

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIA ĐỊNH
KHOA: CÔNG NGHỆ THÔNG TIN**



TIỂU LUẬN

**Xây dựng chương trình quản lý các giao
dịch nhà đất**

MÔN: MẪU THIẾT KẾ CHO PHẦN MỀM

Ngành: KỸ THUẬT PHẦN MỀM

Giảng viên hướng dẫn: LÊ HUỖNH PHƯỚC

Sinh viên thực hiện: NGUYỄN THÀNH HÙNG

MSSV: 22140068

Lớp: 221402

TP. Hồ Chí Minh, tháng 07 năm 2024

NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN
TIỂU LUẬN MÔN: Mẫu thiết kế cho phần mềm

- 1. **Họ và tên sinh viên:** Nguyễn Thành Hưng
- 2. **Tên đề tài:** Xây dựng chương trình quản lý danh sách các giao dịch nhà đất
- 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:.....

Điểm số: Điểm chữ:

TP. HCM, ngày ... tháng ... năm 20.....

Giảng viên chấm thi
(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn thầy Lê Huỳnh Phước, thầy đã tận tình hướng dẫn, chỉ bảo và luôn có sự phản hồi tỉ mỉ trong thời gian nhanh nhất nhằm giúp em trong suốt thời gian qua để có thể hoàn thành bài tiểu luận này.

Em cũng xin gửi lời cảm ơn đến toàn thể thầy cô trong Trường đại học Gia Định đã giảng dạy, và tạo điều kiện cho em trong quá trình học tập và nghiên cứu tại trường. Những kiến thức mà em nhận được sẽ là hành trang giúp em vững bước trong tương lai.

Cuối cùng em xin cảm ơn gia đình, bạn bè, người thân đã luôn ở bên cạnh và động viên là nguồn cổ vũ lớn lao, là động lực giúp em hoàn thành bài tiểu luận này.

Mặc dù đã cố gắng hoàn thành bài tiểu luận trong phạm vi và khả năng có thể. Tuy nhiên sẽ không tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và tận tình chỉ bảo của quý thầy cô và toàn thể các bạn.

Em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Em xin cam đoan đây là công trình nghiên cứu của cá nhân em và được sự hướng dẫn của thầy Lê Huỳnh Phước. Các nội dung nghiên cứu trong đề tài “Phân tích thực trạng phát triển “Kỹ năng mềm” của sinh viên khoa CNTT, Trường đại học Gia Định” của em là trung thực và chưa công bố dưới bất kì hình thức nào trước đây. Những số liệu trong bảng biểu phục vụ cho việc phân tích, đánh giá được cá nhân thu thập từ các nguồn khác nhau có ghi rõ nguồn gốc. Nếu phát hiện có bất kỳ gian lận nào em xin hoàn toàn chịu trách nhiệm về nội dung bài tiểu luận của mình.

TP. HCM, tháng 07 năm 2024

Sinh viên cam đoan

(Ký và ghi rõ họ tên)

MỤC LỤC

PHẦN MỞ ĐẦU	1
1. Lý do chọn đề tài	1
2. Giới hạn của đề tài	1
3. Bố cục đề tài.....	1
PHẦN NỘI DUNG	2
CHƯƠNG 1: LÝ THUYẾT VỀ MẪU THUYẾT KẾ	2
1.1. Mô hình MVC.....	2
1.1.1. Lý thuyết.....	2
1.1.2. Ưu nhược điểm của mô hình MVC	3
1.2. Mô hình 3 tầng Three Layer	3
1.2.1. Lý thuyết.....	3
1.2.2. Ưu nhược điểm của mô hình 3 tầng Three Layer	4
1.3. Mẫu Observer.....	5
1.3.1. Lý thuyết.....	5
1.3.2. Ưu nhược điểm của mẫu Observer	5
1.4. Mẫu Command Processor	6
1.4.1. Lý thuyết.....	6
1.4.2. Ưu nhược điểm của mẫu Command Processor	6
CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ PHẦN MỀM.....	7
2.1. Sơ đồ thiết kế 3 tầng Three Layer	7
2.2. Phân tích các chức năng của ứng dụng.....	8
2.2.1. Chức năng Thêm giao dịch	8
2.2.2. Chức năng Sửa giao dịch.....	8
2.2.3. Chức năng Xóa giao dịch	8
2.2.4. Chức năng Tìm kiếm giao dịch.....	8
2.2.5. Chức năng Làm mới.....	8
2.2.6. Chức năng Tổng tiền	9
2.3. Sơ đồ thiết kế cơ sở dữ liệu.....	9
CHƯƠNG 3: CÀI ĐẶT THỰC NGHIỆM VÀ KẾT QUẢ CHƯƠNG TRÌNH....	10
3.1. Giao diện hiển thị chính.....	10
3.2. Chức năng “Thêm”	10
3.3. Chức năng “Sửa”	11
3.4. Chức năng “Xóa”	12

3.5. Chức năng “Tìm Kiếm”	14
3.6. Chức năng “Làm Mới”	15
3.7. Chức năng “Tổng tiền”	17
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	18
4.1. Kết luận	18
4.2. Hướng phát triển	18
TÀI LIỆU THAM KHẢO	20

PHẦN MỞ ĐẦU

1. Lý do chọn đề tài

Với sự phát triển không ngừng của thị trường bất động sản, việc quản lý thông tin giao dịch nhà đất trở thành một yêu cầu cấp thiết. Thị trường này đang ngày càng phức tạp với số lượng lớn các giao dịch diễn ra hàng ngày, đòi hỏi một hệ thống quản lý hiệu quả để xử lý thông tin một cách nhanh chóng và chính xác. Một hệ thống quản lý giao dịch nhà đất hiệu quả sẽ giúp các doanh nghiệp bất động sản hoạt động trơn tru hơn, nâng cao tính minh bạch, giảm thiểu rủi ro sai sót và gian lận, đồng thời tăng cường sự tin tưởng của khách hàng.

Một chương trình quản lý giao dịch tự động hóa quy trình lưu trữ và xử lý dữ liệu, giúp giảm thiểu sai sót do con người gây ra và tiết kiệm thời gian so với việc quản lý thủ công. Việc tự động hóa không chỉ giúp xử lý khối lượng dữ liệu lớn một cách nhanh chóng mà còn hỗ trợ việc phân tích và báo cáo thông tin, từ đó giúp các nhà quản lý ra quyết định chính xác hơn.

Chương trình quản lý giao dịch nhà đất sẽ cung cấp các chức năng tìm kiếm và tra cứu thông tin một cách nhanh chóng và thuận tiện. Điều này giúp các bên liên quan dễ dàng truy xuất thông tin cần thiết, từ lịch sử giao dịch, tình trạng pháp lý của bất động sản, đến các thông tin chi tiết khác. Hơn nữa, hệ thống có thể tạo ra các báo cáo tùy chỉnh theo yêu cầu, hỗ trợ việc phân tích và đánh giá hiệu quả kinh doanh.

Từ những lý do trên, nên em quyết định chọn đề tài “Xây dựng chương trình quản lý giao dịch nhà đất”.

2. Giới hạn của đề tài

- Đề tài chỉ xoay quanh các chức năng: Thêm giao dịch mới, Sửa giao dịch, Xóa giao dịch, Tìm kiếm giao dịch thông qua mã giao dịch

3. Bố cục đề tài

- CHƯƠNG 1: Lý thuyết về mẫu thuyết kế
- CHƯƠNG 2: Phân tích, thiết kế phần mềm
- CHƯƠNG 3: Cài đặt thực nghiệm và kết quả chương trình
- CHƯƠNG 4: Kết luận và hướng phát triển

PHẦN NỘI DUNG

CHƯƠNG 1: LÝ THUYẾT VỀ MẪU THUYẾT KẾ

1.1. Mô hình MVC

1.1.1. Lý thuyết

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm được sử dụng phổ biến trong phát triển ứng dụng. MVC phân chia ứng dụng thành ba phần chính: Model, View, và Controller. Mỗi phần có nhiệm vụ và trách nhiệm riêng, giúp tổ chức mã nguồn một cách rõ ràng và dễ quản lý.

- *Model*

+ Nhiệm vụ: Chứa dữ liệu và logic nghiệp vụ của ứng dụng. Model là nơi xử lý các thao tác liên quan đến dữ liệu như truy vấn, cập nhật, xóa dữ liệu.

