

ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG ĐIỆN - ĐIỆN TỬ



BÁO CÁO BÀI TẬP LỚN
KỸ THUẬT PHẦN MỀM ỨNG DỤNG

ĐỀ TÀI: QUẢN LÝ GIỐNG VÀ THỨC ĂN CHĂN NUÔI

TRÌNH BÀY BỞI NHÓM SINH VIÊN:

Members	MSSV & LỚP
Nguyễn Hữu Tuyên (Nhóm trưởng)	20233707 - ET1 02 K68
Nguyễn Thái Hoàng Long	20233508 - ET1 01 K68
Nguyễn Minh Quân	20233596 - ET1 01 K68
Hoàng Anh Dũng	20233344 - ET1 02 K68
Bùi Doãn Nhật	20233564 - ET1 02 K68

Giáo viên hướng dẫn: Ths. Vũ Song Tùng

MỤC LỤC

DANH MỤC HÌNH VẼ BẢNG BIỂU	3
LỜI NÓI ĐẦU	6
GIỚI THIỆU	7
CHƯƠNG 1: TỔNG QUAN VỀ CHƯƠNG TRÌNH QUẢN LÝ CSDL CHĂN NUÔI	9
1.1. Yêu cầu về dữ liệu	9
1.2. Yêu cầu về chức năng	10
1.3. Yêu cầu phi chức năng	10
1.4. Giới thiệu công cụ phần mềm sử dụng	11
1.4.1. PYTHON	11
1.4.2. FRAMEWORK: PYQT6	12
1.4.3. MySQL Server	12
1.5. Kết luận chung	12
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG	13
2.1. Usecase Diagram	13
2.2. Activities Diagram	18
2.3. Biểu đồ Sequence Diagram	22
2.4. Sơ đồ ERD	26
2.4.1. Tổng quan về sơ đồ ERD	26
2.4.2. Các sơ đồ ERD	28
2.5. Kết luận chung	30
CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG	31
3.1. Từ điển dữ liệu	31
3.2. Triển khai cơ sở dữ liệu	37
3.3. Triển khai cấu trúc hệ thống	41
3.3.1. Xây dựng lớp truy cập dữ liệu	41
3.3.2. Kiến trúc chương trình theo mô hình MVC	43
3.3.3. Một số thiết kế còn lại	54

3.4. <i>Giao diện hệ thống</i>	55
3.5. <i>Kết luận chương</i>	65
KẾT LUẬN	66
TÀI LIỆU THAM KHẢO	67

DANH MỤC HÌNH VẼ BẢNG BIỂU

HÌNH VẼ

Hình 2.1.1: Sơ đồ trường hợp sử dụng tổng quát	13
Hình 2.1.2: Sơ đồ trường hợp sử dụng cho Officer (Người dùng)	14
Hình 2.1.3: Sơ đồ trường hợp sử dụng cho Admin (Người quản trị)	15
Hình 2.1.4: Sơ đồ trường hợp sử dụng cho Viewer (Người xem)	16
Hình 2.2.1: Sơ đồ hoạt động quy trình quản lý truy cập hệ thống	18
Hình 2.2.2: Sơ đồ hoạt động quy trình quản lý cơ sở dữ liệu của hệ thống	19
Hình 2.2.3: Sơ đồ hoạt động quy trình quản trị hệ thống	20
Hình 2.3.1: Sơ đồ tuần tự quy trình truy cập hệ thống	22
Hình 2.3.2: Sơ đồ tuần tự quy trình đăng kí user mới	23
Hình 2.3.3: Sơ đồ tuần tự quy trình quản lý dữ liệu	24
Hình 2.4.2.1: Sơ đồ thực thể ERD hệ thống	28
Hình 2.4.2.2: Sơ đồ vật lý thiết kế thực thể ERD hệ thống	28
Hình 2.4.2.3: Sơ đồ thực thể ERD giống và chăn nuôi	29
Hình 2.4.2.4: Sơ đồ vật lý ERD giống và chăn nuôi	29
Hình 3.3.2.1: Sơ đồ MODEL - VIEW - CONTROLLER	43
Hình 3.4.1: Giao diện đăng nhập hệ thống	55
Hình 3.4.2: Giao diện chương trình	55
Hình 3.4.3: Thông tin tài khoản	56
Hình 3.4.4: Quản lý tài khoản người dùng	56
Hình 3.4.5: Quản lý nhóm người dùng	57
Hình 3.4.6: Quản lý CSDL đơn vị hành chính	57
Hình 3.4.7: Phân quyền người dùng	58
Hình 3.4.8: Đổi mật khẩu tài khoản	58
Hình 3.4.9: Quản lý CSDL cơ sở sản xuất thức ăn chăn nuôi	59
Hình 3.4.10: Quản lý CSDL thức ăn	59
Hình 3.4.11: Quản lý CSDL gen	60
Hình 3.4.12: Quản lý CSDL nguyên liệu và chất cấm	60
Hình 3.4.13: Quản lý CSDL giống vật nuôi	61
Hình 3.4.14: Chức năng tìm kiếm	61

Hình 3.4.15: Lịch sử hệ thống	62
Hình 3.4.16: Thống kê báo cáo	62
Hình 3.4.17: File báo cáo được xuất	63
Hình 3.4.18: Quên mật khẩu	64

BẢNG

Bảng kế hoạch chi tiết	8
Bảng 1.1.1: Yêu cầu về dữ liệu	9
Bảng 2.1.1: Danh mục sơ đồ trường hợp sử dụng	13
Bảng 2.2.1: Danh mục sơ đồ trường hợp sử dụng	18
Bảng 2.3.1: Danh mục sơ đồ trường hợp sử dụng	22
Bảng 3.1.1: Bảng hồ sơ	31
Bảng 3.1.2: Bảng Đơn vị Hành chính	31
Bảng 3.1.3: Bảng Tài khoản	32
Bảng 3.1.4: Bảng Lịch sử hoạt động	32
Bảng 3.1.5: Bảng quyền	33
Bảng 3.1.6: Bảng Nhóm	33
Bảng 3.1.7: Bảng Gen	34
Bảng 3.1.8: Bảng giống loài	34
Bảng 3.1.9: Bảng cơ sở	35
Bảng 3.1.10: Bảng thức ăn	36
Bảng 3.1.11: Bảng thành phần	36

LỜI NÓI ĐẦU

Trong bối cảnh nông nghiệp hiện đại và hội nhập quốc tế, ngành chăn nuôi đang chuyển mình mạnh mẽ theo hướng công nghiệp hóa, hiện đại hóa. Việc quản lý hiệu quả nguồn giống và thức ăn chăn nuôi không chỉ là yếu tố then chốt quyết định năng suất, chất lượng thương phẩm mà còn ảnh hưởng trực tiếp đến hiệu quả kinh tế và an toàn vệ sinh thực phẩm. Tuy nhiên, với quy mô chăn nuôi ngày càng mở rộng cùng những yêu cầu khắt khe về truy xuất nguồn gốc, các phương pháp quản lý thủ công truyền thống đã bộc lộ nhiều hạn chế, gây khó khăn trong việc kiểm soát chất lượng, theo dõi định mức dinh dưỡng và tối ưu hóa chi phí. Những thách thức này đặt ra yêu cầu cấp thiết về việc ứng dụng công nghệ thông tin để chuẩn hóa quy trình và hỗ trợ ra quyết định chính xác.

Nhận thức được thực trạng và nhu cầu cấp bách trên, nhóm chúng em đã quyết định nghiên cứu và phát triển ứng dụng quản lý cơ sở dữ liệu về giống và thức ăn chăn nuôi. Ứng dụng được thiết kế với các chức năng chuyên sâu như: quản lý hồ sơ lý lịch vật nuôi, theo dõi quá trình nhập - xuất - tồn kho thức ăn, giám sát khẩu phần dinh dưỡng và phân tích hiệu quả chăn nuôi một cách trực quan. Không chỉ dừng lại ở việc cung cấp thông tin chính xác và kịp thời, ứng dụng còn hướng đến việc hỗ trợ các chủ trang trại, doanh nghiệp trong việc xây dựng kế hoạch chăn nuôi khoa học. Qua đó, góp phần giảm thiểu rủi ro dịch bệnh, tiết kiệm chi phí và nâng cao khả năng cạnh tranh trên thị trường.

Mặc dù đã cố gắng và nỗ lực hết sức, nhưng bài báo cáo và sản phẩm của nhóm chúng em chắc chắn không tránh khỏi những hạn chế và thiếu sót. Chúng em rất mong nhận được sự đóng góp ý kiến và đánh giá từ thầy cô và các bạn để hoàn thiện hơn nữa.

Cuối cùng, nhóm chúng em xin gửi lời cảm ơn chân thành đến ThS. Vũ Song Tùng vì sự hướng dẫn tận tình cũng như các nguồn tài liệu quý báu đã giúp đỡ nhóm hoàn thành đề tài này.

GIỚI THIỆU

I Mở đầu

Ngành chăn nuôi ở Việt Nam là một bộ phận quan trọng cấu thành của nền nông nghiệp quốc gia, giữ vai trò then chốt trong việc cung cấp thực phẩm, đảm bảo an ninh dinh dưỡng và ổn định đời sống kinh tế – xã hội. Tình hình quản lý giống vật nuôi và thức ăn chăn nuôi phản ánh thực trạng năng suất, chất lượng sản phẩm, an toàn dịch bệnh cũng như nhu cầu tiêu dùng và xuất khẩu tại Việt Nam. Vấn đề đặt ra là làm thế nào để quản lý nguồn gen giống, chất lượng và nguồn gốc thức ăn chăn nuôi một cách khoa học, thuận tiện, nhanh chóng, đồng thời kiểm soát chặt chẽ an toàn vệ sinh thực phẩm và tiết kiệm được nhân lực, tài lực. Do đó, ý tưởng xây dựng **“Cơ sở dữ liệu (CSDL) quản lý giống và thức ăn chăn nuôi”** ra đời.

II Mục đích

Hiện nay, thị trường giống vật nuôi và thức ăn chăn nuôi chiếm tỉ trọng lớn về quy mô và giá trị trong cơ cấu ngành nông nghiệp. Tuy nhiên, công tác quản lý dữ liệu liên quan đến nguồn gốc giống, thành phần và chất lượng thức ăn còn nhiều bất cập, hạn chế và thiếu đồng bộ, dẫn đến việc kiểm soát, giám sát, truy xuất nguồn gốc và đảm bảo chất lượng chưa đạt hiệu quả cao.

Nhận thấy thực trạng trong công tác quản lý giống, thức ăn chăn nuôi và vai trò đặc biệt quan trọng của chúng đối với năng suất và sức khỏe cộng đồng, việc xây dựng một hệ thống CSDL quản lý nhằm các mục đích chính sau:

- Giúp công tác quản lý danh mục giống và thức ăn chăn nuôi được tối ưu, khoa học và hiệu quả, tiết kiệm thời gian và công sức cho cơ quan quản lý và doanh nghiệp.
- Giao diện thân thiện, dễ sử dụng đối với người quản lý, cán bộ thú y và các cơ sở sản xuất kinh doanh.
- Dễ dàng quản lý, theo dõi tình trạng lưu hành, nguồn gốc xuất xứ của các loại giống vật nuôi, các loại thức ăn (đạm đặc, hỗn hợp, nguyên liệu) và các cơ sở sản xuất, kinh doanh liên quan.
- Thuận tiện trong việc quản lý, tra cứu các văn bản pháp luật, quy chuẩn kỹ thuật quốc gia (QCVN) và tiêu chuẩn cơ sở trong lĩnh vực chăn nuôi.
- Góp phần nâng cao hiệu quả chăn nuôi, giảm thiểu rủi ro dịch bệnh từ nguồn thức ăn và con giống kém chất lượng, từ đó thúc đẩy phát triển kinh tế – xã hội.

Các chức năng chính hướng đến gồm: Quản lý CSDL về danh mục giống và thức ăn được phép lưu hành, thống kê – báo cáo tình hình sản xuất, nhập khẩu và quản lý văn bản pháp luật liên quan. Trong tương lai, hệ thống có thể được mở rộng và phát triển thêm các chức năng như cảnh báo dịch bệnh hoặc truy xuất nguồn gốc chuỗi cung ứng.

III Nhiệm vụ

Giống và thức ăn chăn nuôi đóng vai trò quyết định đến năng suất, chất lượng sản phẩm chăn nuôi và đảm bảo an toàn thực phẩm cho người tiêu dùng. Để nâng cao hiệu quả quản lý nhà nước, minh bạch hóa thị trường và hỗ trợ người chăn nuôi tiếp cận nguồn vật tư chất lượng, cần có một hệ thống quản lý dữ liệu tập trung, chính xác và dễ sử dụng.

Vì vậy, nhiệm vụ xây dựng Cơ sở dữ liệu (CSDL) quản lý giống và thức ăn chăn nuôi nhằm hỗ trợ theo dõi hiện trạng sản xuất, kinh doanh, quản lý chất lượng (kiểm nghiệm, khảo nghiệm), cập nhật thông tin thị trường và tuân thủ các quy định của Luật Chăn nuôi là hết sức cần thiết.

IV Xác lập dự án

Bảng kế hoạch chi tiết

TT	Tên công việc	Bắt đầu	Kết thúc	Kết quả	Trạng thái
1	Xác định đề tài	31/10	31/10	Quản lý giống và thức ăn chăn nuôi	Hoàn thành
2	Tìm hiểu yêu cầu	1/11	5/11	Xác định hướng đi dự án	Hoàn thành
3	Mô tả yêu cầu	6/11	8/11	Xác định các yêu cầu đặt ra	Hoàn thành
4	Phân tích hệ thống	9/11	15/11	Mô tả yêu cầu chức năng & phi chức năng	Hoàn thành
5	Thiết kế CSDL	16/11	20/11	Xây dựng CSDL MySQL	Hoàn thành
6	Thiết kế hệ thống	21/11	1/1	Viết chương trình bằng Python	Hoàn thành
7	Viết báo cáo	2/1	17/1	Tổng hợp báo cáo cuối kì	Hoàn thành

CHƯƠNG 1: TỔNG QUAN VỀ CHƯƠNG TRÌNH QUẢN LÝ CSDL GIỐNG VÀ THỨC ĂN CHĂN NUÔI

Chương này đặt vấn đề cần giải quyết khi đề xuất hệ thống quản lý và phân tích các chức năng cần thiết để giải quyết vấn đề đặt ra. Đồng thời mô tả sơ lược các công cụ hỗ trợ của bên thứ ba được sử dụng trong đề tài.

