữ liệu hợp lệ sẽ được lưu vào cơ sở dữ liệu thông qua Registration Service.
9. Database trả kết quả lưu thành công hoặc thất bại.
10. Registration Service thông báo kết quả cho Business Logic Layer.
11. Business Logic Layer chuyển kết quả đến Controller.
12. Controller cập nhật UI:
    * Nếu thành công: Hiển thị thông báo và chuyển người dùng đến giao diện đăng nhập.
    * Nếu thất bại: Hiển thị thông báo lỗi.

**Sơ đồ tuần tự:** A black and white image of a diagram

Description automatically generated

*Hình 2.4.2: Sơ đồ tuần tự cho chức năng Đăng ký.*

### 2.4.3. Quên mật khẩu

**Mô tả luồng nghiệp vụ:**

1. Người dùng mở giao diện ứng dụng và chọn chức năng **Quên mật khẩu**.
2. Giao diện người dùng (UI) hiển thị form nhập thông tin tài khoản để đặt lại mật khẩu.
3. Người dùng nhập tên tài khoản, email và nhấn nút **Đặt lại mật khẩu**.
4. **Giao diện (UI)** gửi thông tin đến **Bộ điều khiển (Controller)**.
5. **Controller** chuyển thông tin đến **Lớp nghiệp vụ (Business Logic Layer)**.
6. **Business Logic Layer** gọi **Dịch vụ xử lý quên mật khẩu (Forgot Password Service)** để xác minh thông tin.
7. **Dịch vụ** thực hiện các bước:
   * Kiểm tra thông tin tài khoản và email từ cơ sở dữ liệu:
     + Nếu thông tin không hợp lệ:
       - Trả về lỗi và thông báo người dùng.
     + Nếu hợp lệ:
       - Cập nhật mật khẩu mới vào cơ sở dữ liệu.
8. Kết quả từ dịch vụ được trả về **Lớp nghiệp vụ (Business Logic Layer)**:
   * Nếu thành công:
     + Trả về thông báo thành công.
   * Nếu thất bại:
     + Trả về thông báo lỗi.
9. **Lớp nghiệp vụ** thông báo kết quả cho **Controller**.
10. **Controller** cập nhật **UI**:
    * Hiển thị thông báo đặt lại mật khẩu thành công hoặc thất bại.

**Sơ đồ tuần tự:** A black and white diagram

Description automatically generated

*Hình 2.4.3: Sơ đồ tuần tự cho chức năng Quên mật khẩu*

**Mô tả sơ đồ tuần tự tổng quan**

Sơ đồ tuần tự tổng quan này thể hiện luồng hoạt động chính trong hệ thống quản lý hóa đơn tiền điện, bao gồm các chức năng từ giao diện người dùng (UI) đến cơ sở dữ liệu (Database). Các chức năng chính được tích hợp vào sơ đồ gồm:

1. **Đăng nhập**: Xác thực thông tin người dùng để truy cập hệ thống.
2. **Đăng ký**: Tạo mới tài khoản người dùng.
3. **Quên mật khẩu**: Cập nhật mật khẩu cho người dùng quên thông tin đăng nhập.
4. **Thêm hóa đơn**: Lưu thông tin hóa đơn mới vào cơ sở dữ liệu.
5. **Tìm kiếm hóa đơn**: Tìm kiếm các hóa đơn dựa trên tiêu chí cụ thể.
6. **Xóa hóa đơn**: Xóa thông tin hóa đơn từ hệ thống.
7. **Xuất hóa đơn theo tháng**: Trích xuất dữ liệu hóa đơn theo tháng và năm.
8. **Tính toán, thống kê**: Tính tổng số khách hàng hoặc trung bình thành tiền khách hàng nước ngoài.

**Thành phần tham gia**

1. **Actor**: Người dùng hoặc quản trị viên tương tác với hệ thống.
2. **Boundary Objects**:
   * Giao diện đăng nhập, đăng ký, quên mật khẩu.
   * Giao diện hóa đơn (thêm, sửa, tìm kiếm, xóa).
3. **Controllers**: Điều phối các yêu cầu từ giao diện người dùng đến lớp nghiệp vụ.
4. **Use Case Objects**: Thực hiện logic nghiệp vụ cụ thể như xử lý đăng nhập, thêm hóa đơn, hoặc tìm kiếm hóa đơn.
5. **Service Objects**: Thực hiện các thao tác cụ thể như xác thực, thêm mới, tìm kiếm hoặc xóa dữ liệu.
6. **Database Objects**: Tương tác với cơ sở dữ liệu để lưu trữ hoặc truy xuất thông tin.

**Luồng chi tiết**

1. Người dùng gửi yêu cầu thông qua các giao diện khác nhau.
2. Bộ điều khiển (Controller) tiếp nhận yêu cầu và chuyển dữ liệu đến lớp nghiệp vụ tương ứng.
3. Lớp nghiệp vụ kiểm tra tính hợp lệ của dữ liệu và thực hiện xử lý.
4. Các dịch vụ (Service) tương tác với cơ sở dữ liệu để thực hiện các chức năng như:
   * Thêm thông tin hóa đơn.
   * Tìm kiếm dữ liệu hóa đơn.
   * Xóa hoặc sửa dữ liệu hóa đơn.
   * Tính toán và thống kê dữ liệu.
5. Kết quả xử lý được trả về giao diện để hiển thị thông báo cho người dùng.

**Sơ đồ tuần tự:** A black and white screen with white dots

Description automatically generated

*Hình 2.4.4: Sơ đồ tuần tự tổng quan hệ thống quản lý hóa đơn tiền điện.*

* + **Tầng giao diện (UI)**: Nhận thông tin đầu vào từ người dùng và hiển thị kết quả đầu ra.
  + **Tầng điều khiển (Controller)**: Điều phối dữ liệu giữa giao diện và lớp nghiệp vụ.
  + **Tầng nghiệp vụ (Use Case & Services)**: Xử lý các quy tắc nghiệp vụ và logic hệ thống.
  + **Tầng cơ sở dữ liệu (Database)**: Lưu trữ và quản lý thông tin hóa đơn, người dùng.

### 2.4.4. Xóa hóa đơn tiền điện

**Mô tả luồng nghiệp vụ**

1. Người dùng mở giao diện ứng dụng và chọn chức năng **Xóa hóa đơn**.
2. **Giao diện người dùng (UI)** hiển thị danh sách hóa đơn hoặc form nhập thông tin để chọn hóa đơn cần xóa.
3. Người dùng thực hiện một trong hai thao tác:
   * Chọn hóa đơn từ danh sách.
   * Nhập mã hóa đơn để tìm kiếm hóa đơn cần xóa.
4. Sau khi xác định hóa đơn cần xóa, người dùng nhấn nút **Xóa** để xác nhận.
5. **UI** gửi yêu cầu và thông tin hóa đơn cần xóa đến **Controller**.
6. **Controller** chuyển yêu cầu đến **Lớp nghiệp vụ (Business Logic Layer)**.
7. **Business Logic Layer** gọi **Dịch vụ hóa đơn (Invoice Service)** để xử lý:
   * **Invoice Service** thực hiện kiểm tra tính hợp lệ của thông tin:
     + Nếu thông tin không hợp lệ:
       - Trả về thông báo lỗi qua **Business Logic Layer**.
     + Nếu hợp lệ:
       - Thực hiện kiểm tra trong cơ sở dữ liệu:
         * Nếu hóa đơn không tồn tại:

Trả về thông báo lỗi.

* + - * + Nếu hóa đơn tồn tại:

Xóa hóa đơn khỏi cơ sở dữ liệu.

1. **Invoice Service** trả kết quả về **Business Logic Layer**.
2. **Business Logic Layer** thông báo kết quả về **Controller**.
3. **Controller** cập nhật giao diện người dùng:
   * Hiển thị thông báo thành công hoặc lỗi tương ứng.

**Sơ đồ tuần tự** A white text on a black background

Description automatically generated

*Hình 2.4.4: Sơ đồ tuần tự cho chức năng xoá hóa đơn tiền điện*

### 2.4.5. Sửa hóa đơn tiền điện

**Mô tả luồng nghiệp vụ**

1. Người dùng mở giao diện ứng dụng và chọn chức năng **Sửa hóa đơn**.
2. **Giao diện người dùng (UI)** hiển thị danh sách các hóa đơn hoặc cho phép nhập mã hóa đơn cần chỉnh sửa.
3. Người dùng chọn hóa đơn hoặc nhập mã hóa đơn để xác định hóa đơn cần sửa.
4. **UI** hiển thị form với các thông tin chi tiết hiện tại của hóa đơn để người dùng chỉnh sửa.
5. Người dùng thực hiện chỉnh sửa thông tin hóa đơn và nhấn nút **Cập nhật**.
6. **UI** gửi thông tin cập nhật đến **Controller**.
7. **Controller** chuyển yêu cầu đến **Lớp nghiệp vụ (Business Logic Layer)**.
8. **Business Logic Layer** thực hiện:
   * Kiểm tra tính hợp lệ của dữ liệu:
     + Các trường không được để trống.
     + Giá trị số phải dương.
     + Định dạng ngày hợp lệ.
   * Nếu dữ liệu không hợp lệ:
     + Trả về thông báo lỗi.
   * Nếu dữ liệu hợp lệ:
     + Gọi **Dịch vụ hóa đơn (Invoice Service)** để cập nhật dữ liệu.
9. **Dịch vụ hóa đơn (Invoice Service)**:
   * Kiểm tra sự tồn tại của hóa đơn trong cơ sở dữ liệu.
   * Nếu hóa đơn không tồn tại:
     + Trả về thông báo lỗi.
   * Nếu hóa đơn tồn tại:
     + Cập nhật thông tin trong cơ sở dữ liệu.
10. **Invoice Service** trả kết quả về **Business Logic Layer**:
    * Nếu thành công:
      + Trả về thông báo thành công.
    * Nếu thất bại:
      + Trả về thông báo lỗi.
11. **Business Logic Layer** thông báo kết quả cho **Controller**.
12. **Controller** cập nhật **UI**:
    * Hiển thị thông báo thành công hoặc lỗi.