+ Ví dụ: Trong một ứng dụng quản lý học sinh, Model có thể là lớp Student chứa thông tin về học sinh như tên, tuổi, và lớp học, cùng với các phương thức để lưu và truy xuất thông tin này từ cơ sở dữ liệu.

- *View*

+ Nhiệm vụ: Hiển thị dữ liệu cho người dùng và nhận các tương tác từ người dùng. View chỉ phụ trách việc hiển thị và không chứa logic nghiệp vụ.

+ Ví dụ: Một form HTML hiển thị danh sách học sinh và cho phép người dùng thêm học sinh mới là một ví dụ về View.

- *Controller*

+ Nhiệm vụ: Nhận yêu cầu từ người dùng, gọi Model để xử lý dữ liệu, và chọn View phù hợp để hiển thị kết quả cho người dùng. Controller kết nối Model và View với nhau.

+ Ví dụ: Một phương thức trong Controller nhận yêu cầu thêm học sinh mới, gọi Model để lưu thông tin học sinh và sau đó chuyển hướng người dùng tới trang hiển thị danh sách học sinh.

1.1.2. Ưu nhược điểm của mô hình MVC

- Ưu điểm:

- + Phân chia rõ ràng trách nhiệm: Mỗi thành phần có nhiệm vụ riêng biệt, giúp mã nguồn dễ đọc, dễ bảo trì và nâng cấp.

- + Tái sử dụng mã: View và Controller có thể được tái sử dụng cho nhiều Model khác nhau và ngược lại.

- + Dễ dàng bảo trì và phát triển: Việc thay đổi giao diện (View) không ảnh hưởng đến logic nghiệp vụ (Model) và ngược lại. Điều này giúp việc bảo trì và phát triển dễ dàng hơn.

- + Hỗ trợ làm việc nhóm: Các lập trình viên có thể làm việc độc lập trên các thành phần khác nhau mà không ảnh hưởng lẫn nhau.

- Nhược điểm:

- + Độ phức tạp ban đầu: Đối với các ứng dụng nhỏ hoặc đơn giản, việc áp dụng MVC có thể gây ra sự phức tạp không cần thiết.

- + Yêu cầu kiến thức và kỹ năng: Lập trình viên cần hiểu rõ về mô hình MVC và cách thức triển khai nó một cách hiệu quả.

- + Khó khăn trong việc quản lý dự án lớn: Khi dự án lớn, việc tổ chức và duy trì mã nguồn theo mô hình MVC có thể trở nên phức tạp và đòi hỏi sự quản lý chặt chẽ.

1.2. Mô hình 3 tầng Three Layer

1.2.1. Lý thuyết

Mô hình 3 tầng (Three Layer) là một kiến trúc phần mềm phân chia ứng dụng thành ba tầng riêng biệt: Presentation (Tầng trình bày), Domain (Tầng logic), và Persistence (Tầng dữ liệu). Cấu trúc này giúp tối ưu hóa khả năng mở rộng, bảo trì và quản lý ứng dụng. Mỗi tầng có nhiệm vụ và trách nhiệm riêng, đảm bảo rằng các thay đổi trong một tầng không ảnh hưởng đến các tầng khác.

- Presentation

- + Nhiệm vụ: Hiển thị dữ liệu và giao diện người dùng, nhận các tương tác từ người dùng. Tầng này chứa các thành phần giao diện người dùng (UI).

- + Ví dụ: Trong ứng dụng quản lý giao dịch nhà đất thì tầng này dùng để hiển thị dữ liệu và nhận input từ người dùng.

- *Domain*

+ Nhiệm vụ: Xử lý logic nghiệp vụ của ứng dụng. Tầng này chứa các quy tắc, quy trình, và logic nghiệp vụ cần thiết để xử lý dữ liệu nhận được từ tầng trình bày trước khi tương tác với tầng dữ liệu.

+ Ví dụ: Trong ứng dụng quản lý giao dịch nhà đất thì tầng này bao gồm các phương thức tính tổng số tiền thanh toán dựa trên diện tích và đơn giá từ người dùng nhập vào.

- *Persistence*

+ Nhiệm vụ: Quản lý việc lưu trữ và truy xuất dữ liệu. Tầng này tương tác với cơ sở dữ liệu hoặc các hệ thống lưu trữ khác để lưu trữ và truy xuất dữ liệu theo yêu cầu từ tầng logic.

+ Ví dụ: Trong ứng dụng quản lý giao dịch nhà đất thì tầng này sẽ kết nối đến cơ sở dữ liệu và làm các chức năng liên quan như thêm, xóa, sửa, tìm kiếm, ...

1.2.2. Ưu nhược điểm của mô hình 3 tầng Three Layer

- Ưu điểm:

+ Phân chia rõ ràng trách nhiệm: Giúp tách biệt các phần của ứng dụng, làm cho mã nguồn dễ bảo trì và quản lý.

+ Tăng khả năng tái sử dụng mã: Mỗi tầng có thể được tái sử dụng trong các dự án khác nhau hoặc các phần khác của ứng dụng.

+ Dễ dàng mở rộng: Mỗi tầng có thể được mở rộng độc lập, giúp ứng dụng dễ dàng mở rộng khi cần thiết.

+ Bảo mật tốt hơn: Việc tách biệt các tầng giúp tăng cường bảo mật bằng cách kiểm soát truy cập và bảo vệ dữ liệu ở tầng dữ liệu.

- Nhược điểm:

+ Độ phức tạp ban đầu: Việc thiết lập và duy trì mô hình 3 tầng có thể phức tạp, đặc biệt đối với các ứng dụng nhỏ.

+ Hiệu suất: Việc phân chia ứng dụng thành ba tầng có thể làm giảm hiệu suất do cần phải truyền dữ liệu giữa các tầng.

+ Yêu cầu kiến thức và kỹ năng: Lập trình viên cần hiểu rõ về các tầng và cách thức tương tác giữa chúng.

1.3. Mẫu Observer

1.3.1. Lý thuyết

Mẫu thiết kế Observer (Observer Pattern) là một trong những mẫu thiết kế hành vi (Behavioral Design Patterns) được sử dụng phổ biến trong lập trình hướng đối tượng. Mẫu này định nghĩa một mối quan hệ một-nhiều giữa các đối tượng để khi một đối tượng thay đổi trạng thái, tất cả các đối tượng phụ thuộc của nó sẽ được thông báo và cập nhật tự động.

+ *Publisher*: Đối tượng chứa trạng thái và có thể có nhiều Subscriber đăng ký để nhận thông báo khi có sự thay đổi trạng thái. Có các phương thức chủ yếu:

+ `registerSubscriber(subscriber: Subscriber)`: Đăng ký một subscriber.

+ `removeSubscriber(subscriber: Subscriber)`: Hủy đăng ký một subscriber..

+ `notifySubscriber()`: Thông báo đến tất cả các subscriber đăng ký về sự thay đổi trạng thái.

+ *Subscriber*: Đối tượng cần được thông báo khi Subject thay đổi trạng thái. Có các phương thức chủ yếu:

+ `update()`: Được gọi bởi Publisher khi có sự thay đổi trạng thái.

1.3.2. Ưu nhược điểm của mẫu Observer

- Ưu điểm:

+ Giảm sự phụ thuộc: Các đối tượng Observer và Subject tách biệt và độc lập, giúp giảm sự phụ thuộc lẫn nhau.

+ Dễ dàng mở rộng: Có thể dễ dàng thêm các observer mới mà không cần thay đổi Subject.

+ Cập nhật tự động: Observer được cập nhật tự động khi Subject thay đổi, giúp duy trì tính nhất quán của dữ liệu.

- Nhược điểm:

+ Hiệu suất: Khi có nhiều observers, việc thông báo có thể tốn kém về thời gian và tài nguyên.

+ Khó khăn trong gỡ lỗi: Các sự kiện không đồng bộ có thể làm cho việc theo dõi và gỡ lỗi trở nên phức tạp.

1.4. Mẫu Command Processor

1.4.1. Lý thuyết

Mẫu Command là một mẫu thiết kế hành vi (behavioral pattern) cho phép bạn tách biệt việc yêu cầu thực hiện một hành động từ việc thực hiện hành động đó. Mẫu này giúp tổ chức các yêu cầu dưới dạng các đối tượng, cho phép dễ dàng thêm, xóa hoặc thay đổi các hành động mà không cần thay đổi mã nguồn của các đối tượng thực hiện.