1.1. Yêu cầu về dữ liệu

Bảng 1.1.1 - Yêu cầu về dữ liệu

Dữ liệu	Mô tả
Đơn vị hành chính cấp huyện	Lưu thông tin danh mục đơn vị hành chính cấp huyện phục vụ quản lý, phân quyền theo địa bàn.
Đơn vị hành chính cấp xã	Lưu thông tin danh mục đơn vị hành chính cấp xã, liên kết với cấp huyện.
Người dùng	Quản lý thông tin tài khoản người dùng hệ thống.
Nhóm người dùng	Gom nhóm người dùng để phân quyền theo vai trò/chức năng.
Quyền	Định nghĩa các quyền thao tác trên từng chức năng của hệ thống.
Phân quyền nhóm – quyền	Gán quyền cho nhóm người dùng.
Phân quyền người dùng – quyền	Gán quyền trực tiếp cho từng người dùng (ghi đè hoặc bổ sung quyền nhóm).
Menu hệ thống	Quản lý hiển thị menu theo quyền người dùng.
Lịch sử truy cập người dùng	Ghi nhận lịch sử đăng nhập, truy cập hệ thống.
Lịch sử tác động hệ thống	Theo dõi các thao tác làm thay đổi dữ liệu hệ thống.
Báo cáo – thống kê	Phục vụ thống kê số lượng người dùng, tần suất truy cập, mức độ sử dụng hệ thống.
Tổ chức, cá nhân (sản xuất – kinh doanh – sở hữu giống)	Quản lý thông tin các đơn vị tham gia sản xuất, mua bán, phối giống vật nuôi.
Cơ sở khảo nghiệm giống	Lưu thông tin các cơ sở được phép khảo nghiệm dòng, giống vật nuôi.
Giống vật nuôi đặc thù	Quản lý danh mục giống vật nuôi theo quy định đặc biệt của Nhà nước.
Tổ chức, cá nhân về nguồn gen	Quản lý các đơn vị tham gia thu thập, bảo tồn và phát triển nguồn gen vật nuôi.
Cơ sở sản xuất, mua bán thức ăn chăn nuôi	Quản lý thông tin các cơ sở sản xuất, kinh doanh thức ăn chăn nuôi.
Cơ sở khảo nghiệm thức ăn chăn nuôi	Lưu thông tin các cơ sở khảo nghiệm thức ăn chăn nuôi.
Danh mục hóa chất và nguyên liệu	Quản lý danh mục phục vụ kiểm soát an toàn trong thức ăn chăn nuôi.

1.2. Yêu cầu về chức năng

- Tài khoản: Thêm, sửa, xóa, tìm kiếm, đăng nhập, đăng xuất, quên mật khẩu, đổi mật khẩu, quản lý hồ sơ cá nhân.
- Quyền hạn: Thêm, sửa, xóa, tìm kiếm nhóm người dùng, phân quyền chi tiết cho nhóm và từng người dùng.
- Đơn vị hành chính: Thêm, sửa, xóa, tìm kiếm danh mục hành chính cấp Huyện và cấp Xã.
- Giám sát & Báo cáo: Tra cứu lịch sử truy cập, lịch sử tác động hệ thống, xem báo cáo thống kê tổng hợp.
- Cơ sở sản xuất giống: Thêm, sửa, xóa, tìm kiếm thông tin cơ sở sản xuất con giống, tinh, phôi, ấu trùng và ấp trứng.
- Kinh doanh & Dịch vụ giống: Quản lý thông tin, tìm kiếm tổ chức mua bán giống, chủ đực giống và cơ sở khảo nghiệm.
- Danh mục giống đặc biệt: Quản lý và tìm kiếm danh sách giống vật nuôi cần bảo tồn, giống cấm xuất khẩu.
- Nguồn gen vật nuôi: Quản lý danh sách, tìm kiếm đơn vị thu thập, bảo tồn, khai thác và phát triển nguồn gen.
- Cơ sở thức ăn chăn nuôi: Thêm, sửa, xóa, tìm kiếm cơ sở sản xuất (thương mại/có giấy chứng nhận) và cơ sở kinh doanh.
- An toàn & Chất lượng thức ăn: Quản lý danh mục chất cấm, nguyên liệu được phép sử dụng và cơ sở khảo nghiệm thức ăn.

1.3. Yêu cầu phi chức năng

Dựa trên đặc tả hệ thống và các tiêu chuẩn kỹ thuật, các yêu cầu phi chức năng được xác định chi tiết như sau:

- Hiệu suất (Performance):

- + Hệ thống phải đảm bảo độ trễ xử lý truy vấn dữ liệu dưới 3 giây ($< 3s$).
- + Hỗ trợ khả năng chịu tải cho trên 100 người dùng (> 100 users) truy cập đồng thời mà vẫn đảm bảo hoạt động nhanh chóng, mượt mà.

- Khả năng mở rộng (Scalability):

- + Kiến trúc hệ thống cho phép dễ dàng thêm mới các danh mục dữ liệu (giống vật nuôi, nguồn gen,...) khi có yêu cầu phát sinh.
- + Có khả năng tích hợp hệ thống thông tin địa lý (GIS/Bản đồ) để hiển thị dữ liệu phân bố (ví dụ: phân bố cơ sở chăn nuôi, vùng dịch tễ).

- **Bảo mật (Security):**

- + *Phân quyền*: Chỉ người dùng được cấp quyền (authorized users) mới có thể thực hiện thêm, sửa dữ liệu.
- + *An toàn dữ liệu*: Dữ liệu phải được mã hóa và có cơ chế sao lưu định kỳ (hàng ngày/hàng tuần) để phòng tránh rủi ro.
- + *Xác thực*: Bảo vệ chống truy nhập trái phép bằng cơ chế xác thực hai yếu tố (2FA).

- **Khả năng sử dụng (Usability):**

- + Giao diện thiết kế thân thiện, trực quan.
- + Hỗ trợ đa ngôn ngữ: cả Tiếng Việt và Tiếng Anh.
- + Cung cấp tài liệu và chức năng hướng dẫn sử dụng rõ ràng, chi tiết cho từng chức năng hệ thống.

- **Tích hợp và Sẵn sàng (Integration & Availability):**

- + Có khả năng tích hợp, kết nối với các hệ thống hiện có hoặc hệ thống GIS.
- + Đảm bảo tính sẵn sàng cao, hoạt động liên tục với tỷ lệ uptime đạt 99,99%.

- **Báo cáo (Reporting):**

- + Hỗ trợ tính năng tự động tạo báo cáo định kỳ (hàng ngày, hàng tuần) cho từng danh mục quản lý.
- + Hỗ trợ xuất dữ liệu báo cáo dưới các định dạng phổ biến như PDF và Excel.

- **Tính toàn vẹn dữ liệu (Data Integrity):**

- + Hệ thống có cơ chế kiểm tra dữ liệu chặt chẽ.
- + Tự động phát hiện và đưa ra cảnh báo đối với các dữ liệu không hợp lệ hoặc sai định dạng.

1.4. Giới thiệu công cụ phần mềm sử dụng

1.4.1. PYTHON

Python do Guido van Rossum sáng lập với ba tiêu chí nổi bật: dễ học, dễ đọc, dễ nhớ. Ngôn ngữ được dùng để phát triển website cũng như các ứng dụng khác vì cú pháp đơn giản, hoàn toàn tạo kiểu động, hình thức bắt mắt, cấu trúc bài bản, không những vậy mà Python còn cho phép viết mã lệnh với số lần gõ rồi thiếu. Dù là tay ngang mới vào nghề, Python vẫn cho phép bạn tiếp cận một cách dễ dàng.

Python là ngôn ngữ lập trình mã nguồn mở nổi bật với cú pháp đơn giản, dễ học, trở thành lựa chọn hàng đầu cho cả người mới bắt đầu lẫn các lập trình viên chuyên nghiệp. Điểm mạnh của Python nằm ở tính linh hoạt cao, có thể hoạt động mượt mà trên đa nền tảng (Windows, macOS, Linux) và khả năng tự động quản lý bộ nhớ thông minh. Ngôn ngữ này sở hữu kho thư viện khổng lồ, hỗ trợ đắc lực cho nhiều lĩnh vực phức tạp như phát triển website, phân tích dữ liệu, trí tuệ nhân tạo và học máy. Bên cạnh đó,

với khả năng kết hợp cùng C/C++, hỗ trợ lập trình hướng đối tượng và cộng đồng phát triển lớn mạnh, Python vừa là công cụ giáo dục hiệu quả, vừa là nền tảng mạnh mẽ để xây dựng các ứng dụng thực tế và tạo bản mẫu phần mềm nhanh chóng.

Bên cạnh đó, với khả năng kết hợp cùng C/C++, hỗ trợ lập trình hướng đối tượng và cộng đồng phát triển lớn mạnh, Python vừa là công cụ giáo dục hiệu quả, vừa là nền tảng mạnh mẽ để xây dựng các ứng dụng thực tế và tạo bản mẫu phần mềm nhanh chóng.

1.4.2. FRAMEWORK: PYQT6

PyQt là một công cụ tiện ích xây dựng GUI. PyQt cho phép bạn sử dụng Qt GUI, một cross-platform mạnh mẽ và nổi tiếng để xây dựng ứng dụng GUI trên Desktop, hệ thống nhúng và Mobile. Hỗ trợ cho các platform bao gồm: Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS và một số platform khác. PyQt là một Python Interface của Qt, là sự kết hợp giữa Python và Qt library, bao gồm các thành phần giao diện điều khiển (widgets, graphical control elements). PyQt API bao gồm một số lượng lớn các Classes và Functions hỗ trợ cho việc thiết kế, xây dựng và xử lý các nghiệp vụ của người dùng khi sử dụng các phần mềm chức năng. Phiên bản mới nhất của PyQt là phiên bản PyQt6.

PyQt6 là bộ thư viện mạnh mẽ cho phép lập trình viên Python xây dựng các ứng dụng có giao diện đồ họa (GUI) chuyên nghiệp và hoạt động mượt mà trên đa nền tảng như Windows, macOS và Linux. Là phiên bản kế nhiệm dựa trên Qt Framework 6, PyQt6 mang đến nhiều cải tiến hiện đại như hỗ trợ tốt màn hình độ phân giải cao (High DPI) và cấu trúc mã nguồn chặt chẽ, an toàn hơn. Với kho tiện ích (widget) phong phú, cơ chế xử lý sự kiện linh hoạt (Signals và Slots) và khả năng tích hợp công cụ thiết kế kéo thả Qt Designer, đây là giải pháp hàng đầu để phát triển phần mềm desktop, tuy nhiên người dùng cần lưu ý tuân thủ giấy phép GPL v3 đối với các dự án thương mại.

1.4.3. MySQL Server

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được sử dụng rộng rãi để lưu trữ, quản lý và truy xuất dữ liệu. MySQL hoàn toàn miễn phí nằm trong nhóm LAMP (Linux – Apache – MySQL – PHP).

Vì hệ quản trị cơ sở dữ liệu MySQL ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh và Mysql cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL chính vì thế nên MySQL được sử dụng và hỗ trợ của những lập trình viên yêu thích mã nguồn mở.

1.5. Kết luận chung

Như vậy, trong chương này đã trình bày một cách tổng quan về đề tài Cơ sở dữ liệu quản lý giống và chăn nuôi, bao gồm các công cụ, ngôn ngữ lập trình được sử dụng cùng với các yêu cầu chức năng và yêu cầu phi chức năng của hệ thống.

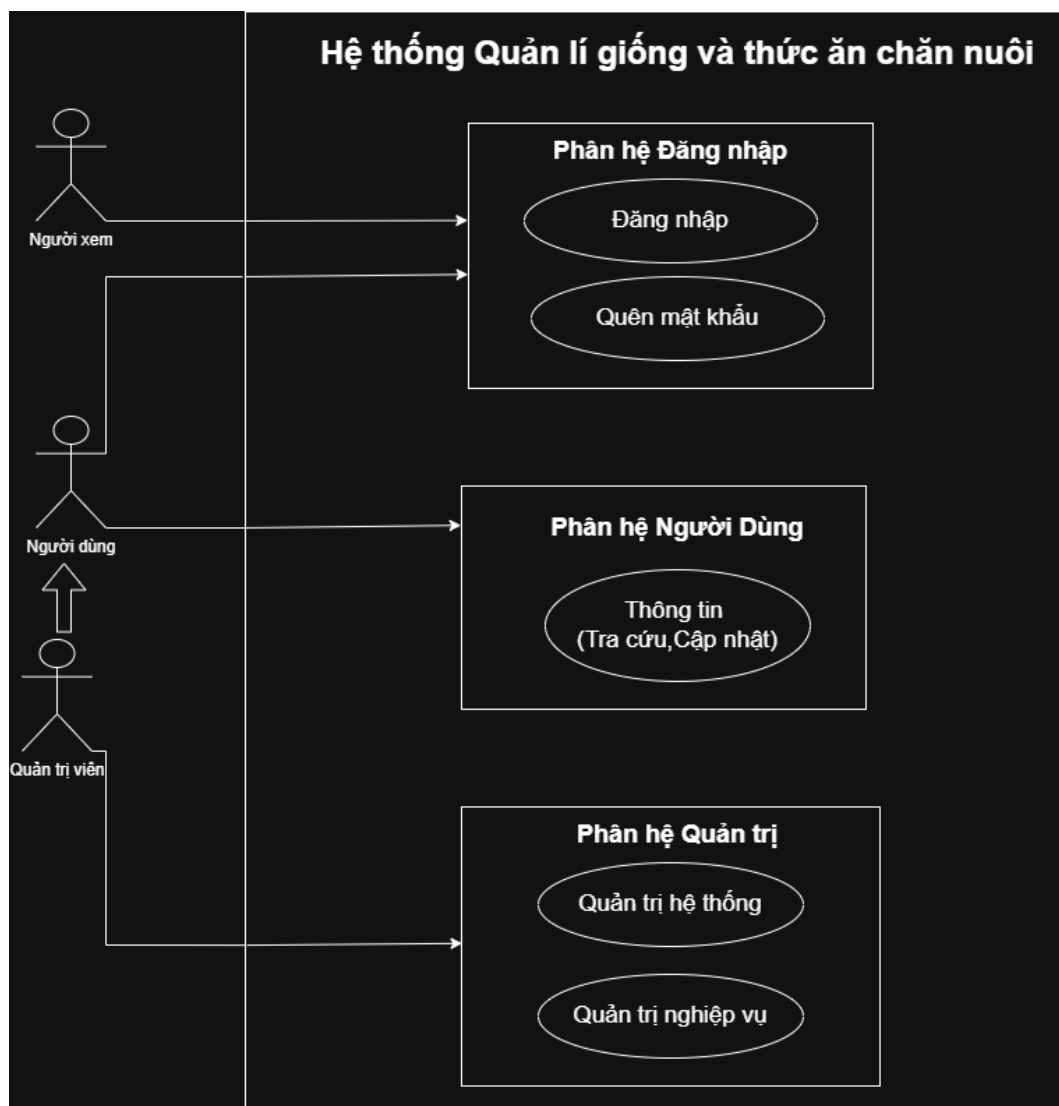
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Chương này phân tích hệ thống qua bằng cách vẽ các sơ đồ.

