

MỤC LỤC

1. GIỚI THIỆU	2
1.1. Mục đích	2
1.2. Môi trường phát triển	2
1.3. Phạm vi.....	2
1.4. Định nghĩa, các từ viết tắt và kí hiệu	3
1.5. Tài liệu tham khảo.....	3
1.6. Tổng quan	3
2. YÊU CẦU PHẦN CỨNG	4
3. BIỂU DIỄN KIẾN TRÚC	4
3.1. Tầng trình diễn (View)	5
3.2. Tầng điều khiển (Controller)	5
3.3. Tầng mô hình dữ liệu (Model).....	6
3.4. Lý do chọn mô hình MVC:	6
4. GÓC NHÌN LOGIC:	7
4.1. Tổng quan	7
4.2. Cấu trúc thiết kế quan trọng của các Packages / Components	7
4.2.1. Tầng trình diễn (View):.....	7
4.2.2. Tầng điều khiển (Controller):.....	8
4.2.3. Tầng mô hình dữ liệu (Model).....	8

1. Giới thiệu

1.1. Mục đích

Tài liệu này cung cấp cái nhìn tổng thể về kiến trúc của các mô-đun trong phần mềm "AireLeaf", sử dụng các cách tiếp cận khác nhau để mô tả chi tiết từng khía cạnh của hệ thống. Mục tiêu của tài liệu là ghi lại và truyền đạt những quyết định quan trọng về kiến trúc đã được đưa ra, đóng vai trò như cầu nối giữa các yêu cầu phần mềm và thiết kế chi tiết của "AireLeaf". Tài liệu cũng hỗ trợ đội ngũ kỹ sư phần mềm trong việc đảm bảo các mô-đun được phát triển đáp ứng đúng nhu cầu chức năng của người dùng, đồng thời phù hợp với nền tảng và công nghệ mà hệ thống yêu cầu, giúp đảm bảo tính ổn định, khả năng bảo trì và khả năng mở rộng của phần mềm.

1.2. Môi trường phát triển

Ứng dụng được phát triển trong môi trường phần mềm sau:

- Môi trường phát triển: Visual Studio.NET 2022
- Khung: .NET Framework 4.7.2
- Giao diện người dùng: Windows Forms
- Ngôn ngữ lập trình: C#.NET Framework
- Truy cập dữ liệu : ADO.NET
- Cơ sở dữ liệu: SQL Server Management Studio 19.3.4.0
- Công cụ phân tích mã: Stylecop

1.3. Phạm vi

Tài liệu này định nghĩa kiến trúc phần mềm của AireLeaf từ một góc độ cao cấp bao gồm các khía cạnh sau:

- Chức năng: Đảm bảo kiến trúc hỗ trợ toàn bộ chức năng cần thiết của hệ thống AireLeaf, từ quản lý nhân viên, hợp đồng đến tương tác và phản hồi với người dùng.

- **Sẵn sàng sử dụng:** Đảm bảo hệ thống có độ sẵn sàng cao, tránh các gián đoạn hoặc sự cố không mong muốn trong quá trình sử dụng.
- **Độ tin cậy:** Kiến trúc được thiết kế để vận hành ổn định, có khả năng xử lý lỗi và duy trì hiệu suất trong mọi tình huống.
- **Khả năng mở rộng:** Đảm bảo tính linh hoạt trong việc mở rộng để thích ứng với sự gia tăng dữ liệu và yêu cầu của người dùng mà không gặp các trở ngại lớn.
- **Dễ bảo trì:** Cung cấp một kiến trúc thuận lợi cho việc bảo trì, giúp cho việc sửa chữa, cập nhật và cải tiến hệ thống diễn ra nhanh chóng và hiệu quả.
- **Dễ quản lý:** Thiết kế kiến trúc với khả năng quản lý dễ dàng, hỗ trợ việc giám sát, điều chỉnh và tối ưu hóa hiệu suất của hệ thống.