**Sơ đồ tuần tự:** A white paper with black text

Description automatically generated with medium confidence

*Hình 2.4.5: Sơ đồ tuần tự cho chức năng Sửa hóa đơn tiền điện*

### 2.4.6. Xuất toàn bộ hóa đơn

**Mô tả luồng nghiệp vụ**

1. Người dùng mở giao diện ứng dụng và chọn chức năng **Xuất toàn bộ hóa đơn**.
2. **Giao diện người dùng (UI)** hiển thị danh sách hoặc cung cấp tùy chọn tải xuống toàn bộ hóa đơn.
3. Người dùng nhấn vào nút **Hiển thị hoặc Xuất toàn bộ hóa đơn**.
4. **UI** gửi yêu cầu tới **Controller**.
5. **Controller** chuyển yêu cầu đến **Lớp nghiệp vụ (Business Logic Layer)**.
6. **Business Logic Layer** gọi **Dịch vụ hóa đơn (Invoice Service)** để truy xuất danh sách hóa đơn.
7. **Invoice Service**:
   * Gửi yêu cầu truy vấn đến cơ sở dữ liệu để lấy toàn bộ hóa đơn.
   * Nhận danh sách hóa đơn từ cơ sở dữ liệu và trả về **Business Logic Layer**.
8. **Business Logic Layer** xử lý dữ liệu (nếu cần) và gửi danh sách hóa đơn cho **Controller**.
9. **Controller** cập nhật **UI**:
   * Hiển thị toàn bộ danh sách hóa đơn trên giao diện.
   * Cung cấp tùy chọn tải xuống danh sách hóa đơn (dưới dạng file PDF, Excel, v.v.).

**Sơ đồ tuần tự:** A diagram of a diagram

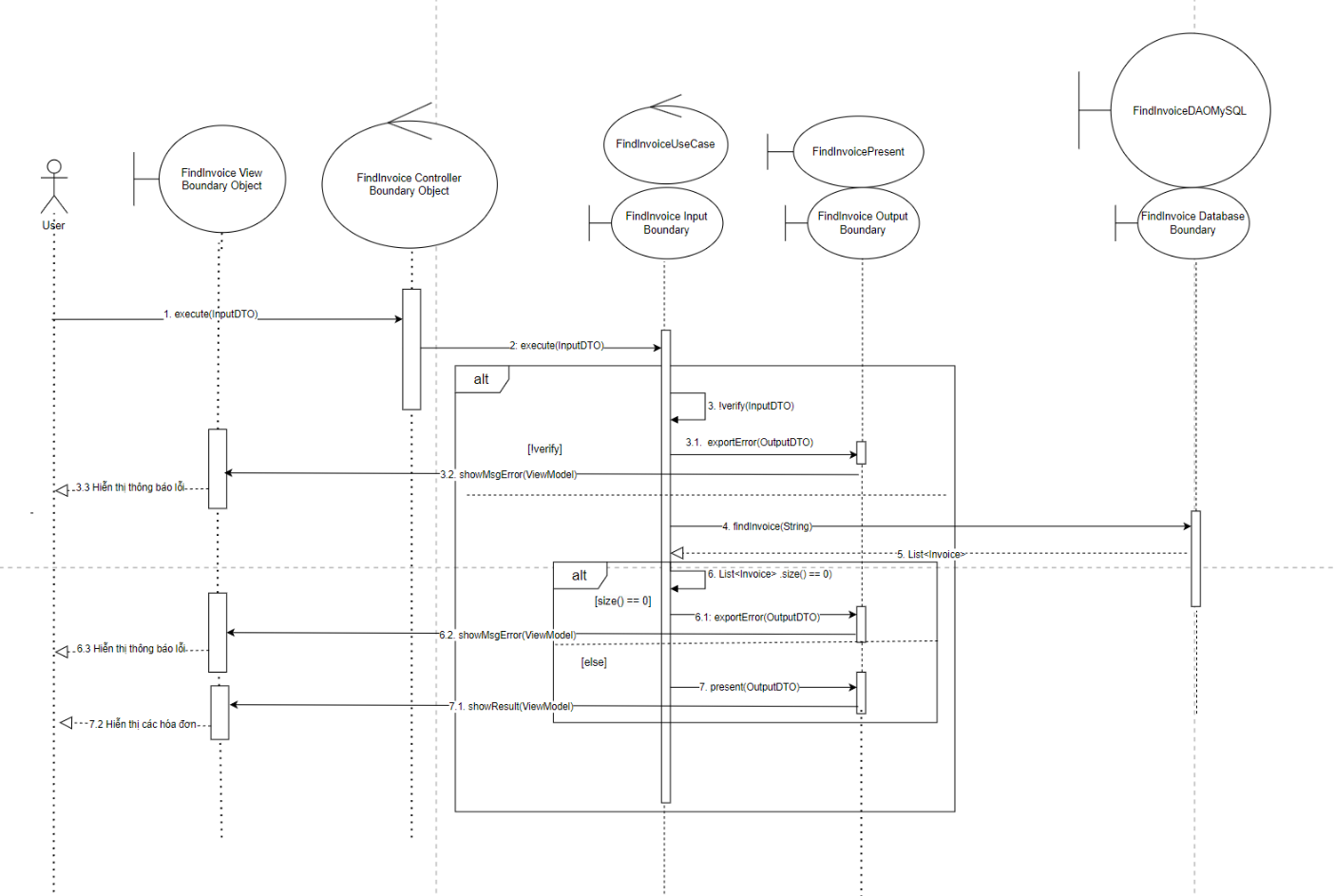
Description automatically generated with medium confidence

*Hình 2.4.6: Sơ đồ tuần tự cho chức năng Xuất toàn bộ hóa*

### 2.4.7. Tìm kiếm hóa đơn

**Mô tả luồng nghiệp vụ**

1. Người dùng mở giao diện ứng dụng và chọn chức năng **Tìm kiếm hóa đơn**.
2. **UI** hiển thị giao diện nhập thông tin tiêu chí tìm kiếm (mã khách hàng, tên khách hàng, ngày lập hóa đơn, v.v.).
3. Người dùng nhập thông tin cần tìm kiếm và nhấn nút **Tìm kiếm**.
4. **UI** gửi thông tin tìm kiếm đến **Controller**.
5. **Controller** chuyển dữ liệu đến **Lớp nghiệp vụ (Business Logic Layer)**.
6. **Business Logic Layer** gọi **Dịch vụ hóa đơn (Invoice Service)** để thực hiện tìm kiếm dựa trên tiêu chí được cung cấp.
7. **Dịch vụ hóa đơn** thực hiện truy vấn cơ sở dữ liệu để tìm kiếm hóa đơn phù hợp:
   * Nếu không tìm thấy hóa đơn:
     + Trả về lỗi và thông báo người dùng.
   * Nếu tìm thấy hóa đơn:
     + Trả về danh sách hóa đơn phù hợp.
8. **Lớp nghiệp vụ (Business Logic Layer)** xử lý kết quả:
   * Nếu không có kết quả:
     + Thông báo lỗi và yêu cầu nhập lại tiêu chí tìm kiếm.
   * Nếu có kết quả:
     + Chuyển danh sách hóa đơn phù hợp cho **Controller**.
9. **Controller** cập nhật **UI**:
   * Hiển thị danh sách hóa đơn tìm được hoặc thông báo lỗi nếu tìm kiếm không thành công.