2.1. Usecase Diagram

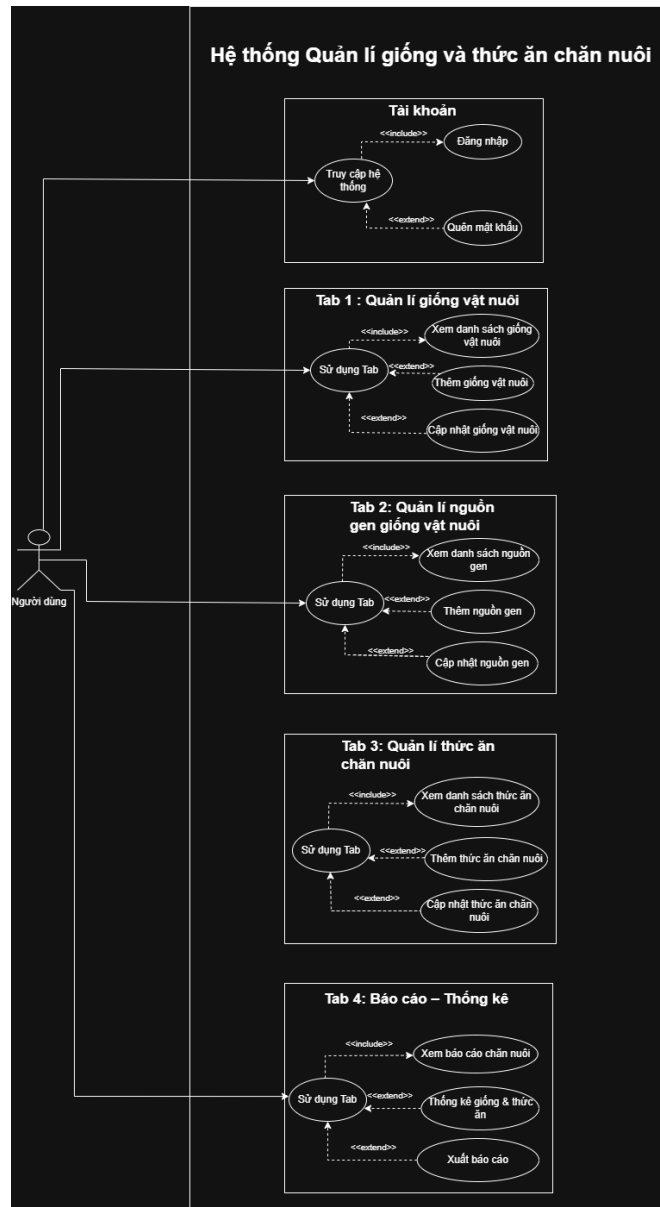
Bảng 2.1.1: Danh mục sơ đồ trường hợp sử dụng

STT	Tên sơ đồ	Tác nhân chính
1	Sơ đồ tổng quát (Hình 2.1.1)	Người dùng, Quản trị viên, Người xem
2	Phân rã Người dùng (Hình 2.1.2)	Người dùng
3	Phân rã Quản trị viên (Hình 2.1.3)	Quản trị viên
4	Phân rã Người xem (Hình 2.1.4)	Người xem



Hình 2.1.1: Sơ đồ trường hợp sử dụng tổng quát

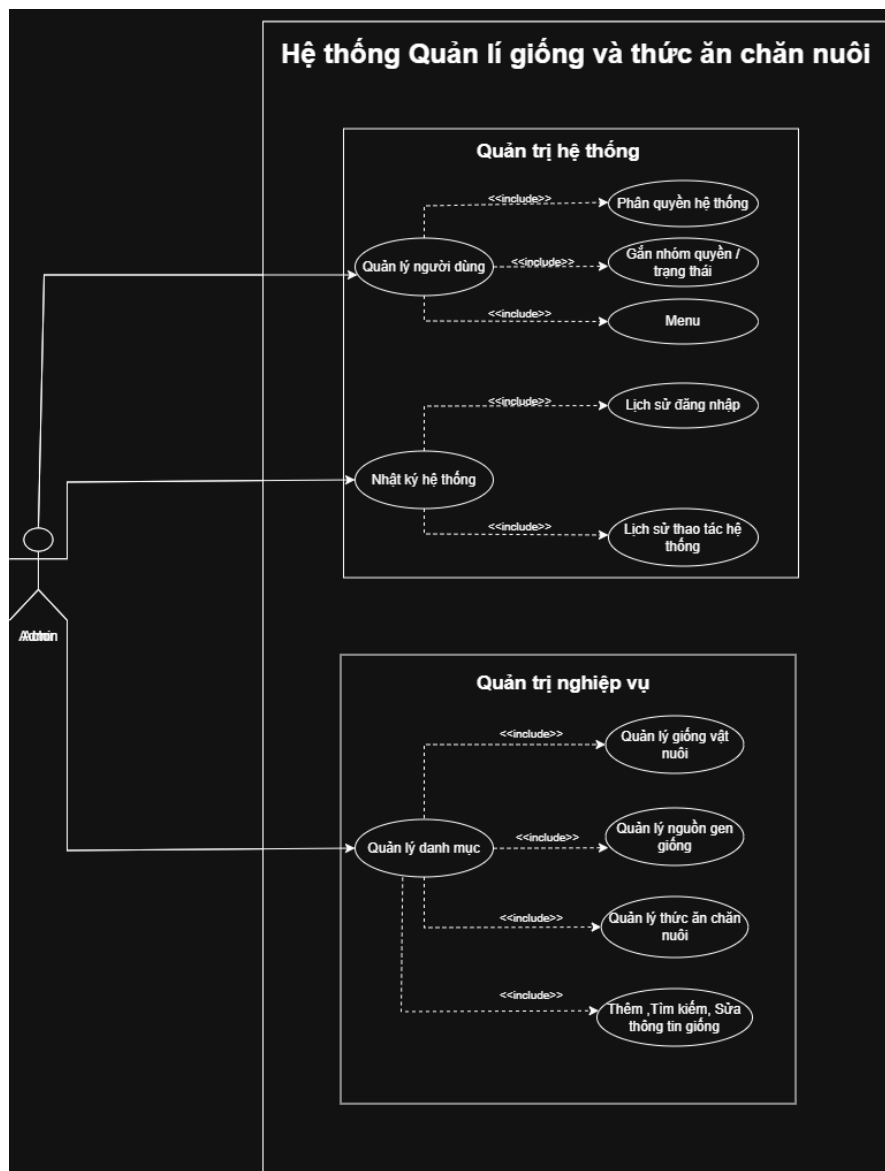
Biểu đồ Use Case tổng quan phân chia ranh giới hoạt động của Hệ thống Quản lý giống và thức ăn chăn nuôi bằng cách xác định các nhóm chức năng chính và vai trò của các tác nhân liên quan bao gồm Viewer, Officer và Admin. Cấu trúc hệ thống được chia thành 3 phân hệ nhiệm vụ chính: đầu tiên là phân hệ Đăng nhập phục vụ việc định danh và cấp quyền truy cập ứng dụng thông qua các chức năng đăng nhập và đăng kí; tiếp theo là phân hệ Người dùng tập trung vào việc khai thác và xử lý dữ liệu, cho phép tác nhân Officer thực hiện tra cứu và cập nhật các thông tin nghiệp vụ; cuối cùng là phân hệ Quản trị dành riêng cho tác nhân Admin (người có quyền hạn cao nhất, kế thừa từ Officer) để thực hiện các công tác điều hành thông qua chức năng quản trị hệ thống và quản trị nghiệp vụ.



Hình 2.1.2: Sơ đồ trường hợp sử dụng cho Officer (Người dùng)

Biểu đồ trường hợp sử dụng minh họa các tương tác giữa tác nhân Officer (Cán bộ) với Hệ thống Quản lý giống và thức ăn chăn nuôi. Các chức năng của hệ thống được tổ chức khoa học thành 5 nhóm chính, tương ứng với quy trình nghiệp vụ và giao diện người dùng. Đầu tiên là nhóm chức năng Tài khoản, cho phép người dùng truy cập vào hệ thống. Hành động truy cập này bao hàm chức năng Đăng

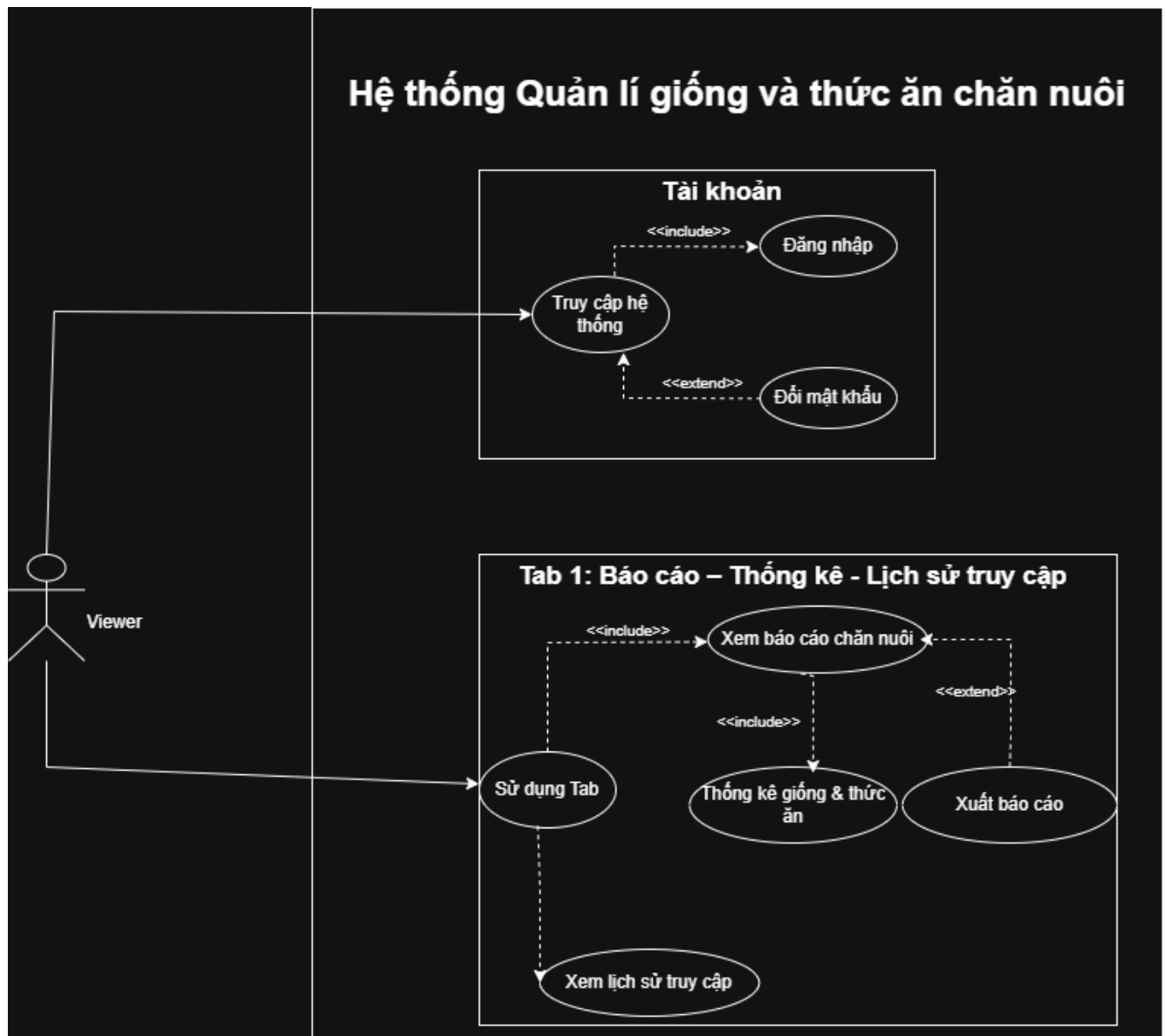
nhập bắt buộc, đồng thời có thể mở rộng sang các tác vụ Đăng ký hoặc Đổi mật khẩu tùy theo nhu cầu của người dùng. Tiếp theo là ba nhóm chức năng nghiệp vụ cốt lõi được phân chia theo các Tab gồm Quản lý giống vật nuôi, Quản lý nguồn gen giống vật nuôi và Quản lý thức ăn chăn nuôi. Cấu trúc của các nhóm này được thiết kế đồng nhất để đảm bảo tính thuận tiện, trong đó khi người dùng sử dụng Tab, hệ thống mặc định cung cấp chức năng xem danh sách tương ứng. Ngoài ra, từ chức năng nền tảng này, người dùng có thể thực hiện các hành động mở rộng như thêm mới hoặc cập nhật dữ liệu cho giống vật nuôi, nguồn gen và thức ăn chăn nuôi. Cuối cùng là nhóm chức năng Báo cáo – Thống kê, nơi người dùng có thể xem các báo cáo tổng quan về chăn nuôi. Chức năng này cũng hỗ trợ các tác vụ mở rộng quan trọng là thực hiện thống kê giống và thức ăn chi tiết, cũng như xuất báo cáo ra các định dạng lưu trữ bên ngoài phục vụ công tác quản lý.



Hình 2.1.3: Sơ đồ trường hợp sử dụng cho Admin (Người quản trị)

Đối với biểu đồ trường hợp sử dụng của Quản trị viên, trọng tâm được đặt vào các công cụ quản lý nghiệp vụ nền tảng và kiểm soát an ninh hệ thống. Trong phân hệ Quản trị nghiệp vụ, Quản trị viên nắm giữ vai trò thiết lập và duy trì các danh mục dữ liệu cốt lõi thông qua các chức năng quản lý giống vật

nuôi, quản lý nguồn gen giống và quản lý thức ăn chăn nuôi. Các hoạt động này bao hàm đầy đủ các thao tác nghiệp vụ cần thiết như thêm mới, tìm kiếm và sửa đổi thông tin giống để đảm bảo tính chính xác của dữ liệu. Bên cạnh việc điều phối dữ liệu nghiệp vụ, sơ đồ còn mô tả chi tiết các nhiệm vụ thuộc nhóm Quản trị hệ thống, nơi Admin thực hiện quản lý người dùng thông qua việc phân quyền hệ thống, gán nhóm quyền hoặc trạng thái hoạt động và thiết lập cấu trúc Menu. Ngoài ra, Quản trị viên còn thực hiện chức năng giám sát chặt chẽ thông qua việc truy xuất nhật ký hệ thống, cho phép theo dõi chi tiết lịch sử đăng nhập cũng như toàn bộ lịch sử thao tác trên hệ thống để đảm bảo an toàn và minh bạch thông tin.