- *Command*: Định nghĩa giao diện cho các lệnh cụ thể. Các phương thức chủ yếu
 - + `execute()`: Thực hiện hành động tương ứng
- *CommandProcessor*: Lưu trữ và thực thi lệnh.

1.4.2. Ưu nhược điểm của mẫu Command Processor

- Ưu điểm:

- + Tách biệt yêu cầu và hành động: Lệnh và hành động được tách biệt, giúp mã nguồn dễ bảo trì và mở rộng.

- + Hỗ trợ undo/redo: Dễ dàng thêm hỗ trợ undo và redo cho các hành động đã thực hiện.

- + Dễ mở rộng: Có thể dễ dàng thêm các lệnh mới mà không cần thay đổi mã nguồn hiện tại.

- + Quản lý lệnh tốt hơn: Lưu trữ và quản lý các lệnh cho phép thực hiện các thao tác như ghi lại lịch sử và thực hiện các hành động đồng bộ.

- Nhược điểm:

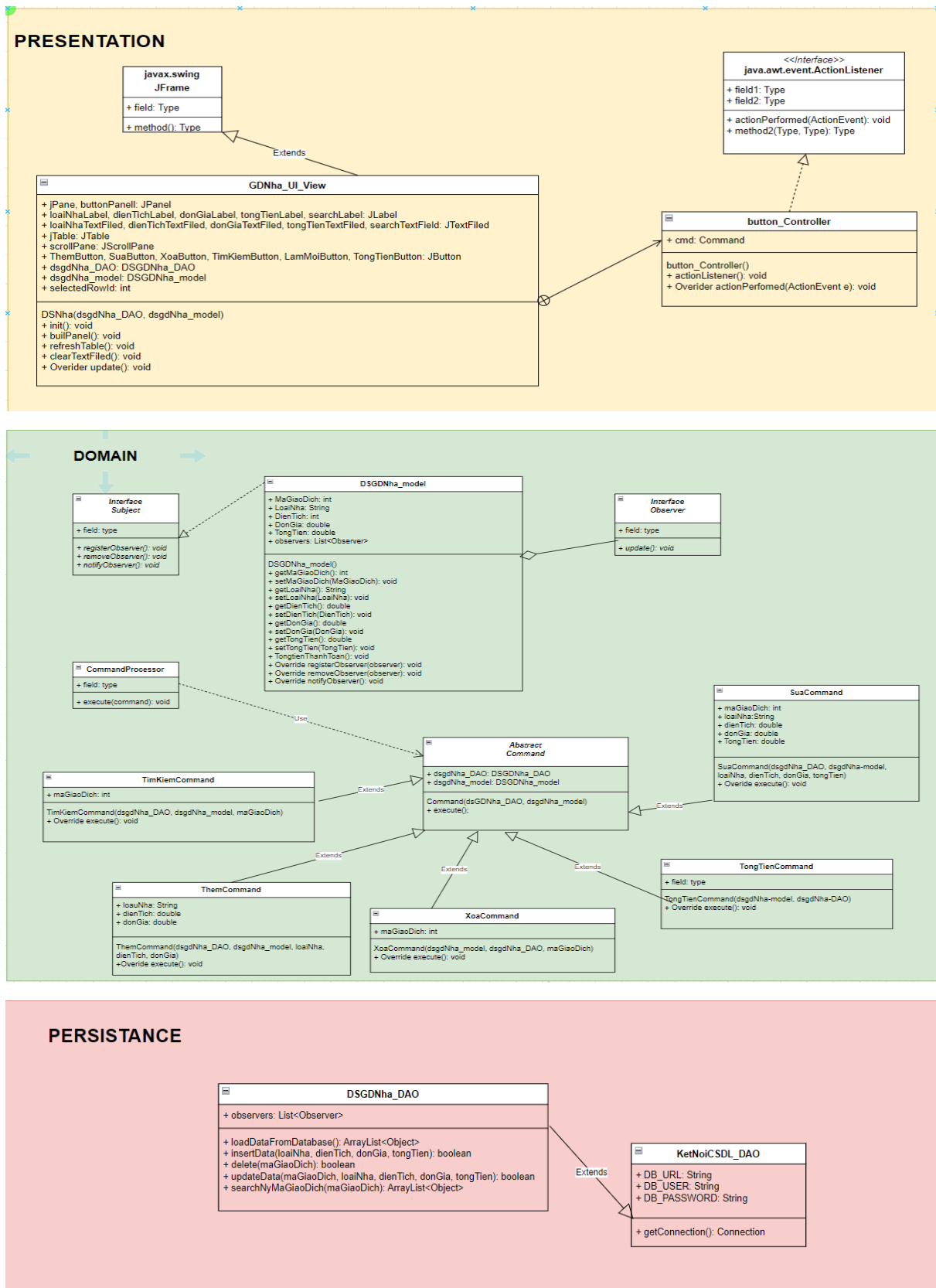
- + Tăng độ phức tạp: Mẫu này có thể làm tăng độ phức tạp của mã nguồn do cần nhiều lớp và đối tượng để tổ chức và quản lý các lệnh.

- + Chi phí tài nguyên: Việc lưu trữ các lệnh và trạng thái của chúng có thể tiêu tốn tài nguyên, đặc biệt là trong các hệ thống lớn với nhiều lệnh.

- + Cần phải quản lý trạng thái: Nếu có hỗ trợ undo/redo, bạn cần phải quản lý trạng thái trước và sau khi thực hiện lệnh, điều này có thể làm mã nguồn trở nên phức tạp hơn.

CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ PHẦN MỀM

2.1. Sơ đồ thiết kế 3 tầng Three Layer



2.2. Phân tích các chức năng của ứng dụng

2.2.1. Chức năng Thêm giao dịch

- Khi người dùng nhập dữ liệu vào các TextField tương ứng và bấm vào nút THÊM thì dữ liệu được người dùng nhập vào sẽ được thêm vào cơ sở dữ liệu MySQL và sẽ được cập nhật và hiển thị lên bảng giao diện người dùng.

2.2.2. Chức năng Sửa giao dịch

- Khi người dùng chọn 1 trường thông tin được hiển thị trong bảng thì các thông tin, dữ liệu đó sẽ được hiển thị lên các TextField tương ứng và người dùng có thể sửa lại thông tin theo ý muốn và khi bấm vào nút SỬA thì dữ liệu sẽ được update lại trong cơ sở dữ liệu MySQL và sẽ được cập nhật và hiển thị lên bảng giao diện người dùng.

2.2.3. Chức năng Xóa giao dịch

- Khi người dùng chọn 1 trường thông tin được hiển thị trong bảng và bấm vào nút XÓA thì sẽ hiển thị lên 1 thông báo “Bạn có chắc muốn xóa” nếu chọn YES thì trường thông tin đó sẽ xóa khỏi cơ sở dữ liệu MySQL và bảng giao diện người dùng còn nếu chọn NO thì nó sẽ không có xóa trường thông tin đó và thông tin đó vẫn sẽ được lưu trong cơ sở dữ liệu MySQL và vẫn hiển thị trong bảng giao diện người dùng.

2.2.4. Chức năng Tìm kiếm giao dịch

- Trong phạm vi đề tài “Xây dựng chương trình quản lý giao dịch nhà đất” này thì việc tìm kiếm sẽ được lọc theo mã giao dịch. Trong giao diện người dùng có TextField dành cho việc tìm kiếm thì khi người dùng nhập vào mã giao dịch cần tìm và nhấn vào nút TÌM KIẾM thì thông tin sẽ được lọc trong cơ sở dữ liệu MySQL nếu có mã giao dịch người dùng cần tìm thì nó sẽ hiển thị lên bảng giao diện người dùng còn không thì sẽ không hiển thị gì hết.


2.2.5. Chức năng Làm mới

- Khi người dùng nhấn nhập thông tin vào các TextField và muốn xóa nó đi mà không muốn xóa thủ công từng TextField thì chỉ cần người dùng bấm vào nút LÀM MỚI thì tất cả dữ liệu được nhập sẽ bị xóa đi.