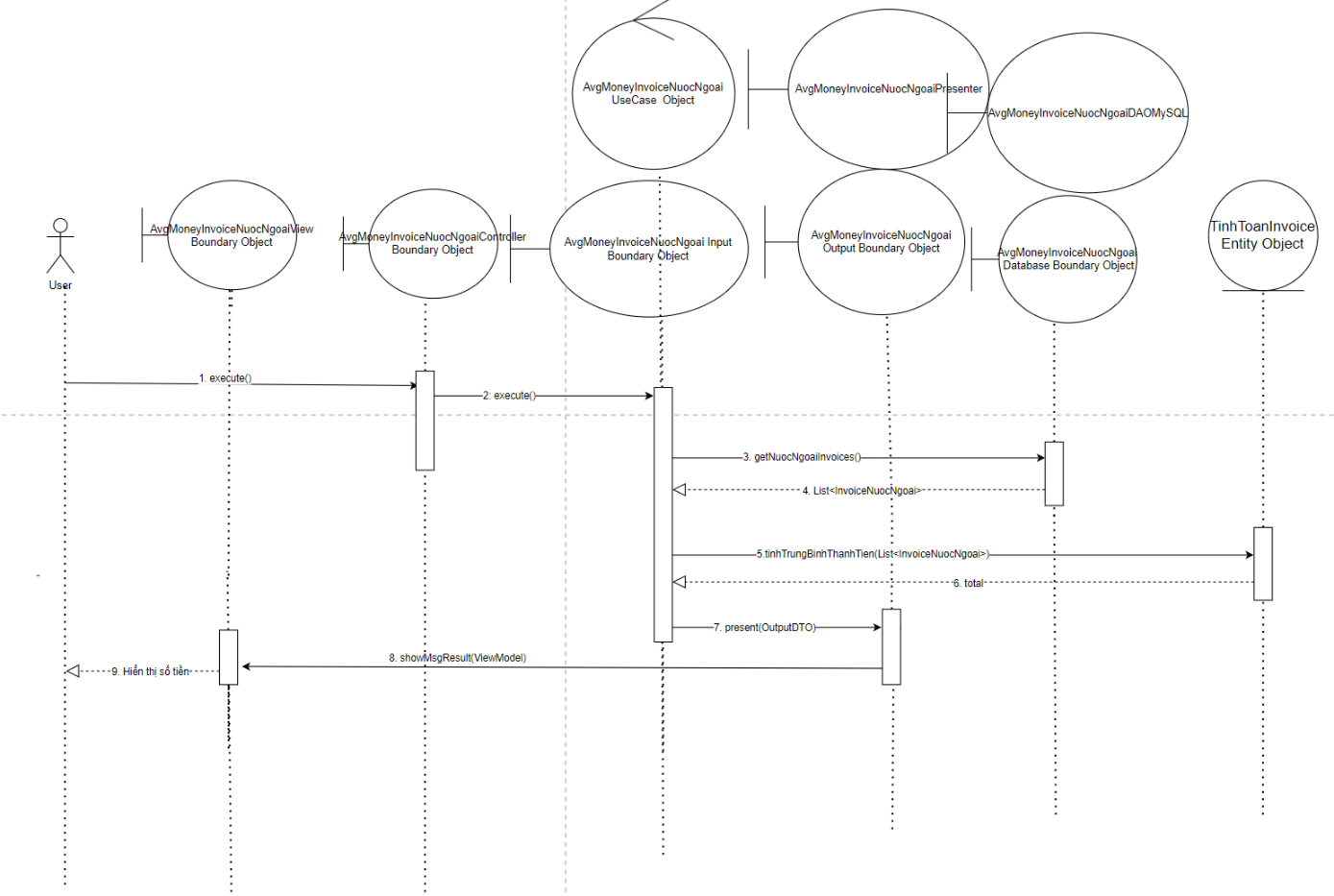
**Sơ đồ tuần tự:** 

*Hình 2.4.7: Sơ đồ tuần tự cho chức năng Tìm kiếm hóa*

### 2.4.8. Tính trung bình thành tiền khách hàng nước ngoài

**Mô tả luồng nghiệp vụ**

1. Người dùng mở giao diện ứng dụng và chọn chức năng **Tính trung bình thành tiền khách hàng nước ngoài**.
2. **UI** hiển thị giao diện yêu cầu và bắt đầu xử lý thông tin.
3. **UI** gửi yêu cầu tính toán đến **Controller**.
4. **Controller** chuyển yêu cầu đến **Lớp nghiệp vụ (Business Logic Layer)**.
5. **Business Logic Layer** gọi **Dịch vụ hóa đơn (Invoice Service)** để lấy danh sách hóa đơn của khách hàng nước ngoài.
6. **Invoice Service** thực hiện truy vấn cơ sở dữ liệu:
   * Lấy tất cả các hóa đơn liên quan đến khách hàng nước ngoài.
7. **Invoice Service** trả danh sách hóa đơn phù hợp về **Business Logic Layer**.
8. **Business Logic Layer** tính giá trị trung bình thành tiền:
   * Tính tổng thành tiền của các hóa đơn.
   * Đếm số lượng hóa đơn.
   * Tính giá trị trung bình = Tổng thành tiền / Số lượng hóa đơn.
9. Kết quả tính toán được trả về **Controller**.
10. **Controller** cập nhật **UI**:
    * Hiển thị kết quả trung bình thành tiền cho người dùng.

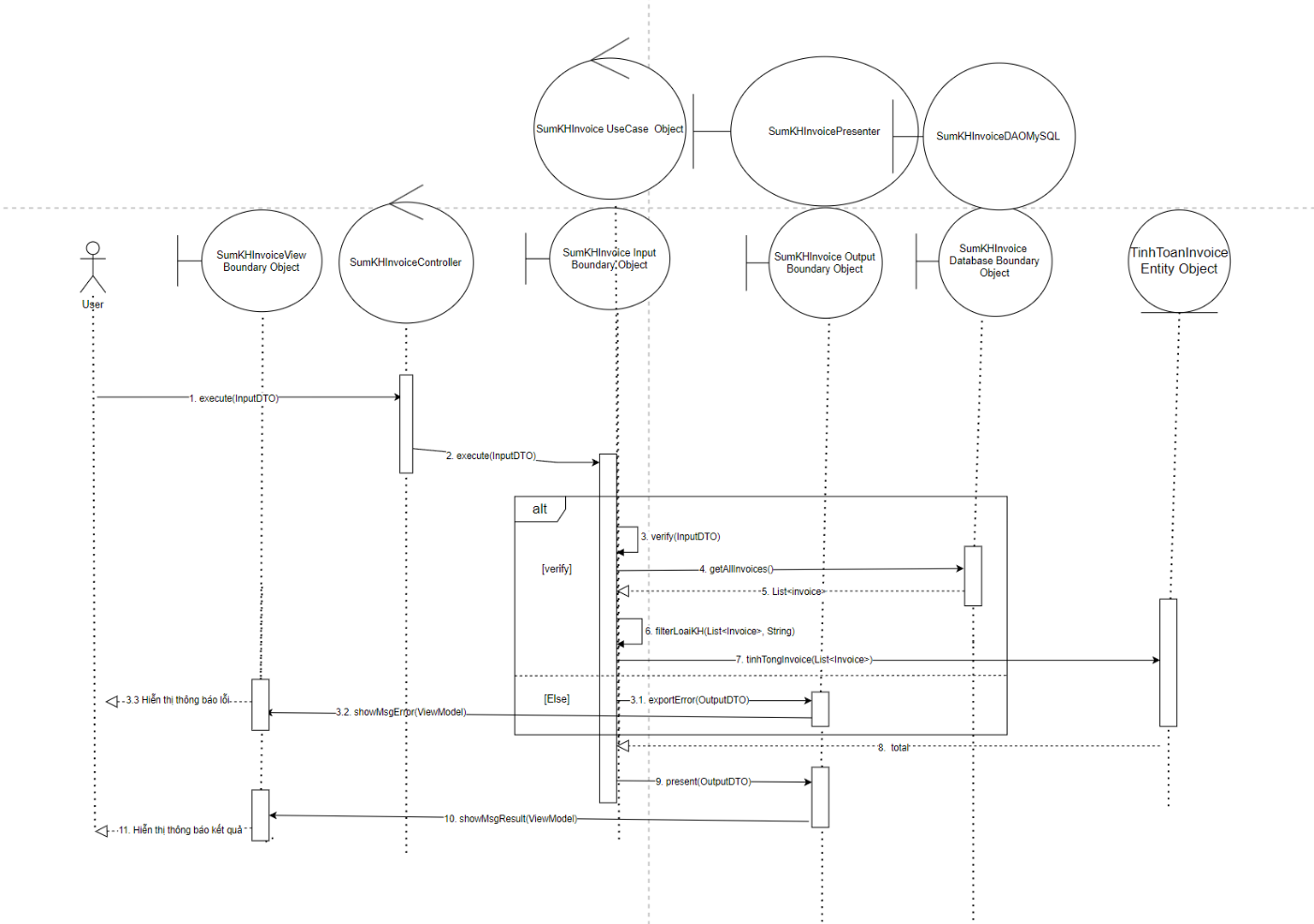
**Sơ đồ tuần tự:** 

*Hình 2.4.8: Sơ đồ tuần tự cho chức năng Tính trung bình thành tiền khách hàng nước ngoài*

### 2.4.9. Tổng khách hàng từng loại của hóa đơn tiền điện

**Mô tả luồng nghiệp vụ**

1. Người dùng mở giao diện ứng dụng và chọn chức năng **Tổng khách hàng từng loại**.
2. **UI** hiển thị yêu cầu người dùng chọn loại khách hàng (khách hàng Việt Nam hoặc nước ngoài).
3. Người dùng chọn loại khách hàng và nhấn nút **Xem kết quả**.
4. **UI** gửi thông tin loại khách hàng đến **Controller**.
5. **Controller** chuyển thông tin đến **Lớp nghiệp vụ (Business Logic Layer)**.
6. **Business Logic Layer** thực hiện các bước:
   * Xác minh dữ liệu loại khách hàng:
     + Nếu dữ liệu không hợp lệ, trả về thông báo lỗi.
     + Nếu dữ liệu hợp lệ, tiếp tục thực hiện xử lý.
   * Gọi **Dịch vụ hóa đơn (Invoice Service)** để lấy danh sách hóa đơn từ cơ sở dữ liệu.
   * Lọc danh sách hóa đơn theo loại khách hàng được chỉ định.
   * Thực hiện tính toán tổng số lượng khách hàng dựa trên danh sách đã lọc.
7. **Invoice Service** truy vấn cơ sở dữ liệu (Database) để lấy danh sách hóa đơn.
8. **Database** trả về danh sách hóa đơn.
9. **Invoice Service** xử lý danh sách, lọc theo loại khách hàng và tính tổng số khách hàng.
10. **Business Logic Layer** nhận kết quả xử lý và chuyển thông tin đến **Controller**.
11. **Controller** cập nhật **UI**:
    * Hiển thị kết quả tổng số khách hàng cho người dùng.
    * Nếu lỗi, hiển thị thông báo lỗi.