1.4. Định nghĩa, các từ viết tắt và kí hiệu

STT	Viết tắt	Tên đầy đủ	Tên tiếng Việt
1	GUI	Graphical User Interface	Giao diện người dùng
2	BLL	Business Logic Layer	Tầng logic nghiệp vụ
3	DAO	Data Access Object	Đối tượng truy cập Dữ Liệu
4	DTO	Data Transfer Object	Đối tượng truyền tải Dữ liệu

1.5. Tài liệu tham khảo

STT	Tài liệu	Ngày	Tác giả
1	SRS	3/10/2024	BA
2	Tài liệu thiết kế giao diện	1/11/2024	Design

1.6. Tổng quan

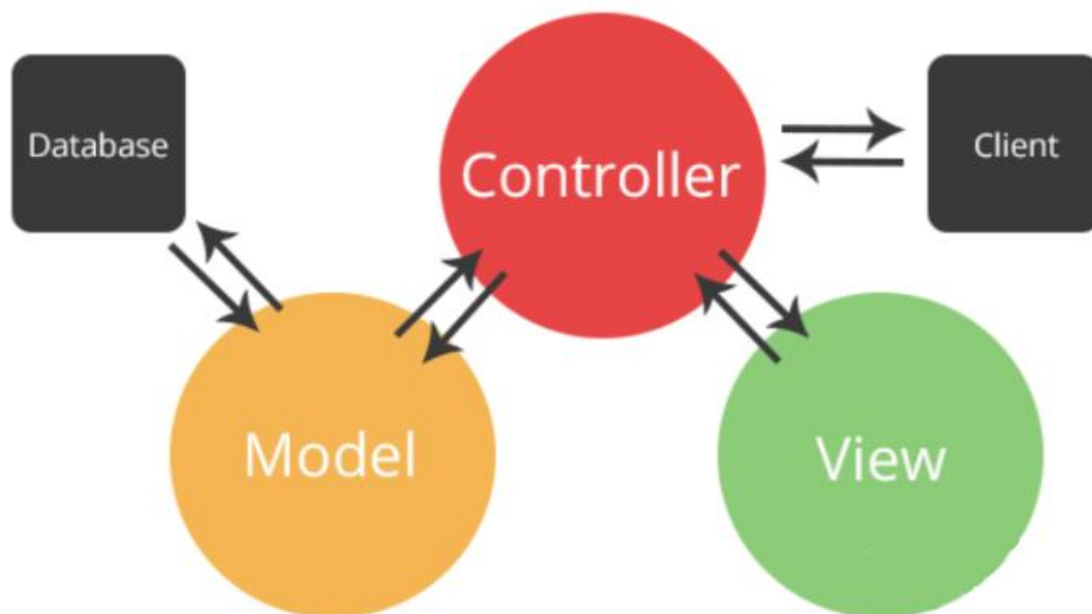
Các phần sau sẽ cung cấp cái nhìn sâu hơn về kiến trúc các mô-đun trong AireLeaf, giải thích chức năng và khả năng mở rộng của kiến trúc để đáp ứng yêu cầu thay đổi từ người dùng hoặc tích hợp các thành phần bên ngoài. Phân tích từ các góc nhìn Chức năng và Logic nhằm đảm bảo kiến trúc đáp ứng đầy đủ yêu cầu.

2. Yêu cầu phần cứng

Yêu cầu phần cứng	Phân loại	Yêu cầu
Hệ điều hành	Bắt buộc	Windows 7 trở lên
Bộ xử lý	Đề xuất	Intel Core i3 trở lên hoặc tương đương
Bộ nhớ	Đề xuất	Ít nhất 4GB RAM
Dung lượng đĩa cứng	Đề xuất	Ít nhất 50MB không gian trống
Đồ họa	Đề xuất	Card đồ họa hỗ trợ DirectX 9
Màn hình	Đề xuất	Độ phân giải ít nhất 1024x768
Kết nối Internet	Tùy chọn	Có thể yêu cầu
Thiết bị ngoại vi	Tùy chọn	Bàn phím, chuột, Máy in (tùy chọn)
Backup và Bảo mật	Đề xuất	Thiết bị lưu trữ dữ liệu, phần mềm diệt virus

3. Biểu diễn Kiến trúc

Biểu diễn Kiến trúc của hệ thống được thiết kế dựa trên một mô hình MVC, giúp phân tách rõ ràng ba thành phần chính: Model, View và Controller. Mỗi thành phần đảm nhiệm một chức năng cụ thể, hỗ trợ tối ưu hóa quản lý, bảo trì và mở rộng ứng dụng.



3.1. Tầng trình diễn (View)

Giao diện người dùng (GUI):

- Thành phần này đại diện cho phần giao diện mà người dùng tương tác, bao gồm các yếu tố đồ họa như các form, điều khiển, và các nút lệnh.
- **Chức năng:** Hiển thị dữ liệu từ Model cho người dùng, đồng thời thu thập và truyền dữ liệu nhập vào từ người dùng đến tầng điều khiển (Controller).