2.2.6. Chức năng Tổng tiền

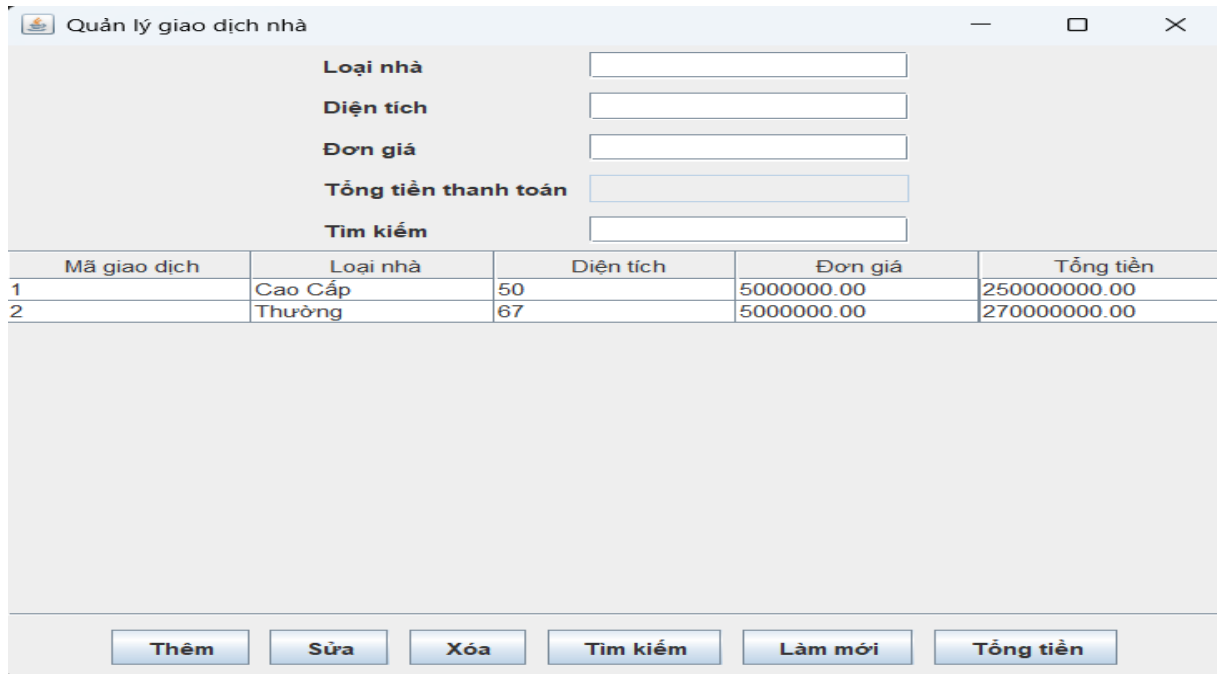
- Trong phạm vi đề tài “Xây dựng chương trình quản lý giao dịch nhà đất” này thì tổng tiền sẽ được tính dựa trên loại nhà, diện tích và đơn giá do người dùng nhập vào. Nếu người dùng nhập vào loại nhà là “Cao cấp” thì “tổng tiền = đơn giá * diện tích”, còn nếu người dùng nhập vào loại nhà là “Thường” thì “tổng tiền = đơn giá * diện tích * 90%”. Thì sau khi người dùng nhập vào thông tin cần thiết và nhấn nút TỔNG TIỀN thì số tiền tổng sẽ được hiện lên TextField Tổng tiền thanh toán.

2.3. Sơ đồ thiết kế cơ sở dữ liệu

DSGDNha	
	maGiaoDich
	loaiNha
	dienTich
	donGia

CHƯƠNG 3: CÀI ĐẶT THỰC NGHIỆM VÀ KẾT QUẢ CHƯƠNG TRÌNH

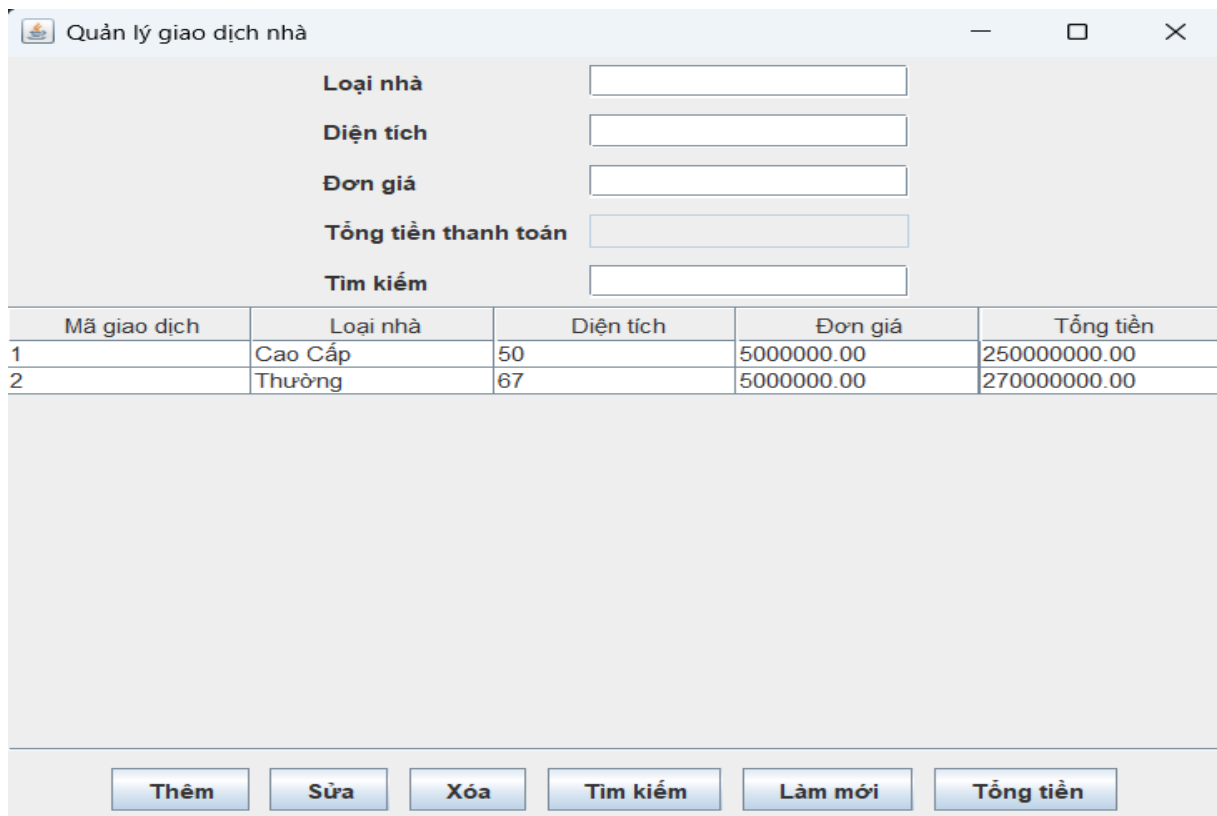
3.1. Giao diện hiển thị chính



Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

3.2. Chức năng “Thêm”

- Giao diện trước khi sử dụng chức năng “Thêm”



Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

- Giao diện sau khi sử dụng chức năng “Thêm”

Quản lý giao dịch nhà

Loại nhà

Cao cấp

Diện tích

100

Đơn giá

4500000

Tổng tiền thanh toán

450000000.00

Tim kiếm

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00
20	Cao cấp	450000000	450000000.00	450000000.00

Thêm

Sửa

Xóa

Tim kiếm

Làm mới

Tổng tiền

3.3. Chức năng “Sửa”

- Giao diện trước khi sử dụng chức năng “Sửa”

Quản lý giao dịch nhà

Loại nhà

Diện tích

Đơn giá

Tổng tiền thanh toán

Tim kiếm

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00
20	Cao cấp	450000000	450000000.00	450000000.00