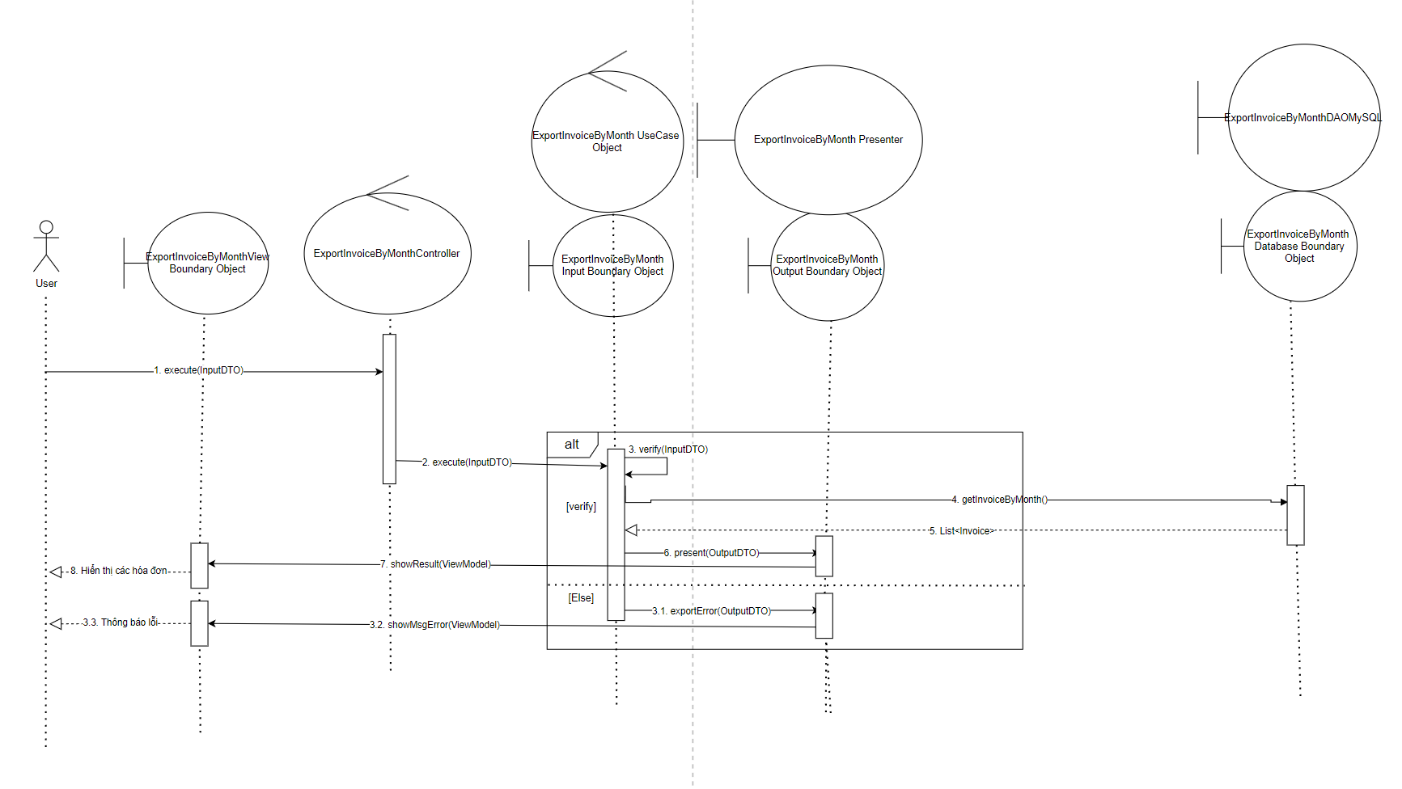
**Sơ đồ tuần tự:** 

*Hình 2.4.9: Sơ đồ tuần tự cho chức năng Tổng khách hàng từng*

### 2.4.10. Xuất hóa đơn tiền điện theo tháng

**Mô tả luồng nghiệp vụ:**

1. Người dùng mở ứng dụng và chọn chức năng **Xuất hóa đơn theo tháng**.
2. Giao diện người dùng (UI) hiển thị form nhập thông tin tháng và năm.
3. Người dùng nhập tháng, năm cần xuất, sau đó nhấn nút **Xuất**.
4. UI gửi thông tin nhập đến **Controller**.
5. **Controller** chuyển thông tin đến **Lớp nghiệp vụ (Business Logic Layer)**.
6. Lớp nghiệp vụ kiểm tra dữ liệu:
   * **Nếu không hợp lệ:** Trả về thông báo lỗi.
   * **Nếu hợp lệ:** Lấy danh sách hóa đơn từ **Database** qua dịch vụ.
7. Danh sách hóa đơn được trả về từ **Database**.
8. Lớp nghiệp vụ xử lý kết quả và gửi về **Controller**.
9. **Controller** cập nhật UI để hiển thị danh sách hóa đơn hoặc xuất file (PDF/Excel).

**Sơ đồ tuần tự:** 

*Hình 2.4.10: Sơ đồ tuần tự chức năng Xuất hóa đơn theo tháng.*

## 2.5. Sơ đồ Lớp (Class Diagram)

Sơ đồ lớp (Class Diagram) thể hiện cấu trúc hệ thống, bao gồm các lớp, thuộc tính, phương thức và mối quan hệ giữa các lớp. Trong phần này, chúng tôi trình bày sơ đồ lớp chi tiết cho từng chức năng chính trong hệ thống quản lý hóa đơn tiền điện.

### 2.5.1. Sơ đồ Lớp cho chức năng Đăng nhập

* **Mô tả chức năng:** Chức năng Đăng nhập cho phép người dùng nhập thông tin tài khoản và mật khẩu để xác thực danh tính. Hệ thống kiểm tra thông tin và phản hồi kết quả.
* **Các thành phần chính:**
  1. **LoginView:** Lớp giao diện người dùng, chịu trách nhiệm nhận thông tin đầu vào (username, password) và hiển thị kết quả.
  2. **LoginController:** Xử lý yêu cầu từ người dùng và kết nối với tầng nghiệp vụ.
  3. **LoginUseCase:** Thực hiện logic nghiệp vụ, kiểm tra thông tin đăng nhập.
  4. **LoginPresenter:** Định dạng dữ liệu từ tầng nghiệp vụ để hiển thị.
  5. **LoginDAOMySQL:** Tương tác với cơ sở dữ liệu MySQL, kiểm tra thông tin tài khoản.
  6. **LoginInputDTO và LoginOutputDTO:** Đóng gói thông tin đầu vào và đầu ra.
* **Mối quan hệ giữa các lớp:**
  1. **LoginView** kết nối với **LoginController** để gửi yêu cầu.
  2. **LoginController** gửi dữ liệu đến **LoginUseCase** qua **LoginInputDTO**.
  3. **LoginUseCase** gọi đến **LoginDAOMySQL** để kiểm tra thông tin tài khoản.
  4. Kết quả được trả về **LoginPresenter** qua **LoginOutputDTO** và hiển thị trên **LoginView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.1: Sơ đồ Lớp cho chức năng Đăng nhập*

### 2.5.2. Sơ đồ Lớp cho chức năng Đăng ký

* **Mô tả chức năng:**  
  Chức năng Đăng ký cho phép người dùng tạo tài khoản mới trên hệ thống. Người dùng nhập thông tin tài khoản, mật khẩu và hệ thống kiểm tra tính hợp lệ, lưu dữ liệu vào cơ sở dữ liệu nếu thành công.
* **Các thành phần chính:**

1. **RegisterView**: Lớp giao diện người dùng, nhận thông tin đăng ký từ người dùng và hiển thị kết quả.
2. **RegisterController**: Xử lý yêu cầu từ RegisterView và gọi tầng nghiệp vụ (Use Case) để thực hiện logic đăng ký.
3. **RegisterUseCase**: Thực hiện các quy tắc nghiệp vụ, kiểm tra dữ liệu đăng ký và tương tác với cơ sở dữ liệu.
4. **RegisterPresenter**: Định dạng dữ liệu từ tầng nghiệp vụ để hiển thị trên giao diện.
5. **RegisterDAOMySQL**: Tầng truy xuất dữ liệu, lưu thông tin tài khoản mới vào cơ sở dữ liệu MySQL.
6. **RegisterInputDTO** và **RegisterOutputDTO**: Đóng gói thông tin đầu vào và đầu ra.
7. **RegisterDatabaseBoundary**: Định nghĩa phương thức giao tiếp giữa Use Case và tầng dữ liệu.

* **Mối quan hệ giữa các lớp:**

1. **RegisterView** gửi yêu cầu đăng ký đến **RegisterController**.
2. **RegisterController** truyền dữ liệu đến **RegisterUseCase** thông qua **RegisterInputDTO**.
3. **RegisterUseCase** gọi đến **RegisterDAOMySQL** để kiểm tra và lưu tài khoản mới.
4. Kết quả trả về từ **RegisterDAOMySQL** được định dạng qua **RegisterPresenter** và gửi đến **RegisterView** qua **RegisterOutputDTO**.

A black and white screen with many white squares

Description automatically generated with medium confidence

*Hình 2.5.2: Sơ đồ Lớp cho chức năng Đăng ký*

### 2.5.3. Sơ đồ Lớp cho chức năng Quên mật khẩu

* **Mô tả chức năng:**  
  Chức năng Quên mật khẩu hỗ trợ người dùng đặt lại mật khẩu khi quên. Người dùng nhập tên tài khoản và email, hệ thống kiểm tra thông tin và cung cấp mật khẩu mới hoặc thông báo lỗi nếu thông tin không hợp lệ.
* **Các thành phần chính:**

1. **ForgotView**: Lớp giao diện người dùng, nhận thông tin đầu vào (username, email) và hiển thị kết quả (thành công/thất bại).
2. **ForgotController**: Xử lý yêu cầu từ ForgotView, gửi dữ liệu đến tầng nghiệp vụ (Use Case).
3. **ForgotUseCase**: Thực hiện logic nghiệp vụ, kiểm tra thông tin tài khoản và cập nhật mật khẩu mới nếu thông tin hợp lệ.
4. **ForgotPresenter**: Định dạng dữ liệu từ tầng nghiệp vụ để hiển thị trên giao diện.
5. **ForgotDAOMySQL**: Tương tác với cơ sở dữ liệu MySQL, kiểm tra và cập nhật thông tin tài khoản.
6. **ForgotInputDTO** và **ForgotOutputDTO**: Đóng gói thông tin đầu vào và đầu ra.
7. **ForgotDatabaseBoundary**: Định nghĩa phương thức giao tiếp giữa Use Case và tầng dữ liệu.

* **Mối quan hệ giữa các lớp:**

1. **ForgotView** gửi yêu cầu đặt lại mật khẩu đến **ForgotController**.
2. **ForgotController** truyền thông tin đến **ForgotUseCase** thông qua **ForgotInputDTO**.
3. **ForgotUseCase** gọi đến **ForgotDAOMySQL** để kiểm tra thông tin tài khoản và cập nhật mật khẩu.
4. Kết quả được định dạng qua **ForgotPresenter** và gửi đến **ForgotView** qua **ForgotOutputDTO**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.3: Sơ đồ Lớp cho chức năng Quên mật khẩu*

### 2.5.4. Sơ đồ Lớp tổng quan cho ứng dụng Quản lý hóa đơn tiền điện

* **Mô tả chức năng tổng quan:**  
  Ứng dụng Quản lý hóa đơn tiền điện bao gồm các chức năng chính như đăng nhập, thêm hóa đơn, sửa hóa đơn, xóa hóa đơn, tìm kiếm hóa đơn, xuất hóa đơn theo tháng, và tính toán trung bình thành tiền của khách hàng nước ngoài. Sơ đồ lớp dưới đây thể hiện cấu trúc các lớp và mối quan hệ giữa chúng trong toàn bộ hệ thống.
* **Các thành phần chính:**

1. **QuanLyHDTienDienView:** Lớp giao diện tổng quát, chịu trách nhiệm tương tác với người dùng, nhận yêu cầu và hiển thị kết quả.
2. **QuanLyHDTienDienController:** Điều phối các yêu cầu từ giao diện và chuyển dữ liệu đến tầng nghiệp vụ.
3. **QuanLyHDTienDienUseCase:** Tầng nghiệp vụ thực hiện các logic xử lý, tích hợp các chức năng cụ thể như thêm, sửa, xóa, tìm kiếm hóa đơn, và tính toán.
4. **QuanLyHDTienDienPresenter:** Định dạng dữ liệu từ tầng nghiệp vụ để trả về giao diện.
5. **QuanLyHDTienDienDAO:** Tầng tương tác với cơ sở dữ liệu để thực hiện các thao tác như truy xuất, thêm mới, sửa và xóa dữ liệu.
6. **QuanLyHDTienDienInputDTO và QuanLyHDTienDienOutputDTO:** Đóng gói dữ liệu đầu vào và đầu ra giữa các tầng.
7. **Các Interface Input/Output Boundary:** Kết nối linh hoạt giữa các thành phần với chức năng cụ thể như thêm, sửa, xóa, tìm kiếm, và tính toán hóa đơn.