3.2. Tầng điều khiển (Controller)

Logic nghiệp vụ (BLL - Business Logic Layer):

- Lớp này chứa các quy trình nghiệp vụ và xử lý logic, đóng vai trò điều phối giữa View và Model để đáp ứng các yêu cầu của người dùng.
- **Chức năng:** Nhận yêu cầu từ View, xử lý các logic nghiệp vụ cần thiết, và gửi các yêu cầu tương tác với tầng Model để truy cập hoặc cập nhật dữ liệu khi cần.

3.3. Tầng mô hình dữ liệu (Model)

- **DAO (Data Access Object):** Lớp này chịu trách nhiệm truy cập và tương tác với cơ sở dữ liệu, thực hiện các thao tác truy vấn và cập nhật dữ liệu theo yêu cầu từ BLL.
- **DTO (Data Transfer Object):** Các đối tượng DTO được sử dụng để truyền dữ liệu giữa các lớp trong hệ thống, thường bao gồm các thuộc tính tương ứng với dữ liệu cần truyền.
- **Chức năng:** Quản lý và lưu trữ dữ liệu của ứng dụng, bảo đảm tính toàn vẹn và bảo mật dữ liệu khi thao tác.

3.4. Lý do chọn mô hình MVC:

- **Phân tách rõ ràng chức năng:** MVC chia ứng dụng thành ba thành phần riêng biệt — Model, View, và Controller — giúp việc quản lý mã nguồn trở nên dễ dàng hơn, cải thiện tính tổ chức và rõ ràng của mã.
- **Dễ bảo trì và mở rộng:** Bằng cách giữ dữ liệu (Model), giao diện (View), và logic xử lý (Controller) riêng biệt, kiến trúc MVC cho phép thực hiện thay đổi hoặc nâng cấp cho từng thành phần mà không ảnh hưởng đến các thành phần còn lại, giảm thiểu lỗi khi bảo trì hoặc nâng cấp.
- **Tái sử dụng mã nguồn:** Các thành phần của MVC có thể tái sử dụng trong nhiều dự án khác nhau hoặc giữa các chức năng khác nhau trong cùng ứng dụng, giúp tiết kiệm thời gian và công sức phát triển.

Tuy nhiên mô hình này vẫn có các nhược điểm không mong muốn:

- **Cần nhiều lớp và mã nguồn:** MVC yêu cầu phải có ít nhất ba lớp (Model, View, Controller), và đôi khi có thể dẫn đến việc mã nguồn trở nên phức tạp và dài dòng. Điều này có thể làm cho việc bảo trì trở nên khó khăn, nhất là với các dự án lớn.
- **Khó khăn trong việc quản lý các mối quan hệ giữa các lớp:** Mặc dù MVC giúp tách biệt các phần của ứng dụng, nhưng sự tương tác giữa các lớp Model, View và Controller vẫn có thể trở nên phức tạp khi có nhiều phụ thuộc lẫn nhau, dẫn đến việc khó khăn trong việc quản lý và kiểm soát các mối quan hệ này.

- **Khó khăn trong việc thay đổi giao diện:** Mặc dù MVC tách biệt giao diện (View) khỏi logic xử lý (Controller), nhưng đôi khi việc thay đổi giao diện lại đụng đến các phần khác của hệ thống, đặc biệt là khi giao diện yêu cầu các thay đổi lớn hoặc cần tích hợp với các hệ thống khác.
- **Hiệu suất có thể bị ảnh hưởng:** Khi có quá nhiều yêu cầu hoặc dữ liệu cần xử lý, sự phân tách giữa các lớp có thể làm giảm hiệu suất do phải thực hiện nhiều lớp trung gian để truyền và xử lý dữ liệu. Điều này đặc biệt rõ rệt trong các ứng dụng có khối lượng công việc lớn hoặc yêu cầu xử lý thời gian thực.

4. Góc nhìn logic:

4.1. Tổng quan

Góc nhìn logic tập trung vào cách tổ chức và tương tác giữa các thành phần bên trong hệ thống phần mềm để thực thi các chức năng nghiệp vụ mà không đi sâu vào chi tiết hoặc cách xử lý dữ liệu.

Trong hệ thống AireLeaf, khía cạnh này giúp hiểu rõ cách thức các gói và thành phần phối hợp để đáp ứng nhu cầu của người dùng và các yêu cầu kinh doanh.