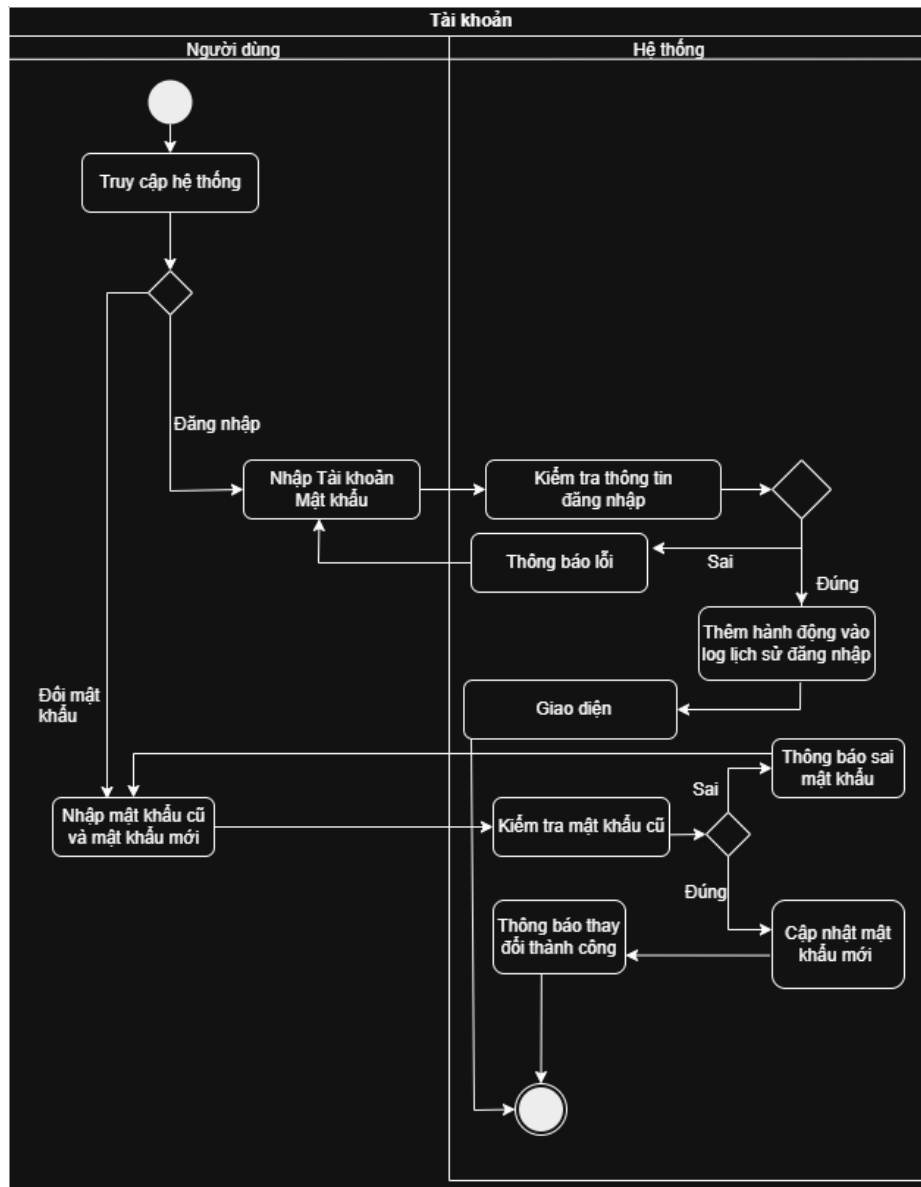
Hình 2.1.4: Sơ đồ trường hợp sử dụng cho Viewer (Người xem)

Đối với tác nhân Viewer (Người xem), phạm vi hoạt động trong hệ thống tập trung chủ yếu vào việc tra cứu và theo dõi số liệu tổng quan. Ở nhóm chức năng Tài khoản, người dùng truy cập hệ thống thông qua bước Đăng nhập bắt buộc, đi kèm các tùy chọn bổ sung là Đăng ký hoặc Đổi mật khẩu. Về mặt nghiệp vụ, Viewer chỉ tương tác với Tab 1: Báo cáo – Thống kê - Lịch sử. Tại đây, khi sử dụng tab, hệ thống mặc định cung cấp chức năng Xem báo cáo chăn nuôi. Ngoài ra, người xem có thể thực hiện các tác vụ mở rộng để khai thác dữ liệu chi tiết hơn như Thống kê giống và thức ăn, tra cứu Lịch sử truy cập cá nhân hoặc Xuất báo cáo ra file lưu trữ.

2.2. Activities Diagram

Bảng 2.2.1: Danh mục sơ đồ trường hợp sử dụng

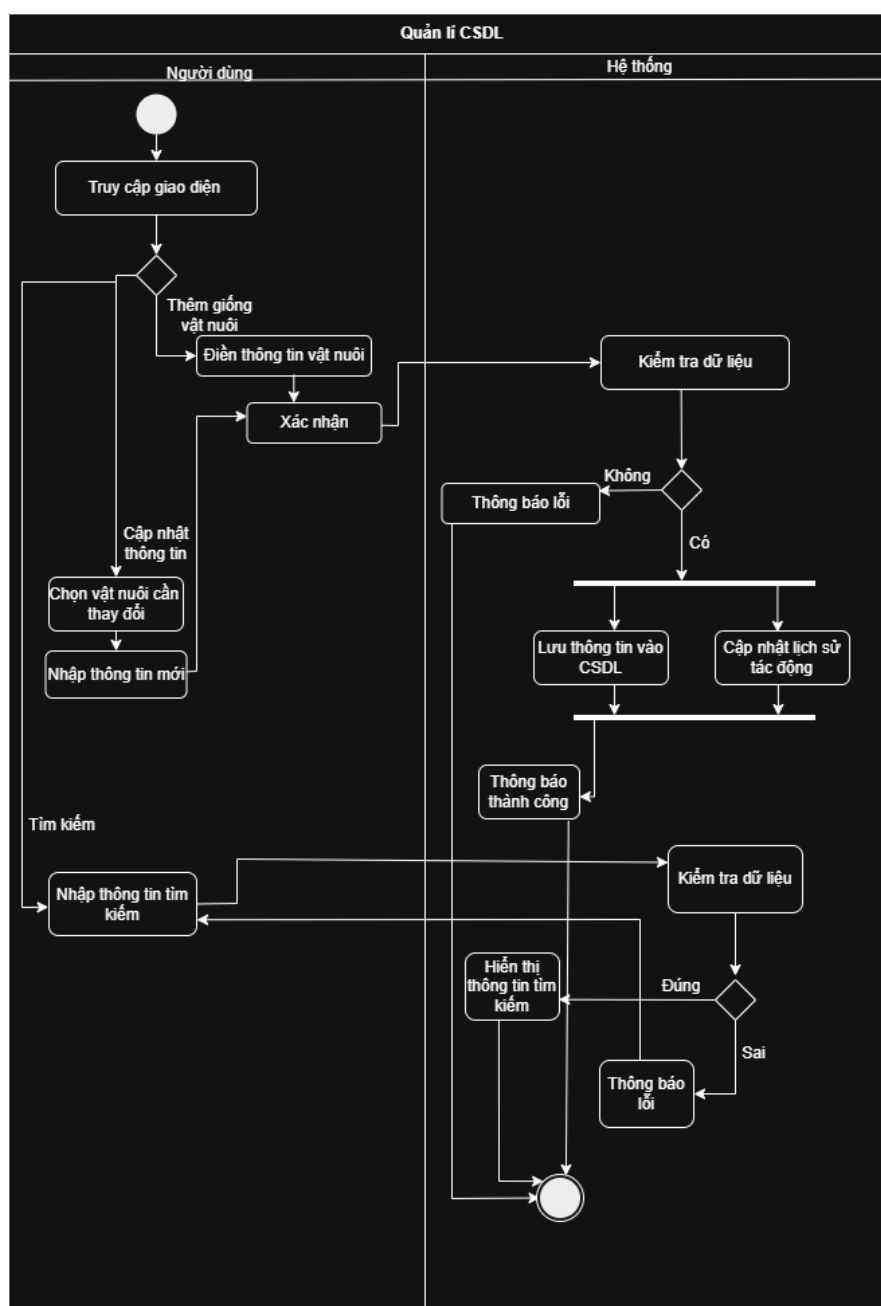
STT	Tên sơ đồ	Tác nhân chính
1	Sơ đồ hoạt động quy trình quản lý hệ thống (Hình 2.21)	Người dùng, Quản trị viên, Người xem
2	Sơ đồ quản lý cơ sở dữ liệu (Hình 2.2.2)	Người dùng, Quản trị viên
3	Sơ đồ quản trị hệ thống (Hình 2.2.3)	Quản trị viên



Hình 2.2.1: Sơ đồ hoạt động quy trình quản lý truy cập hệ thống

Luồng xử lý bắt đầu khi người dùng truy cập hệ thống để thực hiện các tác vụ quản lý tài khoản bao gồm đăng ký, đăng nhập hoặc thay đổi thông tin bảo mật. Sau khi tiếp nhận dữ liệu đầu vào như tên tài khoản và mật khẩu, hệ thống thực hiện kiểm tra tính hợp lệ và xác thực tính duy nhất của dữ liệu, đảm bảo tên đăng nhập không bị trùng lặp trong quá trình khởi tạo tài khoản mới.

Điểm đặc biệt của quy trình nằm ở cơ chế tự động hóa việc thiết lập trạng thái và lưu vết hoạt động. Cụ thể, đối với tài khoản mới đăng ký hợp lệ, hệ thống sẽ tự động gán nhóm quyền "User" và kích hoạt trạng thái "Active". Trong quá trình đăng nhập, sau khi đối soát thành công thông tin xác thực, hệ thống sẽ thực hiện ghi nhận dữ liệu vào log lịch sử đăng nhập trước khi cấp quyền truy cập. Đối với chức năng đổi mật khẩu, hệ thống yêu cầu xác thực mật khẩu cũ trước khi cập nhật giá trị mới, đảm bảo tính an toàn cho tài khoản. Quy trình kết thúc bằng việc phản hồi thông báo trạng thái cụ thể cho người dùng và điều hướng truy cập vào giao diện chính của hệ thống.

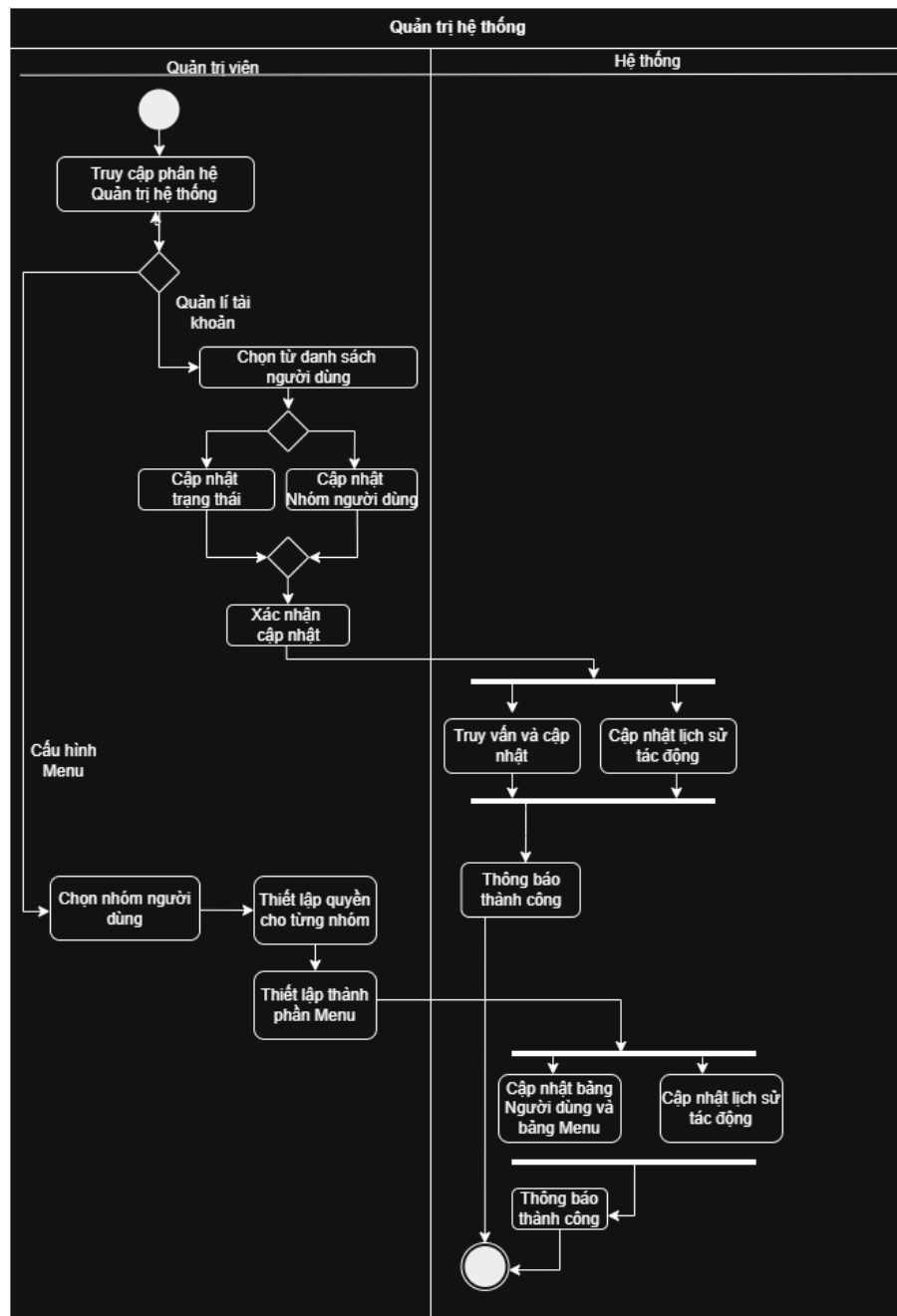


Hình 2.2.2: Sơ đồ hoạt động quy trình quản lý cơ sở dữ liệu của hệ thống

Luồng công việc bắt đầu khi nhân sự truy cập vào hệ thống quản lý để thực hiện các nghiệp vụ như thêm mới, điều chỉnh hoặc tra cứu thông tin giống vật nuôi. Để đảm bảo độ chính xác, mọi dữ liệu đầu vào đều phải trải qua bước kiểm duyệt tự động; nếu phát sinh lỗi, hệ thống sẽ điều hướng người dùng

thực hiện lại.

Đáng chú ý, nhằm đảm bảo tính toàn vẹn và khả năng kiểm soát dữ liệu, hệ thống vận hành theo cơ chế song song. Cụ thể, sau khi thông tin được xác nhận là hợp lệ, hệ thống sẽ cùng lúc lưu trữ vào cơ sở dữ liệu và cập nhật nhật ký hoạt động. Tương tự, chức năng tìm kiếm cũng tuân thủ quy trình kiểm tra từ khóa nghiêm ngặt trước khi hiển thị kết quả, giúp tối ưu hóa trải nghiệm người dùng và đảm bảo quy trình kết thúc trọn vẹn.



Hình 2.2.3: Sơ đồ hoạt động quy trình quản trị hệ thống

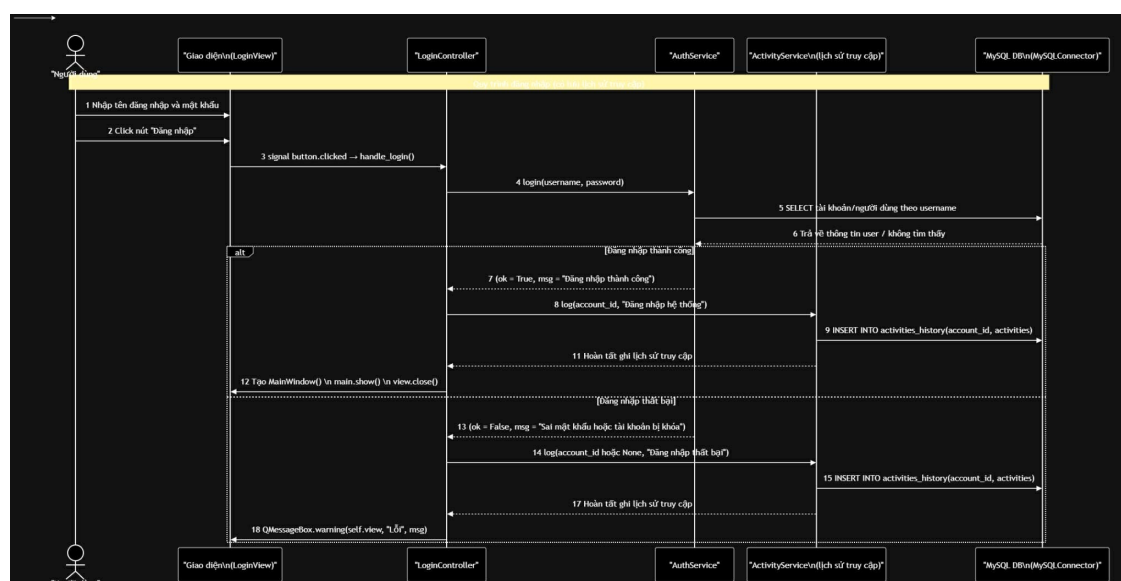
Quy trình bắt đầu khi Quản trị viên truy cập vào giao diện điều hành trung tâm để thực hiện công tác phân quyền và thiết lập hệ thống. Công việc được chia thành hai mảng nghiệp vụ trọng yếu: kiểm soát tài khoản người dùng (điều chỉnh trạng thái, nhóm quyền) và thiết lập hiển thị Menu theo chức năng.