* **Mối quan hệ giữa các lớp:**

1. **QuanLyHDTienDienView** gửi yêu cầu đến **QuanLyHDTienDienController** thông qua sự kiện giao diện.
2. **QuanLyHDTienDienController** truyền dữ liệu đến **QuanLyHDTienDienUseCase** qua **QuanLyHDTienDienInputDTO**.
3. **QuanLyHDTienDienUseCase** thực hiện xử lý logic và gọi các lớp DAO như **AddInvoiceDAO**, **EditInvoiceDAO**, hoặc **FindInvoiceDAO** để thao tác với cơ sở dữ liệu.
4. Kết quả từ DAO được trả về thông qua **QuanLyHDTienDienOutputDTO** và định dạng qua **QuanLyHDTienDienPresenter** để hiển thị trên **QuanLyHDTienDienView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.4: Sơ đồ Lớp tổng quan cho ứng dụng Quản lý hóa đơn tiền điện*

### 2.5.5. Sơ đồ Lớp cho chức năng Thêm hóa đơn

* **Mô tả chức năng:**  
  Chức năng Thêm hóa đơn cho phép người dùng nhập thông tin về hóa đơn tiền điện như tên khách hàng, ngày lập hóa đơn, số lượng điện tiêu thụ, và các thông tin liên quan. Hệ thống kiểm tra tính hợp lệ và lưu thông tin vào cơ sở dữ liệu nếu hợp lệ.
* **Các thành phần chính:**

1. **AddInvoiceView:**
   * Lớp giao diện người dùng.
   * Hiển thị biểu mẫu để nhập thông tin hóa đơn.
   * Gửi dữ liệu đến Controller sau khi người dùng hoàn thành.
2. **AddInvoiceController:**
   * Xử lý yêu cầu từ View.
   * Kết nối với tầng nghiệp vụ (Use Case) để thực hiện xử lý logic.
3. **AddInvoiceUseCase:**
   * Chịu trách nhiệm kiểm tra tính hợp lệ của dữ liệu.
   * Thực hiện các thao tác logic nghiệp vụ cần thiết, bao gồm phân loại hóa đơn (khách hàng cá nhân hoặc nước ngoài).
4. **AddInvoicePresenter:**
   * Định dạng dữ liệu kết quả từ tầng nghiệp vụ để hiển thị trên View.
5. **AddInvoiceDAOMySQL:**
   * Tầng truy vấn cơ sở dữ liệu MySQL.
   * Thực hiện lưu thông tin hóa đơn vào bảng dữ liệu tương ứng.
6. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Định nghĩa các thuộc tính và phương thức cơ bản của một hóa đơn.
   * **InvoiceVN (Lớp con):** Xử lý hóa đơn dành cho khách hàng Việt Nam.
   * **InvoiceNuocNgoai (Lớp con):** Xử lý hóa đơn dành cho khách hàng nước ngoài.
7. **DTOs (Data Transfer Objects):**
   * **AddInvoiceInputDTO:** Đóng gói thông tin đầu vào từ người dùng.
   * **AddInvoiceOutputDTO:** Đóng gói kết quả trả về sau khi thêm hóa đơn.

* **Mối quan hệ giữa các lớp:**

1. **AddInvoiceView** gửi yêu cầu thêm hóa đơn qua **AddInvoiceController**.
2. **AddInvoiceController** chuyển dữ liệu đến **AddInvoiceUseCase** thông qua **AddInvoiceInputDTO**.
3. **AddInvoiceUseCase** xử lý dữ liệu, kiểm tra loại hóa đơn (VN hoặc nước ngoài), và gọi **AddInvoiceDAOMySQL** để lưu dữ liệu.
4. Kết quả được đóng gói trong **AddInvoiceOutputDTO** và gửi qua **AddInvoicePresenter** để định dạng dữ liệu trả về.
5. Kết quả cuối cùng được hiển thị trên **AddInvoiceView**.

A screenshot of a computer

Description automatically generated

*Hình 2.5.5: Sơ đồ Lớp cho chức năng Thêm hóa đơn*

### 2.5.6. Sơ đồ Lớp cho chức năng Xóa hóa đơn

* **Mô tả chức năng:**  
  Chức năng Xóa hóa đơn cho phép người dùng xóa một hóa đơn tiền điện cụ thể khỏi hệ thống. Hệ thống kiểm tra tính hợp lệ của yêu cầu, xác nhận hóa đơn tồn tại trước khi thực hiện thao tác xóa.
* **Các thành phần chính:**

1. **DeleteInvoiceView:**
   * Lớp giao diện người dùng.
   * Hiển thị danh sách hóa đơn để người dùng chọn hóa đơn cần xóa.
   * Gửi yêu cầu xóa đến Controller.
2. **DeleteInvoiceController:**
   * Xử lý yêu cầu từ View.
   * Kết nối với tầng nghiệp vụ (Use Case) để thực hiện thao tác xóa.
3. **DeleteInvoiceUseCase:**
   * Xử lý logic nghiệp vụ.
   * Kiểm tra xem hóa đơn có tồn tại trong cơ sở dữ liệu hay không.
   * Thực hiện xóa nếu hợp lệ.
4. **DeleteInvoicePresenter:**
   * Định dạng dữ liệu kết quả từ tầng nghiệp vụ để hiển thị trên View.
5. **DeleteInvoiceDAOMySQL:**
   * Tầng truy vấn cơ sở dữ liệu MySQL.
   * Thực hiện thao tác xóa hóa đơn trong cơ sở dữ liệu.
6. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Định nghĩa các thuộc tính và phương thức cơ bản của một hóa đơn.
   * **InvoiceVN (Lớp con):** Xử lý hóa đơn dành cho khách hàng Việt Nam.
   * **InvoiceNuocNgoai (Lớp con):** Xử lý hóa đơn dành cho khách hàng nước ngoài.
7. **DTOs (Data Transfer Objects):**
   * **DeleteInvoiceInputDTO:** Đóng gói thông tin đầu vào của yêu cầu xóa.
   * **DeleteInvoiceOutputDTO:** Đóng gói kết quả trả về sau khi xóa hóa đơn.

* **Mối quan hệ giữa các lớp:**

1. **DeleteInvoiceView** gửi yêu cầu xóa hóa đơn qua **DeleteInvoiceController**.
2. **DeleteInvoiceController** chuyển dữ liệu đến **DeleteInvoiceUseCase** thông qua **DeleteInvoiceInputDTO**.
3. **DeleteInvoiceUseCase** kiểm tra tính hợp lệ và gọi **DeleteInvoiceDAOMySQL** để thực hiện thao tác xóa.
4. Kết quả được đóng gói trong **DeleteInvoiceOutputDTO** và gửi qua **DeleteInvoicePresenter** để định dạng dữ liệu trả về.
5. Kết quả cuối cùng được hiển thị trên **DeleteInvoiceView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.6: Sơ đồ Lớp cho chức năng Xóa hóa đơn*

### 2.5.7. Sơ đồ Lớp cho chức năng Sửa hóa đơn

* **Mô tả chức năng:**  
  Chức năng Sửa hóa đơn cho phép người dùng chỉnh sửa thông tin của một hóa đơn đã tồn tại. Hệ thống đảm bảo tính hợp lệ của dữ liệu trước khi cập nhật cơ sở dữ liệu.
* **Các thành phần chính:**

1. **EditInvoiceView:**
   * Lớp giao diện người dùng.
   * Hiển thị thông tin hóa đơn để chỉnh sửa.
   * Gửi yêu cầu cập nhật đến Controller.
2. **EditInvoiceController:**
   * Xử lý yêu cầu từ View.
   * Gửi thông tin cần chỉnh sửa đến tầng nghiệp vụ (Use Case).
3. **EditInvoiceUseCase:**
   * Xử lý logic nghiệp vụ.
   * Kiểm tra thông tin mới trước khi cập nhật vào cơ sở dữ liệu.
4. **EditInvoicePresenter:**
   * Định dạng dữ liệu kết quả từ tầng nghiệp vụ để hiển thị trên View.
5. **EditInvoiceDAOMySQL:**
   * Tầng truy vấn cơ sở dữ liệu MySQL.
   * Thực hiện thao tác chỉnh sửa hóa đơn trong cơ sở dữ liệu.
6. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Định nghĩa các thuộc tính và phương thức cơ bản của một hóa đơn.
   * **InvoiceVN (Lớp con):** Xử lý hóa đơn dành cho khách hàng Việt Nam.
   * **InvoiceNuocNgoai (Lớp con):** Xử lý hóa đơn dành cho khách hàng nước ngoài.
7. **DTOs (Data Transfer Objects):**
   * **EditInvoiceInputDTO:** Đóng gói thông tin đầu vào của yêu cầu sửa.
   * **EditInvoiceOutputDTO:** Đóng gói kết quả trả về sau khi sửa hóa đơn.