Thêm

Sửa

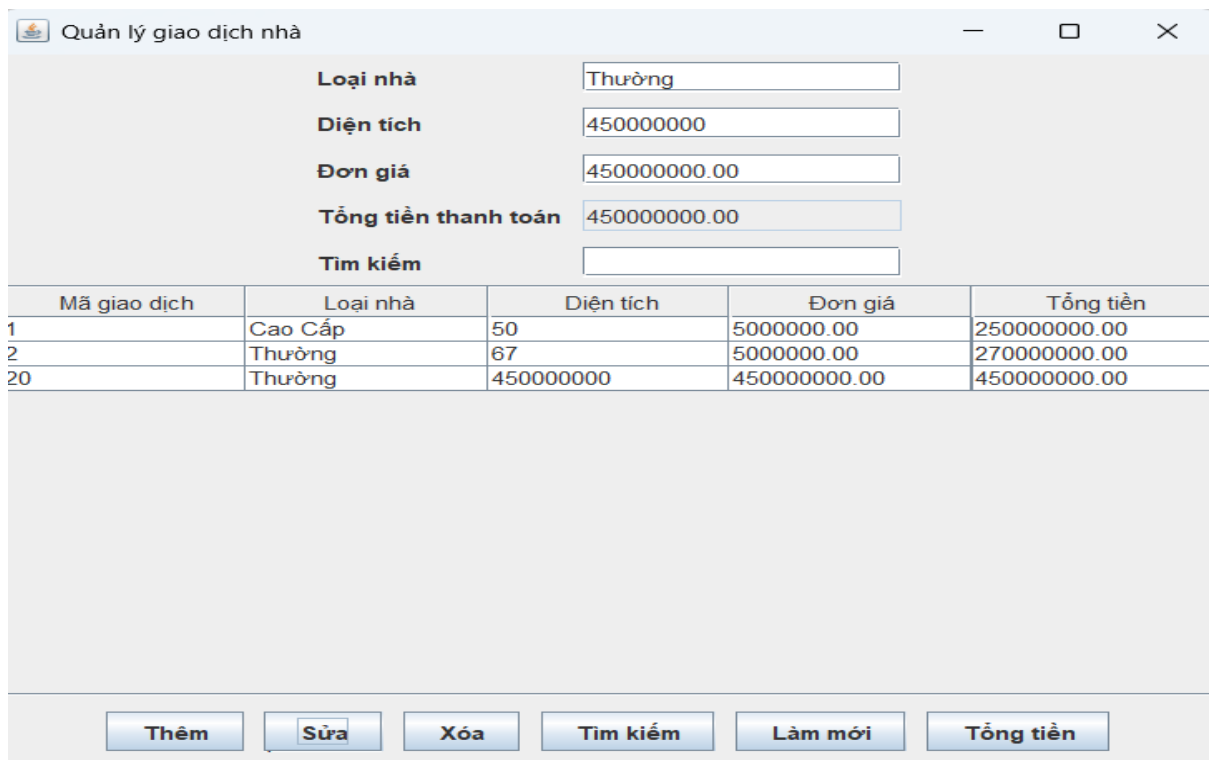
Xóa

Tim kiếm

Làm mới

Tổng tiền

- Giao diện sau khi sử dụng chức năng “Sửa”



Quản lý giao dịch nhà

Loại nhà:

Diện tích:

Đơn giá:

Tổng tiền thanh toán:

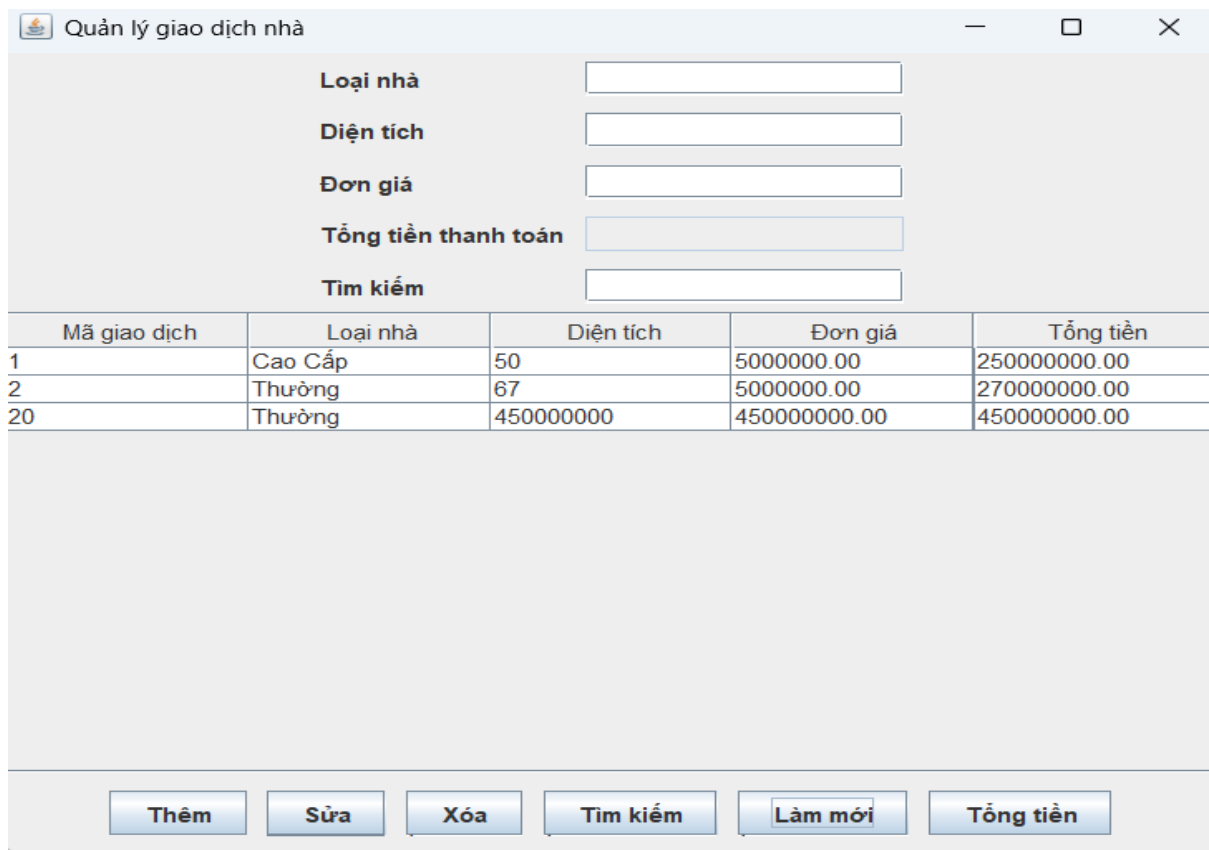
Tìm kiếm:

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00
20	Thường	450000000	450000000.00	450000000.00

Thêm Sửa Xóa Tìm kiếm Làm mới Tổng tiền

3.4. Chức năng “Xóa”

- Giao diện trước khi sử dụng chức năng “Xóa”



Quản lý giao dịch nhà

Loại nhà:

Diện tích:

Đơn giá:

Tổng tiền thanh toán:

Tìm kiếm:

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00
20	Thường	450000000	450000000.00	450000000.00

Thêm Sửa Xóa Tìm kiếm Làm mới Tổng tiền

- Giao diện sau khi sử dụng chức năng “Xóa”

Quản lý giao dịch nhà

Loại nhà: Thường

Diện tích: 450000000


Đơn giá: 450000000.00

Tổng tiền thanh toán: 450000000.00

Tim kiếm:

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00
20	Thường	450000000	450000000.00	450000000.00

Xác nhận xóa

 Bạn có chắc chắn muốn xóa giao dịch này?