Nhằm đảm bảo tính minh bạch và an toàn dữ liệu, hệ thống vận hành theo cơ chế đồng bộ hóa nghiêm ngặt. Cụ thể, mọi quyết định thay đổi cấu hình từ Quản trị viên sẽ được hệ thống xử lý qua hai kênh song song: vừa cập nhật ngay lập tức vào cơ sở dữ liệu hiện hành, vừa lưu trữ biên bản thao tác vào lịch sử hệ thống. Cơ chế này giúp đảm bảo khả năng truy vết đối với mọi thay đổi về quyền hạn trước khi thông báo cập nhật thành công.

2.3. Sequence Diagram

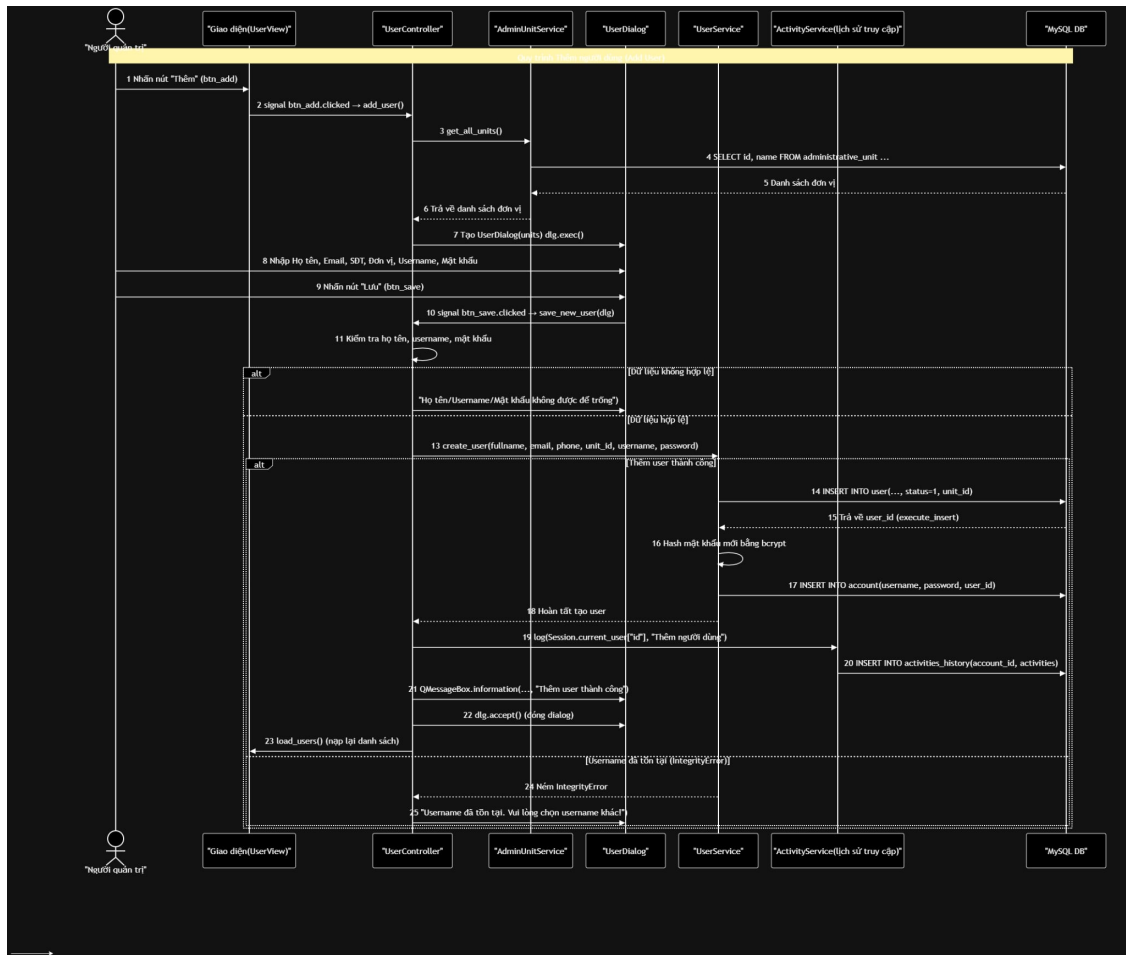
Bảng 2.3.1: Danh mục sơ đồ trường hợp sử dụng

STT	Tên sơ đồ	Tác nhân chính
1	Sơ đồ tuần tự quy trình truy cập hệ thống (Hình 2.3.1)	Người dùng, Quản trị viên, Người xem
2	Sơ đồ tuần tự quy trình đăng kí user mới (Hình 2.3.2)	Người dùng, Quản trị viên
3	Sơ đồ tuần tự quy trình quản lý dữ liệu (Hình 2.3.3)	Người dùng, Quản trị viên



Hình 2.3.1: Sơ đồ tuần tự quy trình truy cập hệ thống

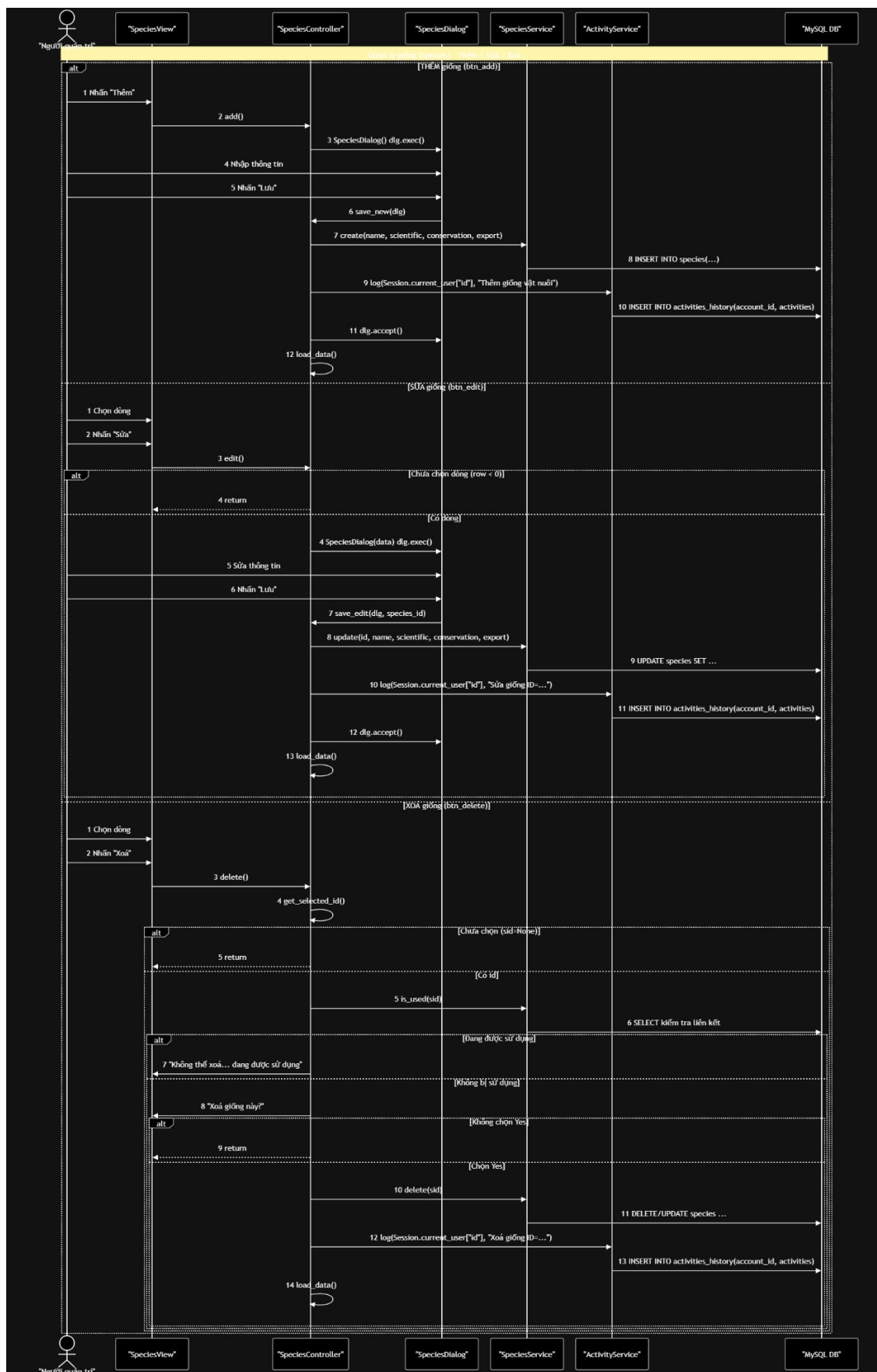
Sơ đồ tuần tự quy trình truy cập hệ thống đặc tả chi tiết trình tự tương tác giữa người dùng và các thành phần kiến trúc View, Controller, Service và Database trong tác vụ Đăng nhập. Tại lớp giao diện LoginView, sau khi nhận thông tin đầu vào, LoginController điều phối yêu cầu xác thực tới AuthService để thực hiện truy vấn đối soát dữ liệu trực tiếp với MySQL DB. Điểm nổi bật của quy trình là việc sử dụng khối điều khiển alt để phân nhánh kịch bản xử lý: trong trường hợp đăng nhập thành công, hệ thống xác nhận thông tin, kích hoạt ActivityService để lưu vết truy cập vào bảng activities-history, sau đó mới khởi tạo và điều hướng sang giao diện chính MainWindow. Ngược lại, nếu thất bại do sai thông tin hoặc tài khoản bị khóa, hệ thống vẫn thực hiện ghi nhật ký hoạt động thất bại trước khi hiển thị cảnh báo QMessageBox tới người dùng. Thiết kế này đảm bảo tính chặt chẽ trong quy trình xác thực, đồng thời duy trì khả năng giám sát an ninh thông qua việc ghi lại đầy đủ lịch sử hoạt động của cả hai trạng thái thành công và thất bại.



Hình 2.3.2: Sơ đồ tuần tự quy trình đăng kí user mới

Quy trình thêm người dùng mới bắt đầu khi người quản trị nhấn nút thêm trên giao diện chính. Hệ thống sẽ truy xuất danh sách đơn vị từ cơ sở dữ liệu để hiển thị trong hộp thoại nhập liệu. Sau khi người quản trị điền các thông tin cần thiết như họ tên, email, tài khoản và nhấn nút lưu, hệ thống sẽ tiến hành kiểm tra tính hợp lệ của dữ liệu.

Nếu thông tin hợp lệ, hệ thống thực hiện lưu người dùng mới vào cơ sở dữ liệu, mã hóa mật khẩu và ghi lại lịch sử hoạt động. Khi hoàn tất, một thông báo thành công sẽ hiển thị và danh sách người dùng trên giao diện được cập nhật lại. Trong trường hợp tên đăng nhập đã tồn tại, hệ thống sẽ thông báo lỗi để người quản trị điều chỉnh.



Hình 2.3.3: Sơ đồ tuần tự quy trình quản lý dữ liệu

Quy trình quản lý giống vật nuôi bao gồm ba chức năng chính là thêm, sửa và xóa. Khi người quản trị chọn thêm mới, hệ thống hiển thị hộp thoại nhập liệu, sau đó lưu thông tin vào cơ sở dữ liệu, ghi lại lịch sử hoạt động và làm mới danh sách hiển thị. Đối với chức năng sửa, hệ thống yêu cầu người dùng phải chọn một dòng dữ liệu trước, sau đó mới mở hộp thoại cập nhật và lưu thay đổi xuống hệ thống.

Riêng với chức năng xóa, quy trình thực hiện kiểm tra chặt chẽ hơn bằng việc xác minh xem giống vật nuôi đó có đang được sử dụng hay không. Nếu dữ liệu đang bị ràng buộc, hệ thống sẽ ngăn chặn thao tác xóa và thông báo cho người dùng. Ngược lại, nếu dữ liệu độc lập và người dùng xác nhận đồng ý xóa, hệ thống sẽ tiến hành loại bỏ bản ghi khỏi cơ sở dữ liệu và cập nhật lại giao diện.

2.4. ERD

2.4.1. Tổng quan về sơ đồ ERD

Mô hình ERD (Entity-Relationship Diagram) là một công cụ quan trọng trong lĩnh vực thiết kế cơ sở dữ liệu. Nó cung cấp một biểu diễn trực quan về các thực thể (entities), mối quan hệ (relationships) và các thuộc tính (attributes) trong hệ thống.

Các thành phần cơ bản của mô hình ERD

Thực thể (Entity):

- + Thực thể đại diện cho một đối tượng hay khái niệm trong thế giới thực, chẳng hạn như “Sinh viên”, “Khách hàng” hoặc “Sản phẩm”.
- + Ký hiệu: Hình chữ nhật.
- + Phân loại:

Thực thể mạnh (Strong Entity): Có khóa chính để phân biệt các bản ghi.

Thực thể yếu (Weak Entity): Phụ thuộc vào thực thể khác để tồn tại.

Thuộc tính (Attribute):

- + Thuộc tính biểu diễn thông tin hoặc đặc điểm của thực thể hoặc mối quan hệ.
- + Ký hiệu: Hình elip.
- + Phân loại:

Thuộc tính đơn giản (Simple Attribute): Chỉ chứa một giá trị.

Thuộc tính phức hợp (Composite Attribute): Có thể chia thành các thành phần nhỏ (ví dụ: “Họ” và “Tên”).

Thuộc tính đa trị (Multivalued Attribute): Chứa nhiều giá trị (địa chỉ email, số điện thoại).

Thuộc tính khóa (Key Attribute): Đóng vai trò là khóa chính.

Mối quan hệ (Relationship):

- + Mối quan hệ diễn tả mối liên hệ giữa các thực thể trong hệ thống.
- + Ký hiệu: Hình thoi.
- + Phân loại:

Mối quan hệ 1-1 (One-to-One): Mỗi thực thể A chỉ kết nối với một thực thể B.

Mối quan hệ 1-N (One-to-Many): Mỗi thực thể A có thể kết nối với nhiều thực thể B.

Mối quan hệ N-M (Many-to-Many): Nhiều thực thể A có thể kết nối với nhiều thực thể B.

Các khái niệm nâng cao

Cardinality (Số lượng):

- + Xác định số lượng thực thể tham gia trong mỗi quan hệ.
- + Ký hiệu: 1, N, hoặc M.

Participation (Sự tham gia):

- + Toàn phần (Total Participation): Một thực thể bắt buộc tham gia mỗi quan hệ.
- + Một phần (Partial Participation): Một thực thể có thể không tham gia mỗi quan hệ.