* **Mối quan hệ giữa các lớp:**

1. **EditInvoiceView** gửi yêu cầu sửa hóa đơn qua **EditInvoiceController**.
2. **EditInvoiceController** chuyển dữ liệu đến **EditInvoiceUseCase** thông qua **EditInvoiceInputDTO**.
3. **EditInvoiceUseCase** kiểm tra tính hợp lệ và gọi **EditInvoiceDAOMySQL** để thực hiện thao tác cập nhật.
4. Kết quả được đóng gói trong **EditInvoiceOutputDTO** và gửi qua **EditInvoicePresenter** để định dạng dữ liệu trả về.
5. Kết quả cuối cùng được hiển thị trên **EditInvoiceView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.7: Sơ đồ Lớp cho chức năng Sửa hóa đơn*

### 2.5.8. Sơ đồ Lớp cho chức năng Xuất toàn bộ hóa đơn

* **Mô tả chức năng:**  
  Chức năng Xuất toàn bộ hóa đơn cho phép người dùng xem danh sách tất cả các hóa đơn đã lưu trữ trong hệ thống. Dữ liệu được lấy từ cơ sở dữ liệu và hiển thị trên giao diện người dùng.
* **Các thành phần chính:**

1. **GetListInvoiceView:**
   * Lớp giao diện người dùng.
   * Hiển thị danh sách hóa đơn.
2. **GetListInvoiceController:**
   * Xử lý yêu cầu từ View.
   * Chuyển dữ liệu cần lấy về từ tầng nghiệp vụ.
3. **GetListInvoiceUseCase:**
   * Xử lý logic nghiệp vụ.
   * Gửi yêu cầu lấy toàn bộ hóa đơn từ cơ sở dữ liệu.
4. **GetListInvoicePresenter:**
   * Định dạng dữ liệu để hiển thị trên View.
5. **GetListInvoiceDAOMySQL:**
   * Tầng truy vấn cơ sở dữ liệu.
   * Thực hiện truy vấn để lấy danh sách hóa đơn.
6. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Lớp cha cho các loại hóa đơn.
   * **InvoiceVN (Lớp con):** Xử lý hóa đơn cho khách hàng Việt Nam.
   * **InvoiceNuocNgoai (Lớp con):** Xử lý hóa đơn cho khách hàng nước ngoài.
7. **DTOs (Data Transfer Objects):**
   * **GetListInvoiceInputDTO:** Đóng gói thông tin yêu cầu lấy dữ liệu.
   * **GetListInvoiceOutputDTO:** Đóng gói kết quả trả về sau khi truy vấn.

* **Mối quan hệ giữa các lớp:**

1. **GetListInvoiceView** gửi yêu cầu đến **GetListInvoiceController**.
2. **GetListInvoiceController** chuyển yêu cầu đến **GetListInvoiceUseCase** thông qua **GetListInvoiceInputDTO**.
3. **GetListInvoiceUseCase** gửi yêu cầu đến **GetListInvoiceDAOMySQL** để lấy dữ liệu.
4. Dữ liệu được trả về dưới dạng danh sách hóa đơn (**List<Invoice>**).
5. Kết quả được định dạng trong **GetListInvoicePresenter** qua **GetListInvoiceOutputDTO**.
6. Kết quả cuối cùng được hiển thị trên **GetListInvoiceView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.8: Sơ đồ Lớp cho chức năng Xuất toàn bộ hóa đơn*

### 2.5.9. Sơ đồ Lớp cho chức năng Tìm kiếm hóa đơn

* **Mô tả chức năng:**  
  Chức năng Tìm kiếm hóa đơn cho phép người dùng nhập thông tin liên quan (như mã khách hàng, ngày tháng) để tìm kiếm và hiển thị hóa đơn phù hợp từ hệ thống.
* **Các thành phần chính:**

1. **FindInvoiceView:**
   * Lớp giao diện người dùng.
   * Nhận thông tin tìm kiếm từ người dùng và hiển thị kết quả hóa đơn.
2. **FindInvoiceController:**
   * Xử lý yêu cầu từ View.
   * Chuyển dữ liệu cần tìm kiếm đến tầng nghiệp vụ.
3. **FindInvoiceUseCase:**
   * Xử lý logic nghiệp vụ.
   * Kiểm tra điều kiện tìm kiếm và gửi yêu cầu truy vấn đến cơ sở dữ liệu.
4. **FindInvoicePresenter:**
   * Định dạng dữ liệu để hiển thị trên View.
5. **FindInvoiceDAOMySQL:**
   * Tầng truy vấn cơ sở dữ liệu.
   * Thực hiện truy vấn để tìm hóa đơn theo tiêu chí.
6. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Lớp cha cho các loại hóa đơn.
   * **InvoiceVN (Lớp con):** Xử lý hóa đơn cho khách hàng Việt Nam.
   * **InvoiceNuocNgoai (Lớp con):** Xử lý hóa đơn cho khách hàng nước ngoài.
7. **DTOs (Data Transfer Objects):**
   * **FindInvoiceInputDTO:** Đóng gói thông tin đầu vào cho quá trình tìm kiếm.
   * **FindInvoiceOutputDTO:** Đóng gói kết quả trả về sau khi tìm kiếm

* **Mối quan hệ giữa các lớp:**

1. **FindInvoiceView** gửi yêu cầu đến **FindInvoiceController** với dữ liệu từ **FindInvoiceInputDTO**.
2. **FindInvoiceController** chuyển yêu cầu đến **FindInvoiceUseCase**.
3. **FindInvoiceUseCase** gửi yêu cầu đến **FindInvoiceDAOMySQL** để thực hiện tìm kiếm trong cơ sở dữ liệu.
4. Dữ liệu hóa đơn phù hợp được trả về dưới dạng danh sách hóa đơn (**List<Invoice>**).
5. Kết quả được định dạng trong **FindInvoicePresenter** qua **FindInvoiceOutputDTO**.
6. Kết quả cuối cùng được hiển thị trên **FindInvoiceView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.9: Sơ đồ Lớp cho chức năng Tìm kiếm hóa đơn*

### 2.5.10. Sơ đồ Lớp cho chức năng Tính trung bình thành tiền khách hàng nước ngoài

* **Mô tả chức năng:**  
  Chức năng tính trung bình thành tiền cho khách hàng nước ngoài thực hiện việc lấy danh sách các hóa đơn liên quan đến khách hàng nước ngoài, tính toán giá trị trung bình của tổng tiền trong các hóa đơn này và hiển thị kết quả cho người dùng.
* **Các thành phần chính:**

1. **AvgMoneyInvoiceNuocNgoaiView:**
   * Lớp giao diện người dùng.
   * Hiển thị thông tin trung bình thành tiền và nhận yêu cầu từ người dùng.
2. **AvgMoneyInvoiceNuocNgoaiController:**
   * Xử lý yêu cầu từ View.
   * Chuyển dữ liệu cần xử lý đến tầng nghiệp vụ.
3. **AvgMoneyInvoiceNuocNgoaiUseCase:**
   * Thực hiện logic nghiệp vụ.
   * Gọi các phương thức cần thiết từ tầng cơ sở dữ liệu và thực hiện tính toán giá trị trung bình.
4. **AvgMoneyInvoiceNuocNgoaiPresenter:**
   * Định dạng dữ liệu để hiển thị trên View.
5. **AvgMoneyInvoiceNuocNgoaiDAOMySQL:**
   * Thực hiện truy vấn cơ sở dữ liệu để lấy danh sách hóa đơn liên quan đến khách hàng nước ngoài.
6. **DTOs (Data Transfer Objects):**
   * **AvgMoneyInvoiceNuocNgoaiInputDTO:** Đóng gói thông tin đầu vào.
   * **AvgMoneyInvoiceNuocNgoaiOutputDTO:** Đóng gói kết quả trả về.
7. **TinhToanInvoice Class:**
   * Chứa các phương thức thực hiện tính toán (tổng tiền, trung bình thành tiền).
8. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Lớp cha cho các loại hóa đơn.
   * **InvoiceNuocNgoai (Lớp con):** Chuyên xử lý thông tin hóa đơn của khách hàng nước ngoài.

* **Mối quan hệ giữa các lớp:**

1. **AvgMoneyInvoiceNuocNgoaiView** gửi yêu cầu đến **AvgMoneyInvoiceNuocNgoaiController**.
2. **AvgMoneyInvoiceNuocNgoaiController** chuyển thông tin đầu vào đến **AvgMoneyInvoiceNuocNgoaiUseCase** qua **AvgMoneyInvoiceNuocNgoaiInputDTO**.
3. **AvgMoneyInvoiceNuocNgoaiUseCase** sử dụng **AvgMoneyInvoiceNuocNgoaiDAOMySQL** để lấy danh sách hóa đơn của khách hàng nước ngoài.
4. Dữ liệu được tính toán trung bình thông qua lớp **TinhToanInvoice**.
5. Kết quả được định dạng qua **AvgMoneyInvoiceNuocNgoaiPresenter** và trả về **AvgMoneyInvoiceNuocNgoaiOutputDTO**.
6. Kết quả cuối cùng hiển thị trên **AvgMoneyInvoiceNuocNgoaiView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.10: Sơ đồ Lớp cho chức năng Tính trung bình thành tiền khách hàng nước ngoài*

### 2.5.11. Sơ đồ Lớp cho chức năng Tổng số khách hàng từng loại của hóa đơn tiền điện

* **Mô tả chức năng:**  
  Chức năng tính tổng số khách hàng theo từng loại hóa đơn tiền điện thực hiện việc lấy danh sách hóa đơn từ cơ sở dữ liệu, phân loại khách hàng theo loại hóa đơn (Sinh hoạt, Kinh doanh, Sản xuất), và tính tổng số khách hàng cho từng loại.
* **Các thành phần chính:**

1. **SumKHInvoiceView:**
   * Lớp giao diện người dùng.
   * Hiển thị thông tin tổng số khách hàng theo loại hóa đơn.
2. **SumKHInvoiceController:**
   * Nhận yêu cầu từ View và chuyển tiếp dữ liệu cần xử lý đến tầng nghiệp vụ.
3. **SumKHInvoiceUseCase:**
   * Xử lý logic nghiệp vụ: lấy danh sách hóa đơn, phân loại và tính tổng số khách hàng.
4. **SumKHInvoicePresenter:**
   * Định dạng dữ liệu để hiển thị trên View.
5. **SumKHInvoiceDAOMySQL:**
   * Tương tác với cơ sở dữ liệu để truy xuất danh sách hóa đơn.
6. **DTOs (Data Transfer Objects):**
   * **SumKHInvoiceInputDTO:** Đóng gói thông tin đầu vào.
   * **SumKHInvoiceOutputDTO:** Đóng gói kết quả trả về.
7. **TinhToanInvoice Class:**
   * Chứa các phương thức hỗ trợ tính toán và xử lý danh sách hóa đơn.
8. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Lớp cha cho các loại hóa đơn.
   * **InvoiceVN và InvoiceNuocNgoai:** Các lớp con quản lý thông tin hóa đơn Việt Nam và nước ngoài.