Yes No

Thêm Sửa Xóa Tim kiếm Làm mới Tổng tiền

Quản lý giao dịch nhà

Loại nhà: Thường

Diện tích: 450000000

Đơn giá: 450000000.00

Tổng tiền thanh toán: 450000000.00

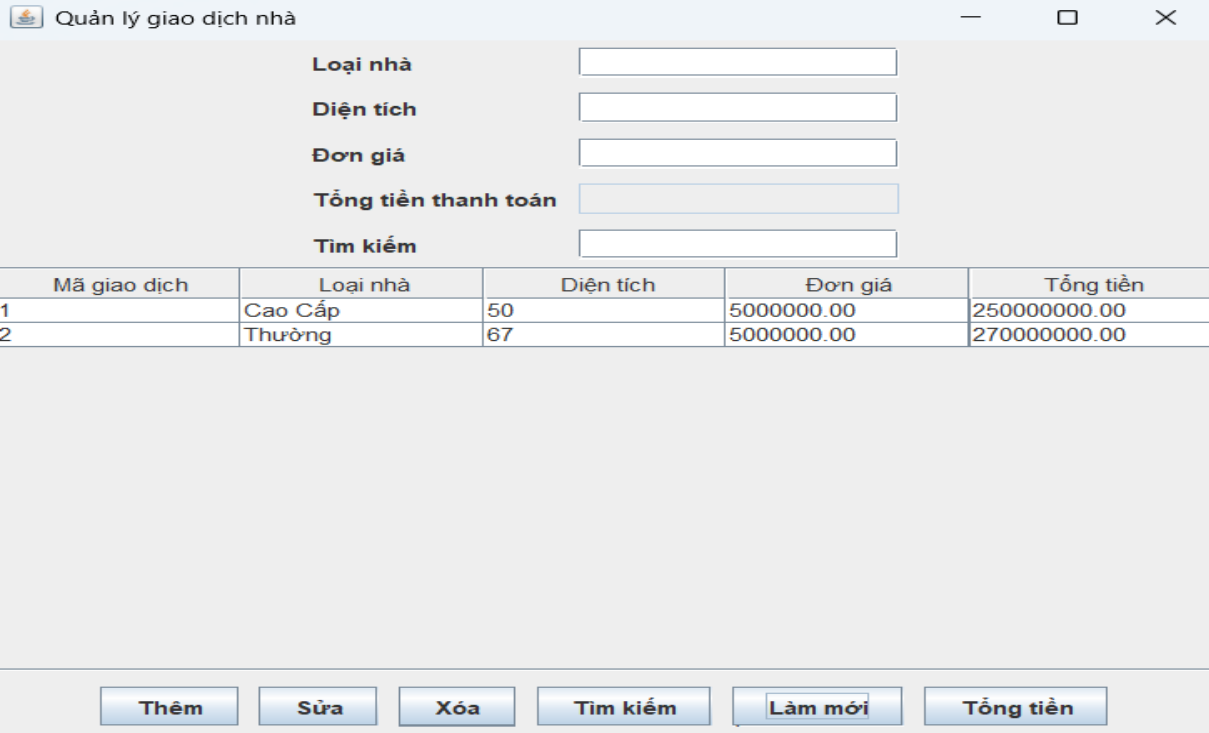
Tim kiếm:

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

Thêm Sửa Xóa Tim kiếm Làm mới Tổng tiền

3.5. Chức năng “Tìm Kiếm”

- Giao diện trước khi sử dụng chức năng “Tìm Kiếm”



Quản lý giao dịch nhà

Loại nhà:

Diện tích:

Đơn giá:

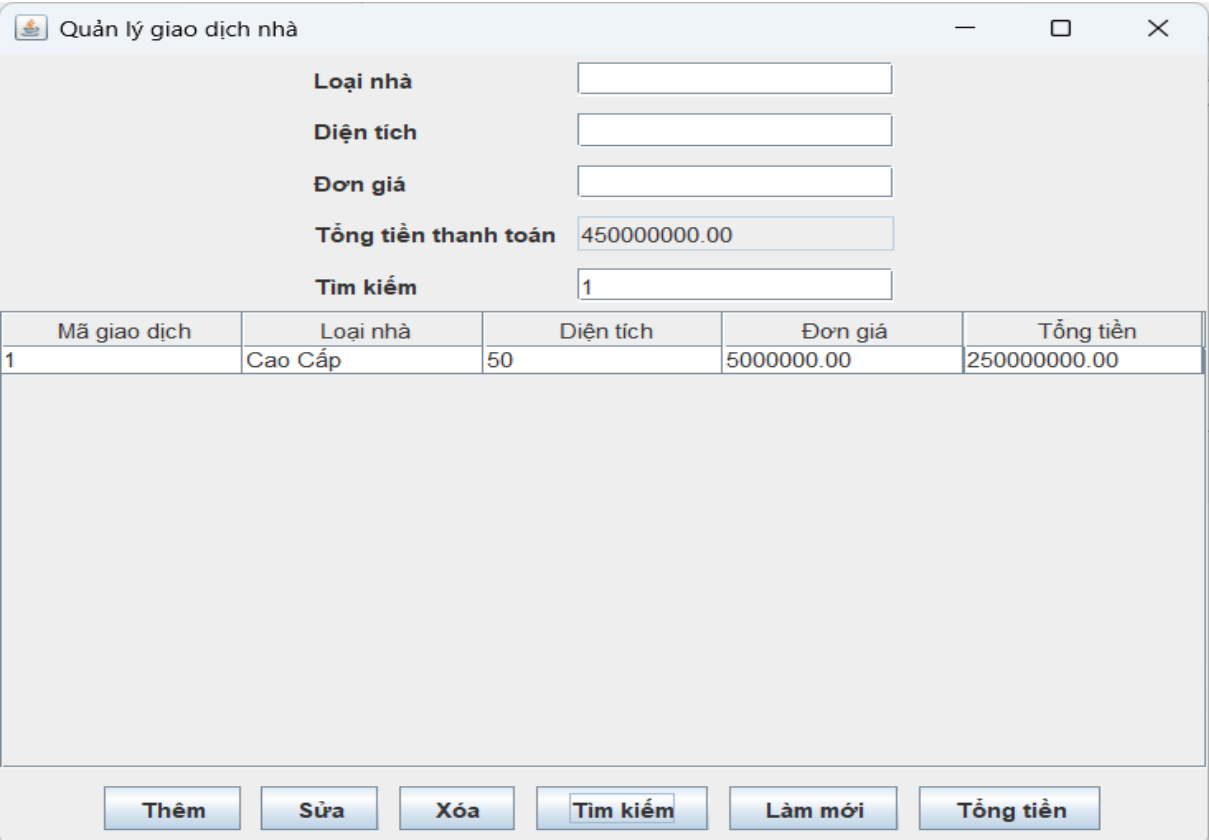
Tổng tiền thanh toán:

Tìm kiếm:

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

Thêm Sửa Xóa Tìm kiếm Làm mới Tổng tiền

- Giao diện sau khi sử dụng chức năng “Tìm Kiếm”



Quản lý giao dịch nhà

Loại nhà:

Diện tích:

Đơn giá:

Tổng tiền thanh toán: 450000000.00

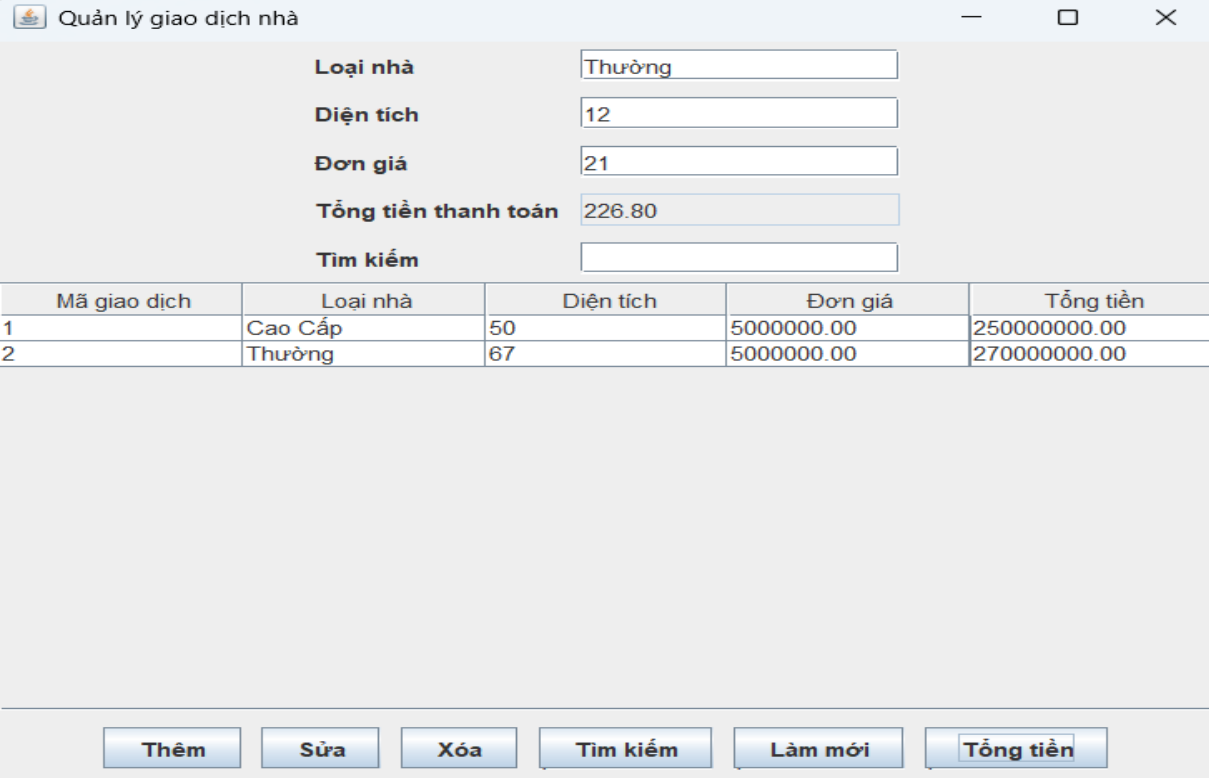
Tìm kiếm: 1

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00

Thêm Sửa Xóa Tìm kiếm Làm mới Tổng tiền

3.6. Chức năng “Làm Mới”

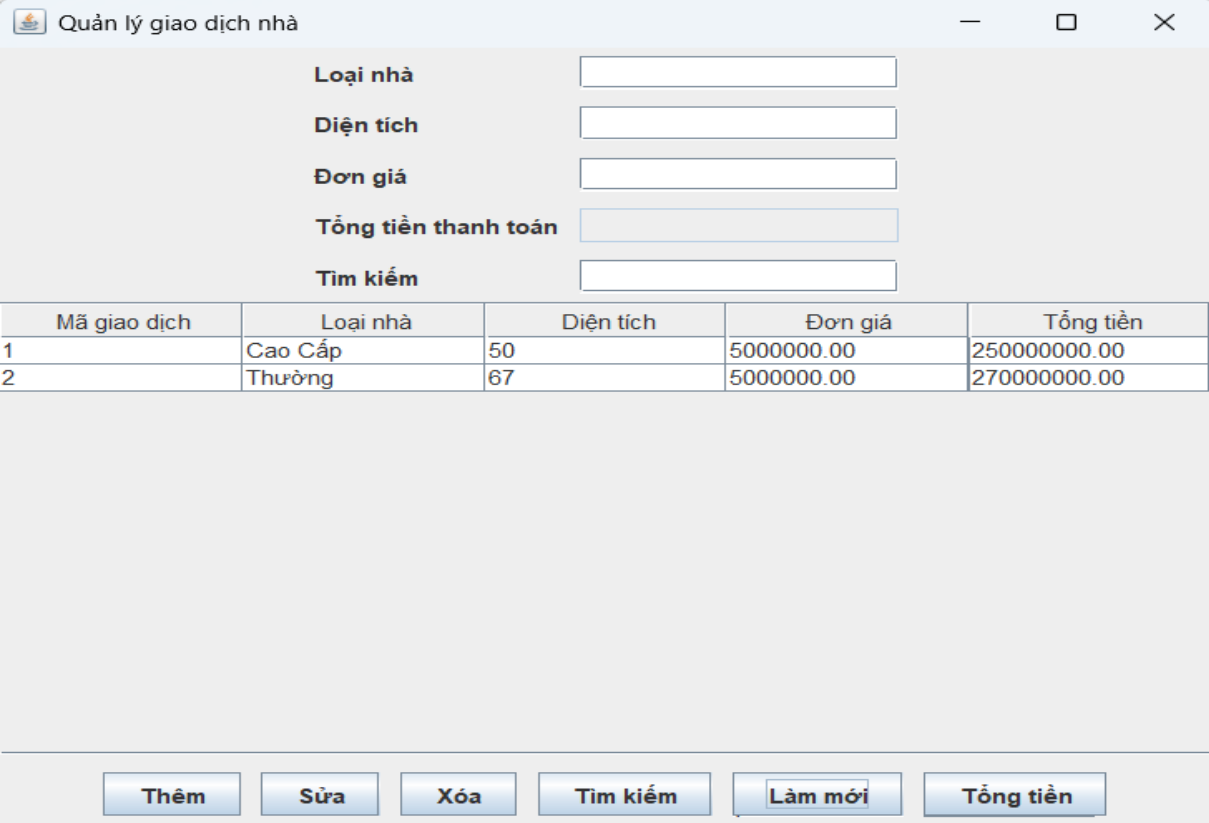
- Giao diện trước khi sử dụng chức năng “Làm Mới” đối với TextField



Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

Buttons: Thêm, Sửa, Xóa, Tim kiếm, Làm mới, Tổng tiền

- Giao diện sau khi sử dụng chức năng “Làm Mới” đối với TextField



Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

Buttons: Thêm, Sửa, Xóa, Tim kiếm, Làm mới, Tổng tiền

- Giao diện trước khi sử dụng chức năng “Làm Mới” đối với dữ liệu trong bảng

Quản lý giao dịch nhà

Loại nhà

Diện tích

Đơn giá

Tổng tiền thanh toán

226.80

Tìm kiếm

2

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
2	Thường	67	5000000.00	270000000.00

Thêm

Sửa

Xóa

Tìm kiếm

Làm mới

Tổng tiền

- Giao diện sau khi sử dụng chức năng “Làm Mới” đối với dữ liệu trong bảng

Quản lý giao dịch nhà

Loại nhà

Diện tích

Đơn giá

Tổng tiền thanh toán

Tìm kiếm

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

Thêm

Sửa

Xóa

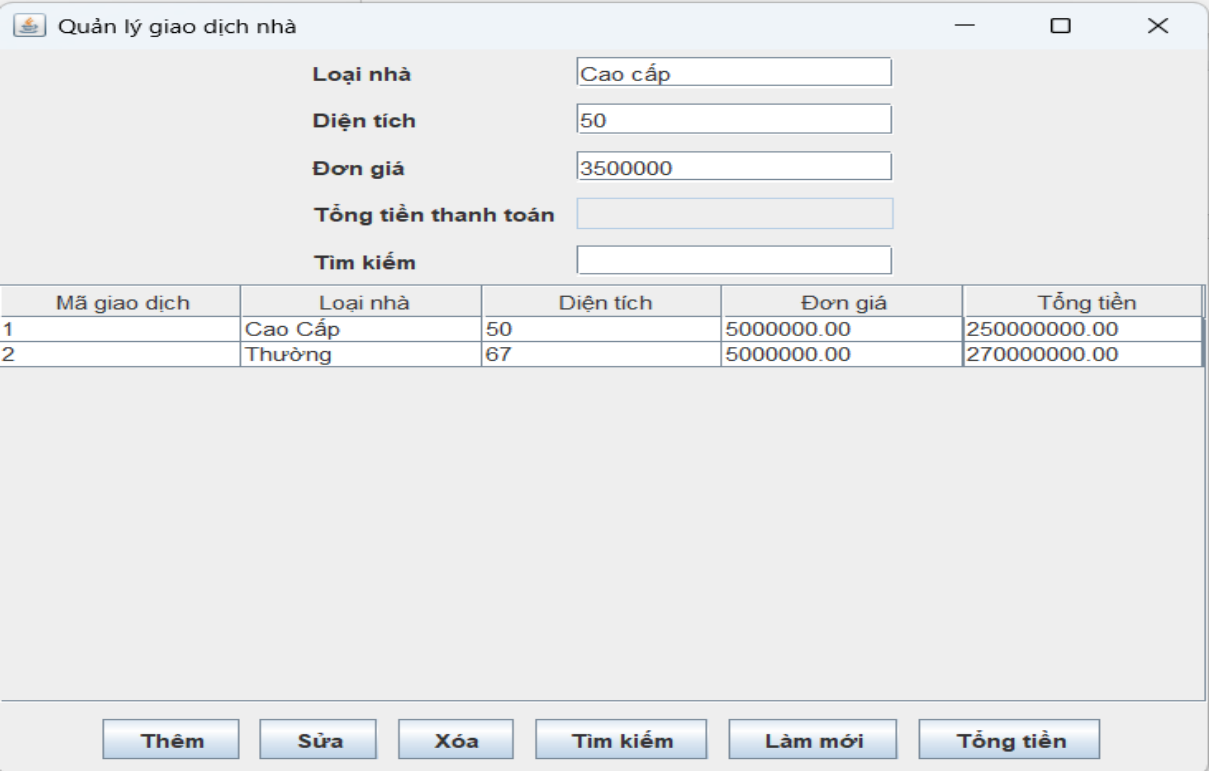
Tìm kiếm

Làm mới

Tổng tiền

3.7. Chức năng “Tổng tiền”

- Giao diện trước khi sử dụng chức năng “Tổng tiền”