Chuyên biệt hóa và tổng quát hóa (Specialization & Generalization):

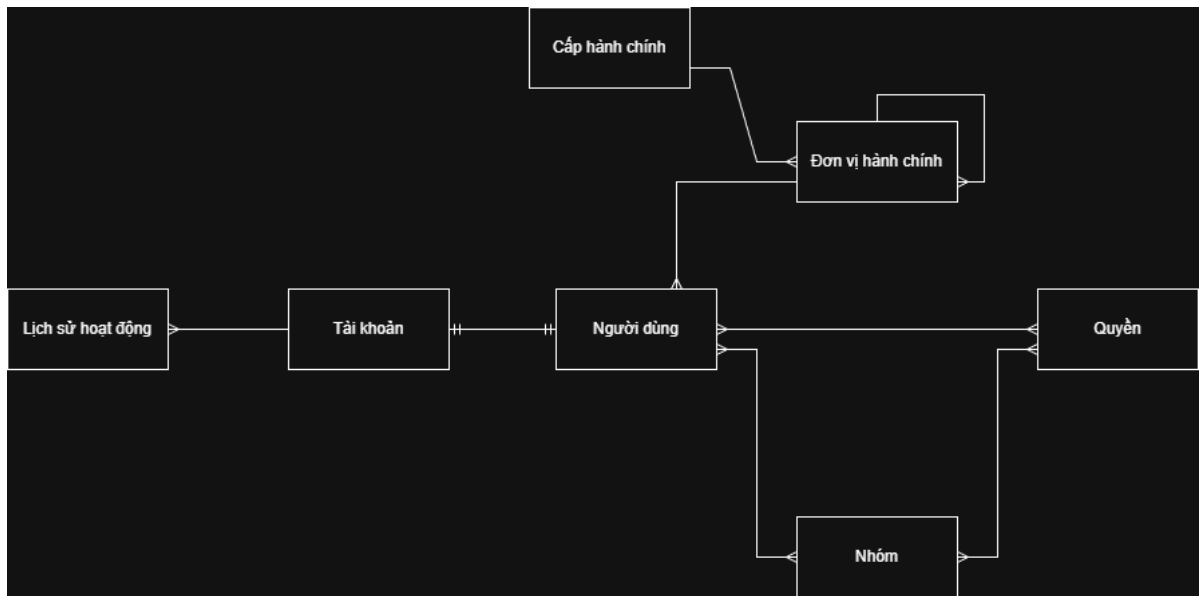
- + Tổng quát hóa: Gộp nhiều thực thể con lại thành thực thể cha.
- + Chuyên biệt hóa: Phân chia thực thể cha thành các thực thể con.

Thực thể yếu (Weak Entity) - Chi tiết:

- + Thực thể yếu không có khóa chính, phụ thuộc vào mỗi quan hệ định danh.
- + Ký hiệu: Được ký hiệu bằng hình thoi kép (cho mỗi quan hệ) hoặc hình chữ nhật kép (cho thực thể).
- + Thực thể yếu không có khóa chính, phụ thuộc vào mỗi quan hệ định danh.
- + Ký hiệu: Được ký hiệu bằng hình thoi kép...

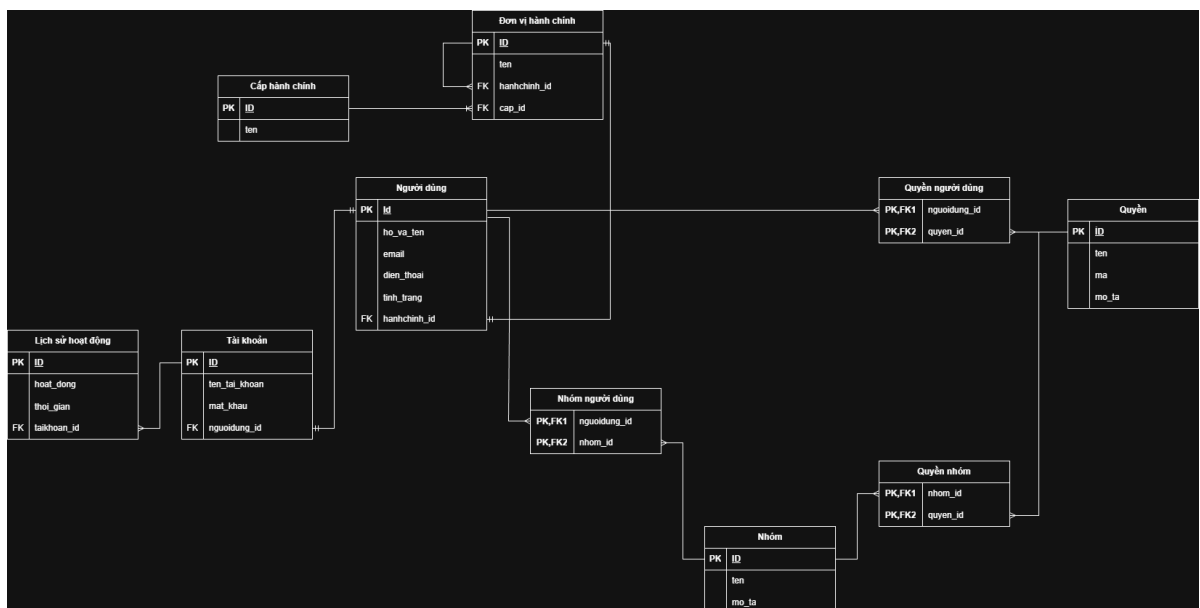
2.4.2. Các sơ đồ ERD

Sơ đồ thực thể ERD hệ thống



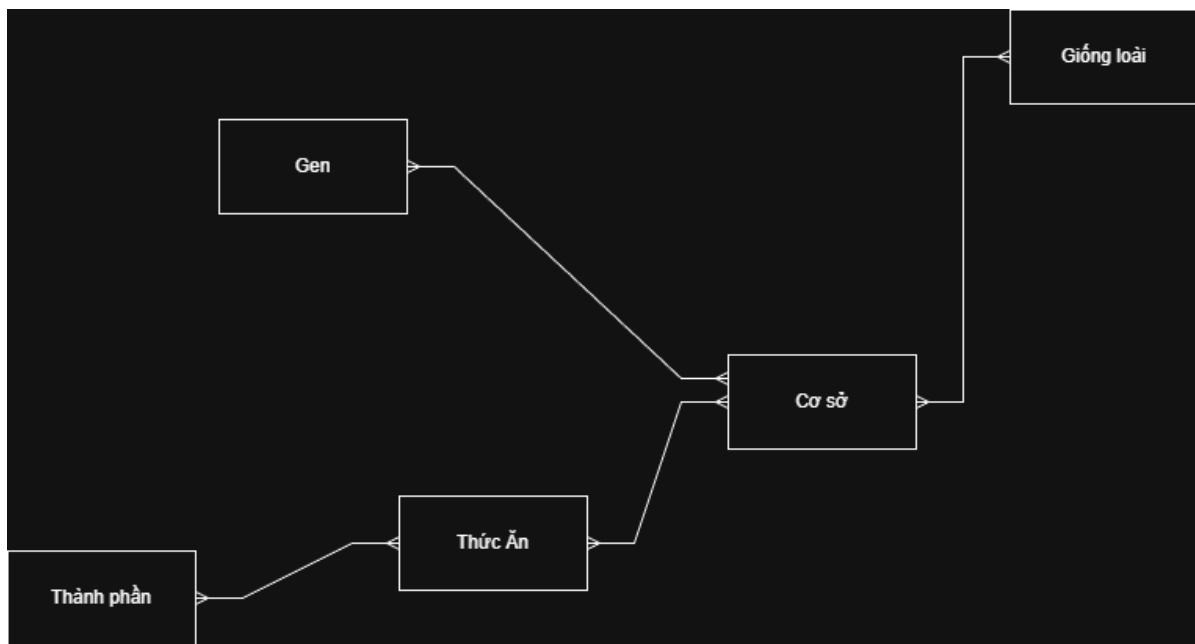
Hình 2.4.2.1: Sơ đồ thực thể ERD hệ thống

Sơ đồ vật lý thiết kế thực thể ERD hệ thống



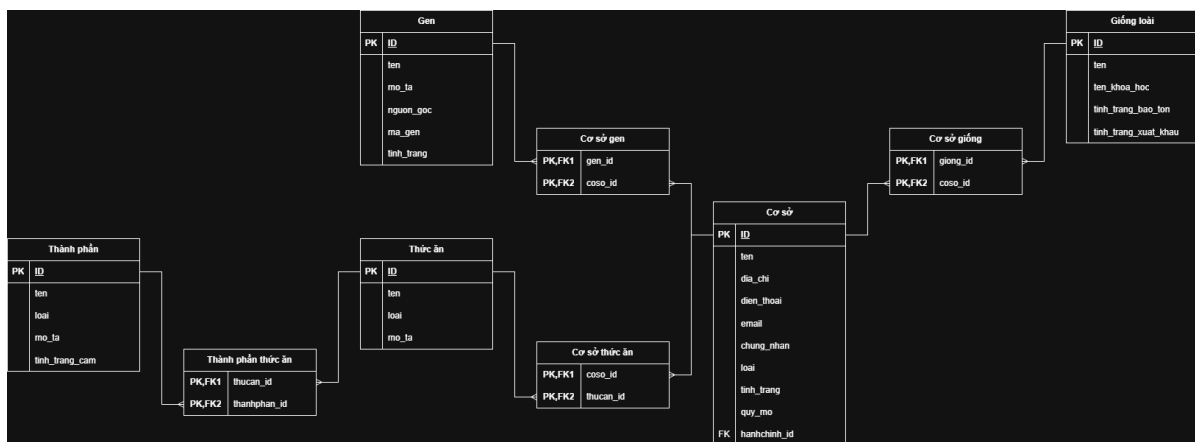
Hình 2.4.2.2: Sơ đồ vật lý thiết kế thực thể ERD hệ thống

Sơ đồ thực thể ERD giống và chăn nuôi



Hình 2.4.2.3: Sơ đồ thực thể ERD giống và chăn nuôi

Sơ đồ vật lý ERD giống và chăn nuôi



Hình 2.4.2.4: Sơ đồ vật lý ERD giống và chăn nuôi

2.5. Kết luận chung

Chương 2 đã hoàn thiện cơ sở kỹ thuật cho hệ thống Quản lý giống và chăn nuôi thông qua việc xác lập nền tảng phát triển dựa trên ngôn ngữ Python và thư viện giao diện PyQt6. Việc áp dụng kiến trúc MVC kết hợp với kỹ thuật ánh xạ đối tượng đã giúp chuẩn hóa quy trình xử lý, đảm bảo sự tách biệt giữa logic nghiệp vụ và giao diện hiển thị, đồng thời tăng cường an ninh hệ thống qua mô hình phân quyền. Bên cạnh các nguyên tắc thiết kế, hệ thống sơ đồ minh họa đã cụ thể hóa toàn diện quy trình vận hành: từ sơ đồ Use Case và ERD mô tả cấu trúc dữ liệu lưu trữ hồ sơ vật nuôi đến các sơ đồ Hoạt động và Tuần tự mô phỏng chi tiết luồng nghiệp vụ đặc thù như theo dõi vòng đời sinh trưởng hay quản lý tiêm chủng. Tổng thể các nội dung này đóng vai trò là bản vẽ kỹ thuật cốt lõi, tạo tiền đề vững chắc để triển khai mã nguồn và hoàn thiện các tính năng thực tế trong giai đoạn tiếp theo.

CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG

Chương này sẽ đi vào phân tích rõ thuộc tính của các bảng ở chương 2 và đi vào xây dựng cơ sở dữ liệu trên MySQL cũng như cấu trúc chương trình quản lý.

3.1. Từ điển dữ liệu

Bảng 3.1.1: Bảng hồ sơ

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng hồ sơ
Fullname	varchar	Nguyễn Minh Quân	Tên người dùng
Phone	varchar	0386138359	Số điện thoại của người dùng
Email	varchar	bitoi2005@gmail.com	Email của người dùng

Bảng 3.1.2: Bảng Đơn vị Hành chính

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng hồ sơ
Name	varchar	Sơn La	Tên đơn vị hành chính
AdministrativeID	Int	2	Khóa ngoại
LevelID	Int	3	Khóa ngoại

Bảng 3.1.3: Bảng Tài khoản

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng hồ sơ
Username	varchar	Admin	Tên tài khoản
Password	varchar	123456	Mật khẩu tài khoản
UserID	Int	1	Khóa ngoại

Bảng 3.1.4: Bảng Lịch sử hoạt động

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng hoạt động
Activities	Text	Đăng nhập	Các hoạt động
AccountID	Int	2	Khóa ngoại

Bảng 3.1.5: Bảng quyền

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng quyền
Name	varchar	Quản lý người dùng	Tên quyền
Code	varchar	USER MANGAGE	Mã phân quyền
Description	Text	CRUD Người dùng	Mô tả quyền

Bảng 3.1.6: Bảng Nhóm

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng hồ sơ
Name	varchar	ADMIN	Tên các nhóm
Description	Text	Quản trị hệ thống	Mô tả chức năng các nhóm

Bảng 3.1.7: Bảng Gen

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng Gen
Name	varchar	Gen giới tính đực	Tên Gen
Description	Text	Gen xác định giới	Mô tả Gen
Origin	varchar	Lợn	Nguồn gốc của Gen
GeneticCode	varchar	XY	Mã gen

Bảng 3.1.8: Bảng giống loài

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng loài
Name	varchar	Chó	Tên loài
Scientific Name	varchar	Canis lupus familiaris	Tên khoa học của loài
Conservation Status	varchar	Ưu tiên thấp	Tình trạng bảo tồn của loài
Export Restriction Status	varchar	Không	Tình trạng cấm xuất khẩu của loài

Bảng 3.1.9: Bảng cơ sở

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng cơ sở
Name	varchar	Hoàng Long	Tên cơ sở
Phone	varchar	0386138359	Số điện thoại của cơ sở
Email	varchar	hoanglong@gmail.com	Email của cơ sở
Adress	varchar	Số 2 Phố Hoàng Mai, Hà Nội	Địa chỉ của cơ sở
Certification	varchar	Đạt chuẩn 5 sao	Chứng nhận của cơ sở
Type	varchar	Quản lý giống chăn nuôi	Loại hình cơ sở
Status	TINYINT	1	Tình trạng hoạt động cơ sở
Scale	varchar	Trung bình	Quy mô cơ sở
UnitID	INT	1	Khóa ngoại

Bảng 3.1.10: Bảng thức ăn

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng thức ăn
Name	varchar	Cám chăn nuôi	Tên thức ăn
Type	varchar	Thức ăn gia súc	Loại thức ăn
Description	Text	Thức ăn dùng cho gia súc tại gia	Mô tả thức ăn

Bảng 3.1.11: Bảng thành phần

Tên thuộc tính	Kiểu dữ liệu	Giá trị mẫu	Mô tả
Id	int	1	Mã định danh cho từng thành phần
Name	varchar	Carbohydrate	Tên thành phần
Type	varchar	Nguyên liệu thức ăn gia súc	Loại thành phần
Description	TEXT	Ngô, cám, gạo	Mô tả thành phần
Banned Status	BOOLEAN	FALSE	Có bị cấm hay không

3.2. Triển khai cơ sở dữ liệu

Từ hệ thống dữ liệu đã xác định được, tiến hành xây dựng hệ thống cơ sở dữ liệu trên MySQL.