4.2. Cấu trúc thiết kế quan trọng của các Packages / Components

4.2.1. Tầng trình diễn (View):

Mô tả: Tầng View (Giao diện người dùng) trong AireLeaf là nơi người dùng tương tác trực tiếp với hệ thống, cung cấp các công cụ để thao tác và truy cập các tính năng. Tầng View bao gồm các form nhập liệu, bảng hiển thị dữ liệu và nút chức năng. Nó cho phép người dùng thực hiện các thao tác như thêm, sửa, lọc thông tin và xuất báo cáo. Giao diện được thiết kế trực quan, dễ sử dụng, hỗ trợ tìm kiếm và thông báo tự động khi có sự kiện quan trọng.

Chức năng:

- Hiển thị thông tin về hợp đồng, mẫu kiểm định, và kết quả kiểm định trên giao diện trực quan, dễ sử dụng.
- Cho phép người dùng thực hiện các thao tác như thêm mới, chỉnh sửa, lọc thông tin.
- Tích hợp các công cụ tìm kiếm và chức năng lọc dữ liệu để hỗ trợ người dùng tìm được thông tin một cách nhanh chóng.

- Gửi thông báo tự động qua Email hoặc thông báo trên giao diện khi có lưu ý hoặc các hợp đồng sắp hết hạn.
- Cung cấp khả năng in/ xuất phiếu trả hàng khi người dùng có nhu cầu.

4.2.2. Tầng điều khiển (Controller):

Mô tả: Tầng Controller (Điều khiển) trong AireLeaf chịu trách nhiệm xử lý các yêu cầu từ người dùng và điều phối các hành động giữa tầng View và Model. Tầng này nhận các sự kiện từ giao diện người dùng, thực hiện các thao tác nghiệp vụ và cập nhật dữ liệu thông qua Model. Controller giúp duy trì tính logic của hệ thống và đảm bảo các thao tác diễn ra chính xác và hiệu quả. Nó cũng quản lý luồng điều hướng giữa các giao diện và các tính năng của hệ thống.

Chức năng:

- Thực hiện các quy trình nghiệp vụ phức tạp, chẳng hạn như quản lý quy trình kiểm định mẫu, phân công công việc cho nhân viên, và quản lý thời hạn hợp đồng.
- Tính toán và xử lý thông tin để tạo ra các báo cáo phân tích, đánh giá hiệu suất làm việc, và theo dõi tiến độ kiểm định.
- Đảm bảo các quy tắc nghiệp vụ được áp dụng chặt chẽ, giúp hệ thống hoạt động ổn định và tuân thủ các tiêu chuẩn kiểm định môi trường.

4.2.3. Tầng mô hình dữ liệu (Model)

Mô tả: Tầng Model quản lý và lưu trữ dữ liệu của hệ thống, đồng thời cung cấp các phương thức truy xuất, xử lý và lưu trữ dữ liệu cho tầng Controller. Model bao gồm các thành phần như Data Transfer Objects (DTO), Business Entities và Data Access Objects (DAO), giúp việc truy xuất và thao tác dữ liệu được tách biệt khỏi logic nghiệp vụ, hỗ trợ việc bảo trì và mở rộng hệ thống.

Thành phần chi tiết :

- Data Transfer Objects (DTO): DTO là các đối tượng được sử dụng để truyền tải dữ liệu giữa các thành phần trong ứng dụng (View và Controller). Chúng thường là các lớp đơn giản đại diện cho dữ liệu, như thông tin khách hàng, hợp đồng.
- Business Entities: Các đối tượng mô tả bản chất của dữ liệu trong hệ thống, chẳng hạn như KháchHang, HopDong, NhanVien. Những thực thể này thường bao gồm các phương

thức để kiểm tra tính hợp lệ của dữ liệu và các quy tắc nghiệp vụ.

- Data Access Objects (DAO): DAO cung cấp các phương thức để thao tác với cơ sở dữ liệu, bao gồm truy xuất, thêm mới, cập nhật và xóa dữ liệu. Những thao tác này tách biệt khỏi logic nghiệp vụ, giúp ứng dụng dễ bảo trì và mở rộng.

Chức năng:

- Thực hiện các truy vấn dữ liệu liên quan đến hợp đồng, nhân viên, phiếu trả hàng, khách hàng, và mẫu kiểm định.
- Đảm bảo tất cả các hoạt động với cơ sở dữ liệu diễn ra một cách an toàn, bảo mật và tối ưu hóa hiệu suất.
- Hỗ trợ các giao dịch dữ liệu phức tạp, đảm bảo tính toàn vẹn trong trường hợp xảy ra lỗi hoặc sự cố.