Quản lý giao dịch nhà

Loại nhà: Cao cấp

Diện tích: 50

Đơn giá: 3500000

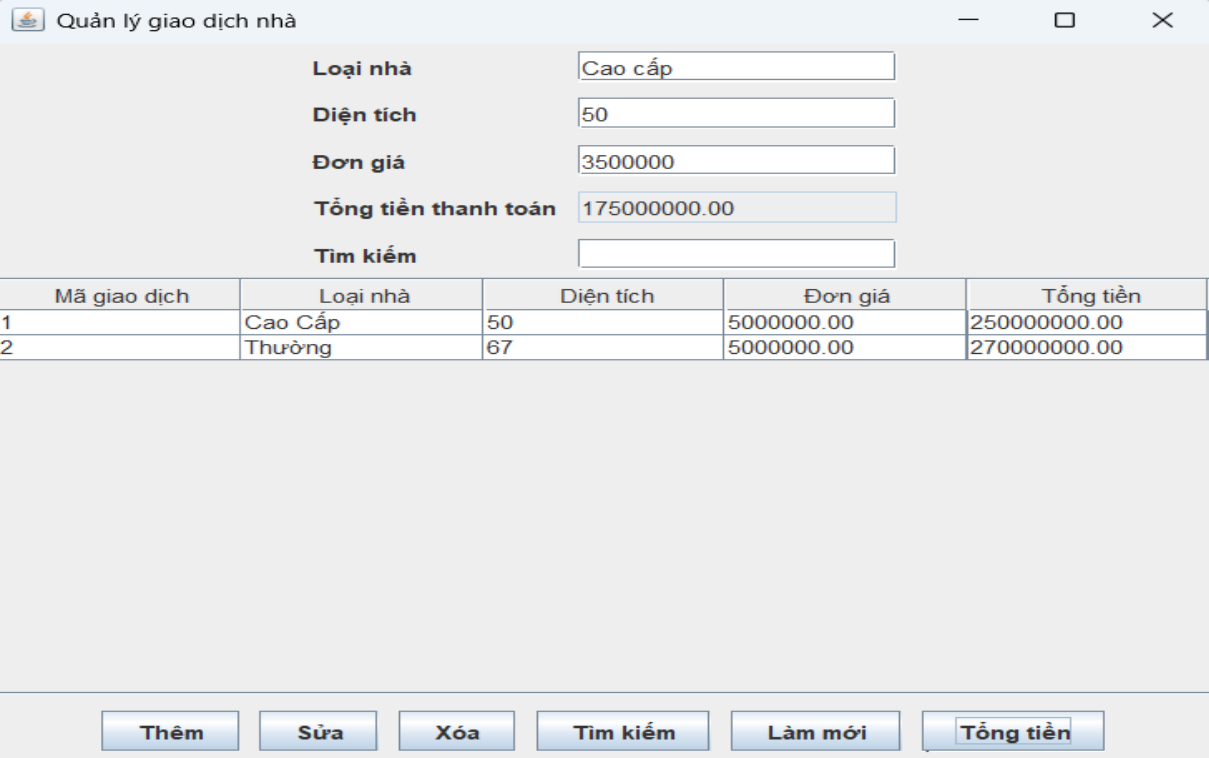
Tổng tiền thanh toán:

Tim kiếm:

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

Thêm Sửa Xóa Tim kiếm Làm mới Tổng tiền

- Giao diện sau khi sử dụng chức năng “Tổng tiền” thì số tiền tổng sẽ hiển thị ở TextField Tổng tiền thanh toán



Quản lý giao dịch nhà

Loại nhà: Cao cấp

Diện tích: 50

Đơn giá: 3500000

Tổng tiền thanh toán: 175000000.00

Tim kiếm:

Mã giao dịch	Loại nhà	Diện tích	Đơn giá	Tổng tiền
1	Cao Cấp	50	5000000.00	250000000.00
2	Thường	67	5000000.00	270000000.00

Thêm Sửa Xóa Tim kiếm Làm mới Tổng tiền

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết luận

- Xây dựng chương trình quản lý các giao dịch nhà đất không chỉ là một nhiệm vụ chiến lược quan trọng mà còn là một cơ hội để tối ưu hóa quy trình quản lý thông tin và giao dịch trong lĩnh vực bất động sản. Đề tài này thường yêu cầu một thiết kế hệ thống có tính tổ chức cao, được triển khai theo mô hình 3 tầng, bao gồm các lớp giao diện người dùng, logic nghiệp vụ và truy cập dữ liệu. Mô hình này giúp tách biệt các phần của hệ thống, từ đó nâng cao khả năng bảo trì và mở rộng, đồng thời đảm bảo tính linh hoạt trong việc cập nhật và tích hợp các tính năng mới. Việc áp dụng mẫu Command trong thiết kế đóng vai trò quan trọng, giúp tổ chức và quản lý các lệnh xử lý dữ liệu một cách rõ ràng và hiệu quả. Mẫu Command hỗ trợ các tính năng nâng cao như undo/redo, cho phép khôi phục trạng thái trước đó khi có sai sót hoặc yêu cầu điều chỉnh, từ đó tăng cường độ tin cậy và khả năng kiểm soát của hệ thống.

- Chương trình quản lý giao dịch nhà đất mang lại nhiều lợi ích rõ ràng, bao gồm tính linh hoạt, khả năng mở rộng dễ dàng và sự chính xác trong việc xử lý và lưu trữ thông tin giao dịch. Điều này không chỉ giúp giảm thiểu sai sót mà còn nâng cao hiệu quả trong việc ra quyết định. Tuy nhiên, phát triển và duy trì hệ thống này không phải là một nhiệm vụ đơn giản. Nó đối mặt với các thách thức như độ phức tạp cao và chi phí tài nguyên đáng kể trong giai đoạn triển khai. Để cải thiện và tận dụng tối đa những lợi ích mà hệ thống mang lại, có thể xem xét việc tích hợp các tính năng mới như phân tích dữ liệu nâng cao, báo cáo tùy chỉnh và tích hợp với các hệ thống khác. Đồng thời, việc tối ưu hóa hiệu suất của hệ thống là cần thiết để đảm bảo đáp ứng nhanh chóng với số lượng giao dịch lớn và dữ liệu phức tạp.

4.2. Hướng phát triển

- Tích Hợp Công Nghệ Mới: Để giữ cho hệ thống luôn cập nhật với công nghệ mới, có thể xem xét việc tích hợp trí tuệ nhân tạo (AI) và học máy (machine learning) để phân tích xu hướng giao dịch, dự đoán giá trị bất động sản và tối ưu hóa quy trình ra quyết định. Công nghệ AI có thể hỗ trợ trong việc phát hiện mẫu giao dịch bất thường và cung cấp các khuyến nghị chính xác dựa trên dữ liệu lịch sử.

- Nâng Cao Trải Nghiệm Người Dùng: Đầu tư vào việc cải thiện giao diện người dùng (UI) và trải nghiệm người dùng (UX) để làm cho hệ thống trở nên dễ sử dụng và trực quan hơn. Việc thiết kế giao diện thân thiện và cung cấp các tính năng tùy chỉnh cho người dùng có thể giúp nâng cao sự hài lòng và hiệu quả làm việc.

- Tích Hợp Dữ Liệu và Hệ Thống: Xem xét việc tích hợp với các hệ thống khác như hệ thống CRM (Quản lý quan hệ khách hàng) và ERP (Hoạch định nguồn lực doanh nghiệp) để cung cấp cái nhìn tổng thể và đồng bộ hóa dữ liệu giữa các bộ phận khác nhau trong tổ chức.

- Cải Thiện Bảo Mật: Đảm bảo rằng hệ thống có các cơ chế bảo mật mạnh mẽ để bảo vệ dữ liệu nhạy cảm và ngăn chặn các mối đe dọa từ bên ngoài. Điều này bao gồm việc mã hóa dữ liệu, xác thực đa yếu tố và thường xuyên kiểm tra bảo mật.

TÀI LIỆU THAM KHẢO

1. NGUYỄN QUANG TRUNG. Xuất bản năm 2016. *“Nhập môn cơ sở dữ liệu”*.
2. NGUYỄN NGỌC HOÀI. Xuất bản năm 2015. *“Thiết kế phần mềm với UML”*.
3. NGUYỄN MINH TUẤN. Xuất bản năm 2019. *“Lập trình Swing với Java”*.
4. TRẦN HOÀI NAM. Xuất bản năm 2018. *“Thiết kế mẫu: Tài liệu tham khảo cho lập trình viên”*.