* **Mối quan hệ giữa các lớp:**

1. **SumKHInvoiceView** gửi yêu cầu đến **SumKHInvoiceController**.
2. **SumKHInvoiceController** chuyển thông tin đầu vào đến **SumKHInvoiceUseCase** qua **SumKHInvoiceInputDTO**.
3. **SumKHInvoiceUseCase** gọi đến **SumKHInvoiceDAOMySQL** để truy xuất danh sách hóa đơn từ cơ sở dữ liệu.
4. Dữ liệu được xử lý và phân loại thông qua lớp **TinhToanInvoice**.
5. Kết quả được định dạng qua **SumKHInvoicePresenter** và trả về **SumKHInvoiceOutputDTO**.
6. Kết quả cuối cùng hiển thị trên **SumKHInvoiceView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.11: Sơ đồ Lớp cho chức năng Tổng số khách hàng từng loại của hóa đơn tiền điện*

### 2.5.12. Sơ đồ Lớp cho chức năng Xuất hóa đơn tiền điện theo tháng

* **Mô tả chức năng:**  
  Chức năng xuất hóa đơn tiền điện theo tháng thực hiện việc truy xuất danh sách hóa đơn từ cơ sở dữ liệu, dựa trên tháng được chỉ định và hiển thị kết quả dưới dạng danh sách hóa đơn phù hợp.
* **Các thành phần chính:**

1. **ExportInvoiceByMonthView:**
   * Lớp giao diện người dùng.
   * Hiển thị thông tin hóa đơn theo tháng.
2. **ExportInvoiceByMonthController:**
   * Nhận yêu cầu từ View và chuyển tiếp dữ liệu cần xử lý đến tầng nghiệp vụ.
3. **ExportInvoiceByMonthUseCase:**
   * Thực hiện logic nghiệp vụ: truy vấn hóa đơn theo tháng từ cơ sở dữ liệu.
4. **ExportInvoiceByMonthPresenter:**
   * Định dạng dữ liệu để hiển thị trên View.
5. **ExportInvoiceByMonthDAOMySQL:**
   * Tương tác với cơ sở dữ liệu để truy xuất danh sách hóa đơn theo tháng.
6. **DTOs (Data Transfer Objects):**
   * **ExportInvoiceByMonthInputDTO:** Đóng gói thông tin đầu vào (tháng).
   * **ExportInvoiceByMonthOutputDTO:** Đóng gói kết quả trả về (danh sách hóa đơn).
7. **Invoice Class Hierarchy:**
   * **Invoice (Lớp trừu tượng):** Lớp cha quản lý thông tin hóa đơn.
   * **InvoiceVN và InvoiceNuocNgoai:** Các lớp con quản lý thông tin hóa đơn Việt Nam và nước ngoài.

* **Mối quan hệ giữa các lớp:**

1. **ExportInvoiceByMonthView** gửi yêu cầu đến **ExportInvoiceByMonthController**.
2. **ExportInvoiceByMonthController** chuyển thông tin đầu vào đến **ExportInvoiceByMonthUseCase** qua **ExportInvoiceByMonthInputDTO**.
3. **ExportInvoiceByMonthUseCase** gọi đến **ExportInvoiceByMonthDAOMySQL** để truy xuất danh sách hóa đơn từ cơ sở dữ liệu dựa trên tháng chỉ định.
4. Kết quả được định dạng qua **ExportInvoiceByMonthPresenter** và trả về **ExportInvoiceByMonthOutputDTO**.
5. Kết quả cuối cùng hiển thị trên **ExportInvoiceByMonthView**.

A screenshot of a computer screen

Description automatically generated

*Hình 2.5.12: Sơ đồ Lớp cho chức năng Xuất hóa đơn tiền điện theo tháng*

# CHƯƠNG 3: KIỂM THỬ CHƯƠNG TRÌNH

# 3.1. Kiểm thử chương trình

### 3.1.1. Đăng nhập

A screenshot of a login form

Description automatically generated

*Hình 3.1.1.1: Giao diện đăng nhập*

*A screenshot of a login form

Description automatically generated*

*Hình 3.1.1.2: Giao diện đăng nhập báo lỗi*

### 3.1.2. Đăng ký

A screenshot of a computer screen

Description automatically generated

*Hình 3.1.2.1: Giao diện đăng ký*

A screenshot of a computer

Description automatically generated

*Hình 3.1.2.2: Giao diện đăng ký báo lỗi*

### 3.1.3. Quên mật khẩu

A screenshot of a computer

Description automatically generated

*Hình 3.1.3.1: Giao diện quên mật khẩu*

A screenshot of a computer

Description automatically generated

*Hình 3.1.3.2: Giao diện quên mật khẩu báo lỗi*

### 3.1.4. Quản lý hóa đơn tiền điện

A screenshot of a computer

Description automatically generated

*Hình 3.1.4.1: Giao diện quản lý hóa đơn quyền Admin*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.4.2: Giao diện quản lý hóa đơn quyền User*

### 3.1.5. Thêm hóa đơn tiền điện

A screenshot of a computer

Description automatically generated

*Hình 3.1.5.1: Giao diện thêm hóa đơn tiền điện*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.5.2: Giao diện thêm hóa đơn tiền điện báo lỗi*

### 3.1.6. Xóa hóa đơn tiền điện

A screenshot of a computer

Description automatically generated

*Hình 3.1.6.1: Giao diện Xóa hóa đơn tiền điện*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.6.2: Thông báo đả xóa thành công*

### 3.1.7. Sửa hóa đơn tiền điện

A screenshot of a computer

Description automatically generated

*Hình 3.1.7.1: Giao diện Sửa hóa đơn tiền điện*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.7.2: Thông báo đã sửa thành công*

### 3.1.8. Xuất hóa đơn tiền điện theo tháng

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.8.1: Giao diện xuất hóa đơn tiền điện theo tháng*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.8.2: Giao diện xuất hóa đơn tiền điện theo tháng báo lỗi*

*A screenshot of a computer

Description automatically generated*

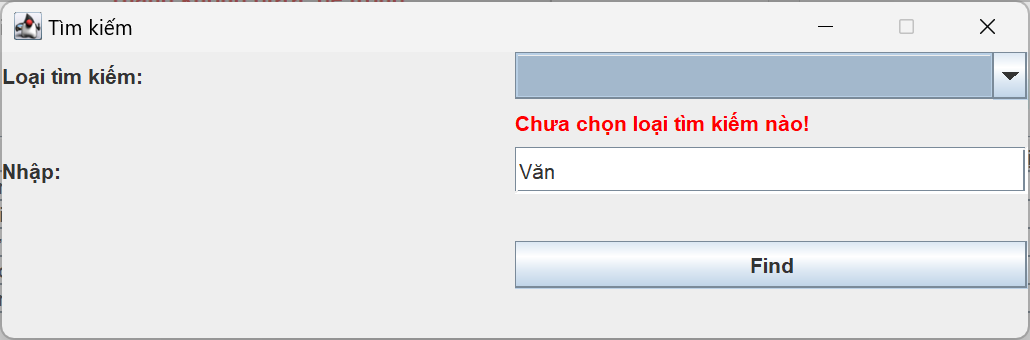
*Hình 3.1.8.3: Giao diện xuất hóa đơn tiền điện theo tháng thành công*

### 3.1.9. Tìm kiếm hóa đơn tiền điện

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.9.1: Giao diện Tìm kiếm hóa đơn tiền điện*

**

*Hình 3.1.9.2: Giao diện Tìm kiếm hóa đơn tiền điện báo lỗi*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.9.3: Giao diện Tìm kiếm hóa đơn tiền điện thành công*

### 3.1.10. Tính trung bình thành tiền nước ngoài

*A screenshot of a computer error

Description automatically generated*

*Hình 3.1.10.1: Giao diện Tính trung bình thành tiền nước ngoài*

### 3.1.11. Tính tổng khách hàng theo loại

A screenshot of a computer

Description automatically generated

*Hình 3.1.11.1: Giao diện Tính tổng khách hàng theo loại*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.11.2: Giao diện Tính tổng khách hàng theo loại báo lỗi*

*A screenshot of a computer

Description automatically generated*

*Hình 3.1.11.3: Giao diện Tính tổng khách hàng theo loại thành công*

# CHƯƠNG 4: TỔNG KẾT

## 4.1. Kết quả đạt được

Trong quá trình thực hiện đề tài "Xây dựng chương trình quản lý danh sách hóa đơn tiền điện của khách hàng", chúng tôi đã đạt được những kết quả sau:

* **Tổ chức và cấu trúc mã nguồn rõ ràng**: Mã nguồn của hệ thống được tổ chức một cách khoa học, chia thành các lớp và gói riêng biệt cho từng chức năng. Sự phân chia này giúp dễ dàng quản lý và bảo trì mã nguồn, đồng thời tạo điều kiện thuận lợi cho việc phát triển các tính năng mới mà không ảnh hưởng đến các chức năng hiện có.
* **Chức năng hoàn thiện**: Hệ thống đã hoàn thành đầy đủ các chức năng cần thiết cho việc quản lý hóa đơn, bao gồm thêm mới, sửa đổi, xóa, tìm kiếm và tính toán hóa đơn. Điều này cho phép người dùng thực hiện các thao tác một cách hiệu quả và chính xác, đáp ứng tốt các yêu cầu nghiệp vụ đặt ra.
* **Xử lý lỗi hiệu quả**: Hệ thống được trang bị cơ chế xử lý lỗi rõ ràng và thân thiện với người dùng. Khi có vấn đề xảy ra, hệ thống cung cấp thông báo chi tiết, giúp người dùng nhanh chóng nhận biết và khắc phục. Điều này nâng cao trải nghiệm người dùng và giảm thiểu sự nhầm lẫn trong quá trình sử dụng.
* **Khả năng mở rộng**: Thiết kế của hệ thống cho phép dễ dàng mở rộng trong tương lai. Các interface và lớp use case được xây dựng linh hoạt, có thể mở rộng để thêm các tính năng mới mà không cần thay đổi cấu trúc hiện tại. Điều này đảm bảo hệ thống có thể thích nghi với sự phát triển của công nghệ và đáp ứng nhu cầu ngày càng đa dạng của người dùng.
* **Tính nhất quán trong quản lý dữ liệu**: Hệ thống đảm bảo tính nhất quán và chính xác trong việc quản lý dữ liệu hóa đơn thông qua các lớp biên giới cơ sở dữ liệu (database boundary classes). Việc này giúp tránh các lỗi liên quan đến việc quản lý và truy xuất dữ liệu, đảm bảo thông tin luôn được cập nhật và đồng bộ.

## 4.2. Đánh giá ưu điểm và khuyết điểm

**Ưu điểm**

* **Cấu trúc tổ chức tốt**: Hệ thống quản lý hóa đơn được xây dựng với cấu trúc tổ chức rõ ràng và logic. Mỗi thành phần của hệ thống được phân chia hợp lý theo chức năng, tạo điều kiện thuận lợi cho việc quản lý, bảo trì và phát triển trong tương lai. Điều này giúp giảm thiểu thời gian và chi phí khi cần mở rộng hoặc cập nhật hệ thống.
* **Chức năng đầy đủ**: Hệ thống cung cấp đầy đủ các chức năng cần thiết cho việc quản lý hóa đơn, bao gồm thêm mới, sửa đổi, xóa và tìm kiếm hóa đơn. Ngoài ra, với sự hỗ trợ của các tính năng báo cáo và phân tích, người dùng có thể theo dõi tình hình tài chính và đưa ra quyết định kinh doanh chính xác hơn.
* **Khả năng mở rộng**: Thiết kế linh hoạt của hệ thống cho phép dễ dàng mở rộng để thêm các tính năng mới mà không ảnh hưởng đến các chức năng hiện có. Điều này rất quan trọng trong bối cảnh công nghệ liên tục thay đổi và nhu cầu người dùng ngày càng đa dạng. Hệ thống có thể được nâng cấp để tích hợp các công nghệ mới hoặc đáp ứng các yêu cầu nghiệp vụ phát sinh.

**Khuyết điểm**

* **Khả năng bảo mật chưa cao**: Nếu không có các biện pháp bảo mật thích hợp, hệ thống có thể trở thành mục tiêu của các cuộc tấn công mạng, dẫn đến rò rỉ thông tin nhạy cảm của người dùng. Bảo mật dữ liệu là yếu tố cực kỳ quan trọng, đặc biệt khi xử lý thông tin cá nhân và tài chính. Việc thiếu các biện pháp như mã hóa dữ liệu, xác thực hai yếu tố và chính sách bảo mật nghiêm ngặt có thể gây hậu quả nghiêm trọng, ảnh hưởng đến uy tín của tổ chức.
* **Tiêu tốn tài nguyên**: Nếu không được tối ưu hóa, hệ thống có thể tiêu tốn nhiều tài nguyên, ảnh hưởng đến hiệu suất, đặc biệt khi xử lý lượng lớn hóa đơn. Việc tiêu tốn tài nguyên không chỉ làm giảm tốc độ xử lý mà còn có thể dẫn đến tình trạng quá tải hệ thống, gây gián đoạn cho người dùng. Cần có các biện pháp giám sát và tối ưu hóa thường xuyên để duy trì hiệu suất hệ thống, đảm bảo hoạt động mượt mà ngay cả khi phải xử lý khối lượng công việc lớn.
* **Giao diện người dùng cần cải thiện**: Giao diện người dùng của hệ thống chưa được phát triển đầy đủ, điều này có thể gây khó khăn cho người dùng trong việc tương tác với phần mềm, đặc biệt là những người không quen thuộc với công nghệ. Một giao diện không thân thiện có thể dẫn đến nhầm lẫn trong quá trình sử dụng và giảm hiệu suất làm việc của người dùng. Việc cải thiện giao diện người dùng là cần thiết để nâng cao trải nghiệm và đảm bảo rằng họ có thể dễ dàng sử dụng tất cả các chức năng của hệ thống.

## 4.3. Hướng phát triển và mở rộng đề tài

* **Cải thiện giao diện người dùng**: Để nâng cao trải nghiệm người dùng, việc cải thiện giao diện là ưu tiên hàng đầu. Một giao diện thân thiện và trực quan sẽ giúp người dùng dễ dàng tương tác với hệ thống, tăng cường sự hài lòng và hiệu quả làm việc. Thiết kế giao diện với các biểu tượng rõ ràng và hướng dẫn sử dụng chi tiết sẽ giúp người dùng mới nhanh chóng làm quen. Bên cạnh đó, áp dụng **Responsive Design** sẽ đảm bảo giao diện hoạt động tốt trên nhiều thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động, giúp người dùng quản lý hóa đơn mọi lúc, mọi nơi.
* **Tích hợp công nghệ mới**: Việc tích hợp công nghệ mới vào hệ thống quản lý hóa đơn sẽ mang lại nhiều lợi ích. Phát triển phiên bản web của hệ thống sẽ cho phép người dùng truy cập từ bất kỳ đâu mà không cần cài đặt phần mềm, tăng tính linh hoạt và tiện dụng. Ngoài ra, xây dựng ứng dụng di động sẽ giúp người dùng dễ dàng quản lý hóa đơn ngay trên điện thoại, tạo điều kiện thuận lợi cho việc theo dõi và xử lý hóa đơn mọi lúc.
* **Tăng cường bảo mật**: Bảo mật thông tin là yếu tố quan trọng trong bất kỳ hệ thống nào, đặc biệt khi xử lý dữ liệu nhạy cảm như hóa đơn và thông tin khách hàng. Để tăng cường bảo mật, hệ thống có thể áp dụng **xác thực đa yếu tố**, giúp ngăn chặn truy cập trái phép vào tài khoản người dùng. Bên cạnh đó, việc **mã hóa dữ liệu** cũng là giải pháp hiệu quả để bảo vệ thông tin nhạy cảm, đảm bảo dữ liệu của khách hàng luôn được an toàn.
* **Mở rộng chức năng**: Hệ thống có thể được mở rộng với nhiều chức năng mới để đáp ứng nhu cầu ngày càng cao của người dùng. Một trong những chức năng quan trọng là cung cấp các **báo cáo chi tiết và phân tích dữ liệu**, giúp người dùng có cái nhìn sâu sắc hơn về tình hình tài chính và hoạt động kinh doanh. Thêm vào đó, **chức năng quản lý thông tin khách hàng** sẽ cho phép người dùng theo dõi lịch sử giao dịch và ghi chú liên quan, làm cho hệ thống trở nên toàn diện hơn.
* **Tối ưu hóa hiệu suất**: Tối ưu hóa hiệu suất của hệ thống là yếu tố quan trọng để đảm bảo người dùng có trải nghiệm tốt nhất. Cải thiện tốc độ truy xuất dữ liệu bằng cách **tối ưu hóa các truy vấn cơ sở dữ liệu** sẽ nâng cao hiệu suất và giảm thời gian chờ đợi. Ngoài ra, việc quản lý tài nguyên hiệu quả sẽ giúp giảm thiểu tiêu tốn tài nguyên hệ thống, đảm bảo hoạt động mượt mà ngay cả khi xử lý lượng lớn hóa đơn. Những cải tiến này không chỉ nâng cao trải nghiệm người dùng mà còn giúp hệ thống hoạt động ổn định và bền vững hơn.

**Kết luận**

Đề tài "Xây dựng chương trình quản lý danh sách hóa đơn tiền điện của khách hàng" đã hoàn thành các mục tiêu đề ra, xây dựng được một hệ thống với cấu trúc rõ ràng, chức năng đầy đủ và khả năng mở rộng cao. Mặc dù còn một số hạn chế cần khắc phục, nhưng với những định hướng phát triển và mở rộng đã đề cập, hệ thống có tiềm năng trở thành một công cụ quản lý hiệu quả, đáp ứng tốt nhu cầu của người dùng trong tương lai.

# TÀI LIỆU THAM KHẢO

1. Nguyễn, Q. T. (2019). *Phân tích và thiết kế hệ thống thông tin quản lý*. Nhà xuất bản Khoa học và Kỹ thuật.
2. Lê, T. T. (2020). *Lập trình Java cơ bản và nâng cao*. Nhà xuất bản Bách Khoa Hà Nội.
3. Nguyễn, V. A. (2021). *Cơ sở dữ liệu: Lý thuyết và thực hành*. Nhà xuất bản Giáo dục Việt Nam.
4. Martin, R. C. (2008). *Clean code: A handbook of agile software craftsmanship*. Prentice Hall.
5. Fowler, M. (2004). *UML distilled: A brief guide to the standard object modeling language* (3rd ed.). Addison-Wesley.
6. Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.
7. Bộ Tài chính. (2021). *Thông tư 78/2021/TT-BTC ngày 17/9/2021 hướng dẫn thực hiện một số điều của Luật Quản lý thuế số 38/2019/QH14 và Nghị định số 123/2020/NĐ-CP về hóa đơn, chứng từ*. Truy cập từ https://thuvienphapluat.vn/van-ban/Thue-Phi-Le-Phi/Thong-tu-78-2021-TT-BTC-hoa-don-dien-tu-493287.aspx
8. Bộ Tài chính. (2020). *Nghị định 123/2020/NĐ-CP ngày 19/10/2020 quy định về hóa đơn, chứng từ*. Truy cập từ https://thuvienphapluat.vn/van-ban/Thue-Phi-Le-Phi/Nghi-dinh-123-2020-ND-CP-hoa-don-chung-tu-453398.aspx