Tạo database

```
1 CREATE DATABASE livestock_app
2 CHARACTER SET utf8mb4
3 COLLATE utf8mb4_unicode_ci;
4
5 USE livestock_app;
6
7 CREATE TABLE administrative_level (
8     id INT AUTO_INCREMENT PRIMARY KEY,
9     name VARCHAR(100) NOT NULL
10 );
11
12 CREATE TABLE administrative_unit (
13     id INT AUTO_INCREMENT PRIMARY KEY,
14     name VARCHAR(255) NOT NULL,
15     administrative_id INT,
16     level_id INT NOT NULL,
17     FOREIGN KEY (administrative_id) REFERENCES administrative_unit(id),
18     FOREIGN KEY (level_id) REFERENCES administrative_level(id)
19 );
20
21 CREATE TABLE user (
22     id INT AUTO_INCREMENT PRIMARY KEY,
23     fullname VARCHAR(255),
24     email VARCHAR(255),
25     phone VARCHAR(20),
26     status TINYINT DEFAULT 1,
27     unit_id INT,
28     FOREIGN KEY (unit_id) REFERENCES administrative_unit(id)
29 );
30
31 CREATE TABLE account (
32     id INT AUTO_INCREMENT PRIMARY KEY,
33     username VARCHAR(100) UNIQUE NOT NULL,
34     password VARCHAR(255) NOT NULL,
35     user_id INT NOT NULL,
36     FOREIGN KEY (user_id) REFERENCES user(id)
```

```

37 );
38
39 CREATE TABLE activities_history (
40     id INT AUTO_INCREMENT PRIMARY KEY,
41     account_id INT NOT NULL,
42     activities TEXT,
43     time DATETIME DEFAULT CURRENT_TIMESTAMP,
44     FOREIGN KEY (account_id) REFERENCES account(id)
45 );
46
47 CREATE TABLE permissions (
48     id INT AUTO_INCREMENT PRIMARY KEY,
49     name VARCHAR(255),
50     code VARCHAR(100) UNIQUE NOT NULL,
51     description TEXT
52 );
53
54 CREATE TABLE `group` (
55     id INT AUTO_INCREMENT PRIMARY KEY,
56     name VARCHAR(255),
57     description TEXT
58 );
59
60 CREATE TABLE user_groups (
61     user_id INT,
62     group_id INT,
63     PRIMARY KEY (user_id, group_id),
64     FOREIGN KEY (user_id) REFERENCES user(id),
65     FOREIGN KEY (group_id) REFERENCES `group`(id)
66 );
67
68 CREATE TABLE user_permissions (
69     user_id INT,
70     permissions_id INT,
71     PRIMARY KEY (user_id, permissions_id),
72     FOREIGN KEY (user_id) REFERENCES user(id),
73     FOREIGN KEY (permissions_id) REFERENCES permissions(id)
74 );
75
76 CREATE TABLE group_permissions (
77     group_id INT,

```

```

78     permissions_id INT,
79     PRIMARY KEY (group_id, permissions_id),
80     FOREIGN KEY (group_id) REFERENCES `group`(id),
81     FOREIGN KEY (permissions_id) REFERENCES permissions(id)
82 );
83
84 CREATE TABLE gen (
85     id INT AUTO_INCREMENT PRIMARY KEY,
86     name VARCHAR(255),
87     description TEXT,
88     origin VARCHAR(255),
89     genetic_code VARCHAR(100),
90     status TINYINT
91 );
92
93 CREATE TABLE species (
94     id INT AUTO_INCREMENT PRIMARY KEY,
95     name VARCHAR(255),
96     scientific_name VARCHAR(255),
97     conservation_status VARCHAR(255),
98     export_restriction_status VARCHAR(255)
99 );
100
101 CREATE TABLE facility (
102     id INT AUTO_INCREMENT PRIMARY KEY,
103     name VARCHAR(255),
104     address VARCHAR(255),
105     phone VARCHAR(20),
106     email VARCHAR(255),
107     certification VARCHAR(255),
108     type VARCHAR(255),
109     status TINYINT,
110     scale VARCHAR(100),
111     unit_id INT,
112     FOREIGN KEY (unit_id) REFERENCES administrative_unit(id)
113 );
114
115 CREATE TABLE facility_gen (
116     gen_id INT,
117     facility_id INT,
118     PRIMARY KEY (gen_id, facility_id),

```



```

119     FOREIGN KEY (gen_id) REFERENCES gen(id),
120     FOREIGN KEY (facility_id) REFERENCES facility(id)
121 );
122
123 CREATE TABLE facility_species (
124     species_id INT,
125     facility_id INT,
126     PRIMARY KEY (species_id, facility_id),
127     FOREIGN KEY (species_id) REFERENCES species(id),
128     FOREIGN KEY (facility_id) REFERENCES facility(id)
129 );
130
131 CREATE TABLE food (
132     id INT AUTO_INCREMENT PRIMARY KEY,
133     name VARCHAR(255),
134     type VARCHAR(255),
135     description TEXT
136 );
137
138 CREATE TABLE substances (
139     id INT AUTO_INCREMENT PRIMARY KEY,
140     name VARCHAR(255),
141     type VARCHAR(255),
142     description TEXT,
143     banned BOOLEAN DEFAULT FALSE
144 );
145
146 CREATE TABLE food_substance (
147     food_id INT,
148     substances_id INT,
149     PRIMARY KEY (food_id, substances_id),
150     FOREIGN KEY (food_id) REFERENCES food(id),
151     FOREIGN KEY (substances_id) REFERENCES substances(id)
152 );
153
154 CREATE TABLE facility_food (
155     facility_id INT,
156     food_id INT,
157     PRIMARY KEY (facility_id, food_id),
158     FOREIGN KEY (facility_id) REFERENCES facility(id),
159     FOREIGN KEY (food_id) REFERENCES food(id)

```

```

160 );
161
162 ALTER TABLE `group`
163 ADD COLUMN is_active TINYINT(1) DEFAULT 1;
164
165 ALTER TABLE permissions
166 ADD COLUMN is_active TINYINT(1) DEFAULT 1;
167
168 CREATE TABLE instructions (
169     id INT AUTO_INCREMENT PRIMARY KEY,
170     title VARCHAR(255),
171     content TEXT
172 );

```

3.3. Triển khai kiến trúc hệ thống

3.3.1. Xây dựng lớp truy cập dữ liệu

Để đảm bảo việc kết nối cơ sở dữ liệu được thống nhất và tối ưu, cần xây dựng một lớp chung MySQLConnector - chịu trách nhiệm cho việc mở kết nối, thực thi câu lệnh SQL và đóng kết nối an toàn. Tất cả Services trong hệ thống sẽ gọi đến lớp này.

Kết nối cơ sở dữ liệu

```

1  import mysql.connector
2  from mysql.connector import Error
3
4  class MySQLConnector:
5      def __init__(self):
6          self.config = {
7              "host": "localhost",
8              "user": "root",
9              "password": "",
10             "database": "livestock_app"
11         }
12
13     def connect(self):
14         return mysql.connector.connect(**self.config)
15
16     def fetch_one(self, query, params=None):
17         conn = self.connect()

```

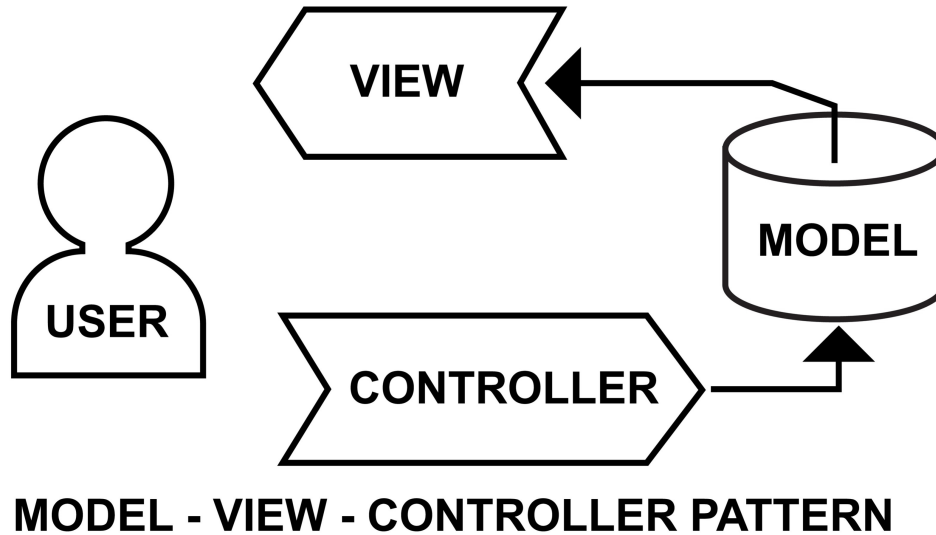
```

18     cursor = conn.cursor(dictionary=True)
19     cursor.execute(query, params or ())
20     result = cursor.fetchone()
21     cursor.close()
22     conn.close()
23     return result
24
25 def fetch_all(self, query, params=None):
26     conn = self.connect()
27     cursor = conn.cursor(dictionary=True)
28     cursor.execute(query, params or ())
29     result = cursor.fetchall()
30     cursor.close()
31     conn.close()
32     return result
33
34 def execute(self, query, params=None):
35     conn = self.connect()
36     cursor = conn.cursor()
37     cursor.execute(query, params or ())
38     conn.commit()
39     cursor.close()
40     conn.close()
41
42 def execute_insert(self, query, params=None):
43     conn = self.connect()
44     cursor = conn.cursor()
45     cursor.execute(query, params or ())
46     conn.commit()
47     last_id = cursor.lastrowid
48     cursor.close()
49     conn.close()
50     return last_id

```

3.3.2. Kiến trúc chương trình theo mô hình MVC

thiết kế và xây dựng dựa trên mô hình kiến trúc MVC (Model - View - Controller). Việc áp dụng mô hình này giúp tách biệt rõ ràng giữa giao diện người dùng, logic điều khiển và xử lý dữ liệu, giúp mã nguồn dễ dàng bảo trì và mở rộng.



Hình 3.3.2.1: Sơ đồ MODEL - VIEW - CONTROLLER

Cụ thể, các thành phần trong chương trình được phân chia như sau:

Tầng Model (Mô hình dữ liệu và nghiệp vụ): Các file trong thư mục services/ và database/.

Đây là "trái tim" của hệ thống. Tầng này chịu trách nhiệm quản lý dữ liệu, thực thi các quy tắc nghiệp vụ (như tính toán lượng thức ăn, kiểm tra tồn kho) và giao tiếp trực tiếp với cơ sở dữ liệu MySQL.

System history services

```
1 from database.mysql_connector import MySQLConnector
2
3 class SystemHistoryService:
4     def __init__(self):
5         self.db = MySQLConnector()
6
7     def get_all(self):
8         return self.db.fetch_all("""
9             SELECT ah.id,
10                a.username,
11                ah.activities,
12                ah.time,
13                a.user_id
```

```

14         FROM activities_history ah
15         JOIN account a ON ah.account_id = a.id
16         ORDER BY ah.time DESC
17     """)
18
19 def search(self, keyword):
20     kw = f"%{keyword}%"
21     return self.db.fetch_all("""
22         SELECT ah.id,
23                a.username,
24                ah.activities,
25                ah.time,
26                a.user_id
27         FROM activities_history ah
28         JOIN account a ON ah.account_id = a.id
29         WHERE a.username LIKE %s
30                OR ah.activities LIKE %s
31         ORDER BY ah.time DESC
32     """, (kw, kw))

```

Tầng View (Giao diện người dùng): Các file trong thư mục views/ (sử dụng thư viện PyQt6).

Là tầng hiển thị, cung cấp các biểu mẫu (Form) để người dùng nhập liệu (như thông tin vật nuôi, phiếu nhập/xuất) và xem các báo cáo thống kê. View hoàn toàn không chứa logic xử lý dữ liệu.

species view

```
1 from PyQt6.QtWidgets import (
2     QWidget, QVBoxLayout, QHBoxLayout,
3     QPushButton, QTableWidgetItem,
4     QLineEdit
5 )
6
7 class GenView(QWidget):
8     def __init__(self):
9         super().__init__()
10        self.setWindowTitle("Quan ly nguồn gen vật nuôi")
11        self.resize(900, 500)
12
13        # ===== Table =====
14        self.table = QTableWidgetItem(0, 6)
15        self.table.setHorizontalHeaderLabels([
16            "ID", "Ten Gen", "Mo Ta",
17            "Nguon goc", "Ma Gen", "Trang thai"
18        ])
19
20        self.txt_search = QLineEdit()
21        self.txt_search.setPlaceholderText("Nhap ten Gen")
22
23        self.btn_search = QPushButton("Tim kiem")
24        self.btn_refresh = QPushButton("Lam moi")
25
26        self.btn_add = QPushButton("Them")
27        self.btn_edit = QPushButton("Sua")
28        self.btn_delete = QPushButton("Xoa")
29
30        topbar_layout = QHBoxLayout()
31        topbar_layout.addWidget(self.txt_search)
32        topbar_layout.addWidget(self.btn_search)
33        topbar_layout.addWidget(self.btn_refresh)
34        topbar_layout.addStretch()
35        topbar_layout.addWidget(self.btn_add)
36        topbar_layout.addWidget(self.btn_edit)
```

```

37         topbar_layout.addWidget(self.btn_delete)
38
39         # ===== Main layout =====
40         layout = QVBoxLayout()
41         layout.addLayout(topbar_layout)
42         layout.addWidget(self.table)
43
44         self.setLayout(layout)

```

Thiết kế UI chính của hệ thống

main window

```

1  from PyQt6.QtWidgets import (
2      QMainWindow, QMessageBox, QWidget,
3      QHBoxLayout, QTreeWidget, QTreeWidgetItem,
4      QStackedWidget, QLabel
5  )
6  from PyQt6.QtGui import QIcon
7  from PyQt6.QtCore import Qt
8
9  from auth.permission_checker import has_permission
10 from auth.session import Session
11
12 from controllers.admin_unit_controller import AdminUnitController
13 from controllers.user_controller import UserController
14 from controllers.change_password_controller import
15     ChangePasswordController
16 from controllers.group_controller import GroupController
17 from controllers.system_history_controller import
18     SystemHistoryController
19 from controllers.species_controller import SpeciesController
20 from controllers.gen_controller import GenController
21 from controllers.food_controller import FoodController
22 from controllers.substance_controller import SubstanceController
23 from controllers.permission_controller import PermissionController
24 from controllers.account_controller import AccountController
25 from controllers.manual_controller import ManualController
26 from controllers.report_controller import ReportController
27 from controllers.breeding_facility_controller import

```

```

    BreedingFacilityController
26 from controllers.breeding_material_facility_controller import
    BreedingMaterialFacilityController
27 from controllers.male_breeding_owner_controller import
    MaleBreedingOwnerController
28 from controllers.trading_facility_controller import
    TradingFacilityController
29 from controllers.testing_facility_controller import
    TestingFacilityController
30 from controllers.gen_collection_facility_controller import
    GenCollectionFacilityController
31 from controllers.gen_conservation_facility_controller import
    GenConservationFacilityController
32 from controllers.gen_development_facility_controller import
    GenDevelopmentFacilityController
33 from controllers.feed_facility_controller import FeedFacilityController
34 from controllers.feed_trading_facility_controller import
    FeedTradingFacilityController
35 from controllers.feed_testing_facility_controller import
    FeedTestingFacilityController
36 import os
37
38 class MainWindow(QMainWindow):
39     def __init__(self):
40         super().__init__()
41         self.setWindowTitle("He thong quan ly thuc an chan nuoi")
42         self.resize(1300, 750)
43
44         self.controllers = {} # cache controller
45         self._init_ui()
46
47         # ===== LOAD QSS STYLE =====
48         try:
49             base_dir = os.path.dirname(os.path.abspath(__file__))
50             qss_path = os.path.join(base_dir, "..", "fonts",
51                                     "style.qss")
52             qss_path = os.path.abspath(qss_path)
53
54             with open(qss_path, "r", encoding="utf-8") as f:
55                 self.setStyleSheet(f.read())
56         except FileNotFoundError:

```



```

56         print("Khong tim thay file style.qss, su dung style mac
           dinh")
57
58     # ===== UI =====
59     def _init_ui(self):
60         central = QWidget()
61         layout = QHBoxLayout(central)
62         layout.setContentsMargins(0, 0, 0, 0)
63
64         # ===== SIDEBAR =====
65         self.sidebar = QTreeWidget()
66         self.sidebar.setHeaderHidden(True)
67         self.sidebar.setMaximumWidth(300)
68         self.sidebar.itemClicked.connect(self._handle_sidebar_click)
69
70         self._build_sidebar()
71
72         # ===== STACKED WIDGET =====
73         self.stack = QStackedWidget()
74
75         # Trang m c .inh
76         self.home = QLabel("Chao mung ban den voi he thong quan ly")
77         self.home.setAlignment(Qt.AlignmentFlag.AlignCenter)
78         self.stack.addWidget(self.home)
79
80         layout.addWidget(self.sidebar)
81         layout.addWidget(self.stack)
82         self.setCentralWidget(central)
83
84     # ===== SIDEBAR =====
85     def _build_sidebar(self):
86         self.sidebar.clear()
87
88         system = QTreeWidgetItem(["He thong"])
89         system.setIcon(0, QIcon.fromTheme("preferences-system"))
90
91         if Session.current_user:
92             self._add_item(system, "Thong tin tai khoan",
93                             "user-identity")
94
95         if "MANUAL_VIEW" in Session.permissions:

```

```

95         self._add_item(system, "Huong dan su dung", "help-browser")
96
97     if has_permission("USER_MANAGE"):
98         self._add_item(system, "Quan ly nguoi dung",
99             "user-group-new")
100
101     if has_permission("GROUP_MANAGE"):
102         self._add_item(system, "Quan ly nhom", "system-users")
103
104     if has_permission("PERMISSION_MANAGE"):
105         self._add_item(system, "Quan ly phan quyen",
106             "dialog-password")
107
108     if has_permission("ADMIN_UNIT_MANAGE"):
109         self._add_item(system, "Don vi hanh chinh", "folder")
110
111     if has_permission("HISTORY_VIEW"):
112         self._add_item(system, "Lich su he thong", "view-history")
113
114     self._add_item(system, "Doi mat khau", "document-edit")
115     self._add_item(system, "Dang xuat", "system-log-out")
116
117     if has_permission("FACILITY_MANAGE"):
118         facility = QTreeWidgetItem(["Quan ly co so"])
119         facility.setIcon(0, QIcon.fromTheme("office-building"))
120
121         self._add_item(facility, "Co so san xuat con giong",
122             "list-add")
123         self._add_item(facility, "Co so tinh phoi, ap trung",
124             "network-server")
125         self._add_item(facility, "Co so huu duc giong",
126             "user-group")
127         self._add_item(facility, "Co so mua ban giong", "folder")
128         self._add_item(facility, "Co so khao nghiem giong", "lab")
129         self._add_item(facility, "Danh muc giong dac biet",
130             "special")
131
132     if has_permission("GEN_MANAGE"):
133         gen = QTreeWidgetItem(["Nguon gen giong vat nuoi"])
134         gen.setIcon(0, QIcon.fromTheme("gen-facility"))

```

```

130         self._add_item(gen, "Quan ly nguon gen", "gen-manage")
131         self._add_item(gen, "Co so thu thap nguon gen",
            "collection")
132         self._add_item(gen, "Co so bao ton nguon gen",
            "conservation")
133         self._add_item(gen, "Co so phat trien nguon gen",
            "development")
134
135         food = QTreeWidgetItem(["Thuc an chan nuoi"])
136         food.setIcon(0, QIcon.fromTheme("applications-utilities"))
137
138         if has_permission("FOOD_MANAGE"):
139             self._add_item(food, "Thuc an", "package-x-generic")
140             self._add_item(food, "Co so san xuat thuc an chan nuoi",
                "production-feed")
141             self._add_item(food, "Co so mua ban thuc an chan nuoi",
                "trading-feed")
142             self._add_item(food, "Co so khao nghiem thuc an chan nuoi",
                "testing-feed")
143
144         if has_permission("SUBSTANCE_MANAGE"):
145             self._add_item(food, "Danh muc nguyen lieu \& chat cam",
                "dialog-warning")
146
147         report = QTreeWidgetItem(["Bao cao"])
148         report.setIcon(0, QIcon.fromTheme("document-preview"))
149         if has_permission("REPORT_VIEW"):
150             self._add_item(report, "Bao cao he thong",
                "document-preview")
151
152         self.sidebar.addTopLevelItems([system, facility, gen, food,
            report])
153         self.sidebar.expandAll()
154
155     def _add_item(self, parent, text, icon_name):
156         item = QTreeWidgetItem(parent, [text])
157         item.setIcon(0, QIcon.fromTheme(icon_name))
158         return item
159
160     def _handle_sidebar_click(self, item, column):
161         text = item.text(0)

```

```

162
163     if text == "Dang xuat":
164         self.logout()
165         return
166
167     if text == "Doi mat khau":
168         self.change_password()
169         return
170
171     mapping = {
172         "Quan ly nguoi dung": UserController,
173         "Quan ly nhom": GroupController,
174         "Quan ly phan quyen": PermissionController,
175         "Don vi hanh chinh": AdminUnitController,
176         "Lich su he thong": SystemHistoryController,
177         "Quan ly giong": SpeciesController,
178
179         "Quan ly co so": FacilityController,
180         "Co so san xuat con giong": BreedingFacilityController,
181         "Co so tinh, phoi, ap trung":
182             BreedingMaterialFacilityController,
183         "Co so so huu duc giong": MaleBreedingOwnerController,
184         "Co so mua ban giong": TradingFacilityController,
185         "Co so khao nghiem giong": TestingFacilityController,
186         "Danh muc giong dac biet": SpeciesController,
187
188         "Quan ly nguon gen": GenController,
189         "Co so thu thap nguon gen": GenCollectionFacilityController,
190         "Co so bao ton nguon gen":
191             GenConservationFacilityController,
192         "Co so phat trien nguon gen":
193             GenDevelopmentFacilityController,
194         "Thuc an": FoodController,
195         "Co so san xuat thuc an chan nuoi": FeedFacilityController,
196         "Co so mua ban thuc an chan nuoi":
197             FeedTradingFacilityController,
198         "Co so khao nghiem thuc an chan nuoi":
199             FeedTestingFacilityController,
200         "Danh muc nguyen lieu \& chat cam": SubstanceController,
201         "Thong tin tai khoan": AccountController,
202         "Huong dan su dung": ManualController,

```

```

198         "Bao cao he thong": ReportController
199     }
200
201     if text not in mapping:
202         return
203
204     Neu da mo roi -> chuyen trang
205     if text in self.controllers:
206         self.stack.setCurrentWidget(self.controllers[text].view)
207         return
208
209     # Chua mo -> tao controller + add vao stack
210     controller = mapping[text]()
211     self.controllers[text] = controller
212     self.stack.addWidget(controller.view)
213     self.stack.setCurrentWidget(controller.view)
214
215     # ===== ACTIONS =====
216     def change_password(self):
217         dlg = ChangePasswordController()
218         dlg.view.exec()
219
220     def logout(self):
221         reply = QMessageBox.question(
222             self,
223             "Dang xuat",
224             "Ban co chac chan muon dang xuat khong?",
225             QMessageBox.StandardButton.Yes |
226                 QMessageBox.StandardButton.No
227         )
228
229         if reply == QMessageBox.StandardButton.Yes:
230
231             Session.current_user = None
232             Session.permissions = set()
233             QMessageBox.information(self, "Logout", "Da dang xuat")
234             self.close()

```

Tầng Controller (Bộ điều khiển): Các file trong thư mục controllers/

Đóng vai trò cầu nối trung gian. Controller lắng nghe các sự kiện từ View (người dùng bấm nút Thêm, Sửa, Xóa), thực hiện kiểm tra tính hợp lệ của dữ liệu, sau đó gọi xuống Model để xử lý và cập nhật lại kết quả hiển thị trên View.

login controller

```
1 from auth.auth_service import AuthService
2 from views.login_view import LoginView
3 from views.main_window import MainWindow
4 from PyQt6.QtWidgets import QMessageBox
5 from controllers.forgot_password_controller import
   ForgotPasswordController
6
7 class LoginController:
8     def __init__(self):
9         self.view = LoginView()
10        self.auth = AuthService()
11        self.view.button.clicked.connect(self.handle_login)
12        self.view.btn_forgot.clicked.connect(self.open_forgot_password)
13
14
15    def handle_login(self):
16        ok, msg = self.auth.login(
17            self.view.username.text(),
18            self.view.password.text()
19        )
20        if ok:
21            self.main = MainWindow()
22            self.main.show()
23            self.view.close()
24        else:
25            QMessageBox.warning(self.view, "Loi", msg)
26
27    def open_forgot_password(self):
28        self.forgot = ForgotPasswordController()
29        self.forgot.show()
```

3.3.3: Một số thiết kế còn lại

Hiện hộp dialog popup

feed facility dialog

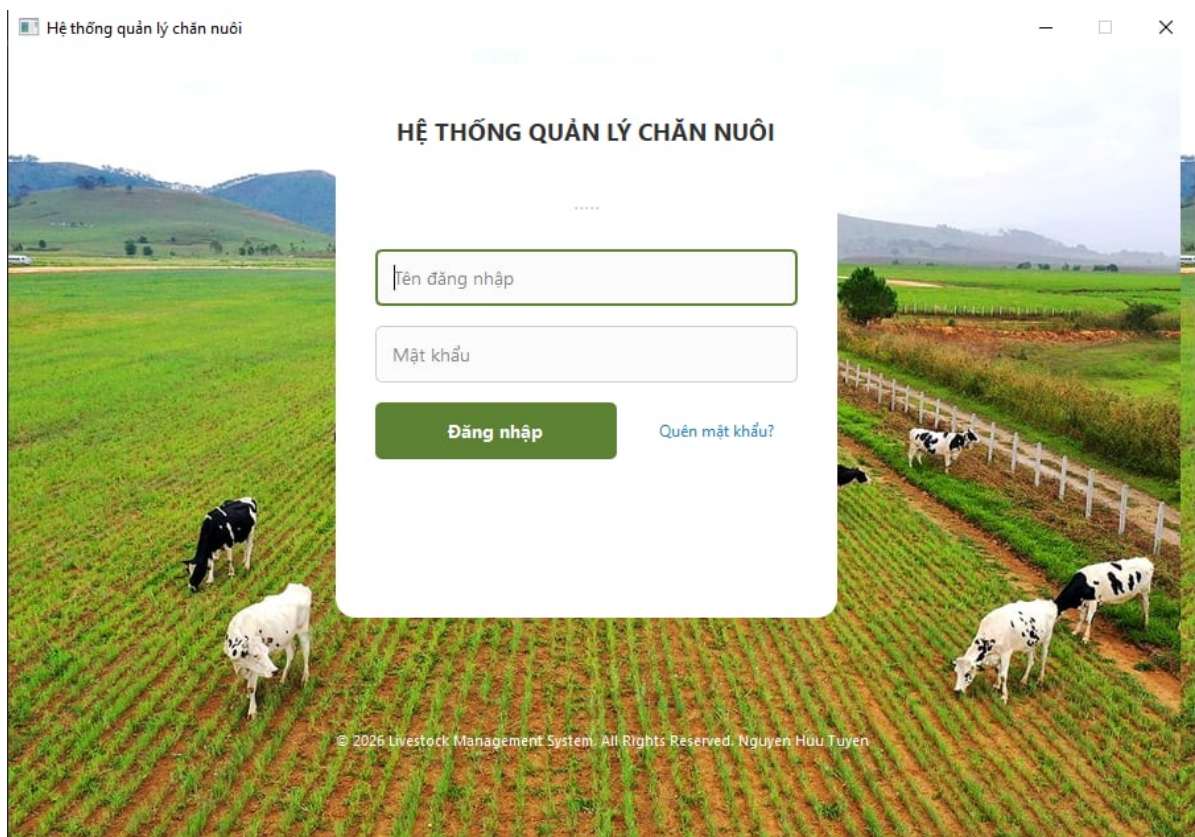
```
1 from dialogs.breeding_facility_dialog import BreedingFacilityDialog
2
3 class FeedFacilityDialog(BreedingFacilityDialog):
4     def __init__(self, data=None):
5         super().__init__(data)
6         self.setWindowTitle("Thông tin cơ sở thực an chăn nuôi")
```

Main chạy chương trình

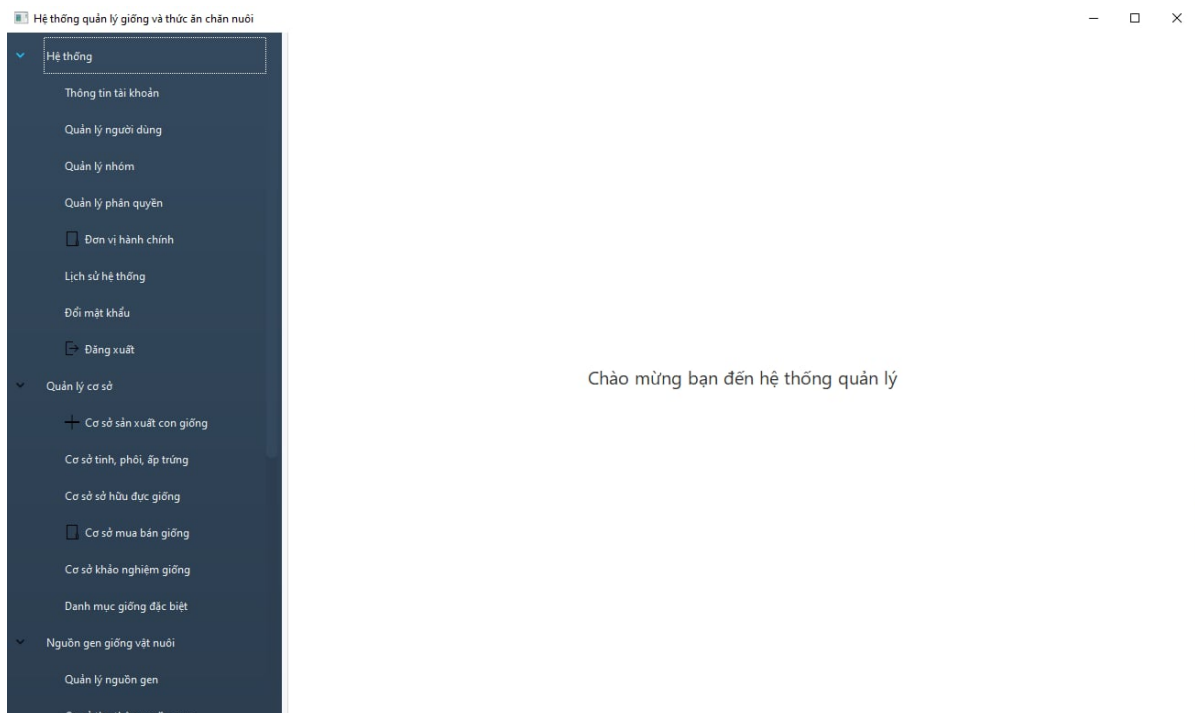
main.py

```
1 import sys
2 from PyQt6.QtWidgets import QApplication
3 from controllers.login_controller import LoginController
4
5 if __name__ == '__main__':
6     app = QApplication(sys.argv)
7     login = LoginController()
8     login.view.show()
9     sys.exit(app.exec())
```

3.4. Giao diện hệ thống



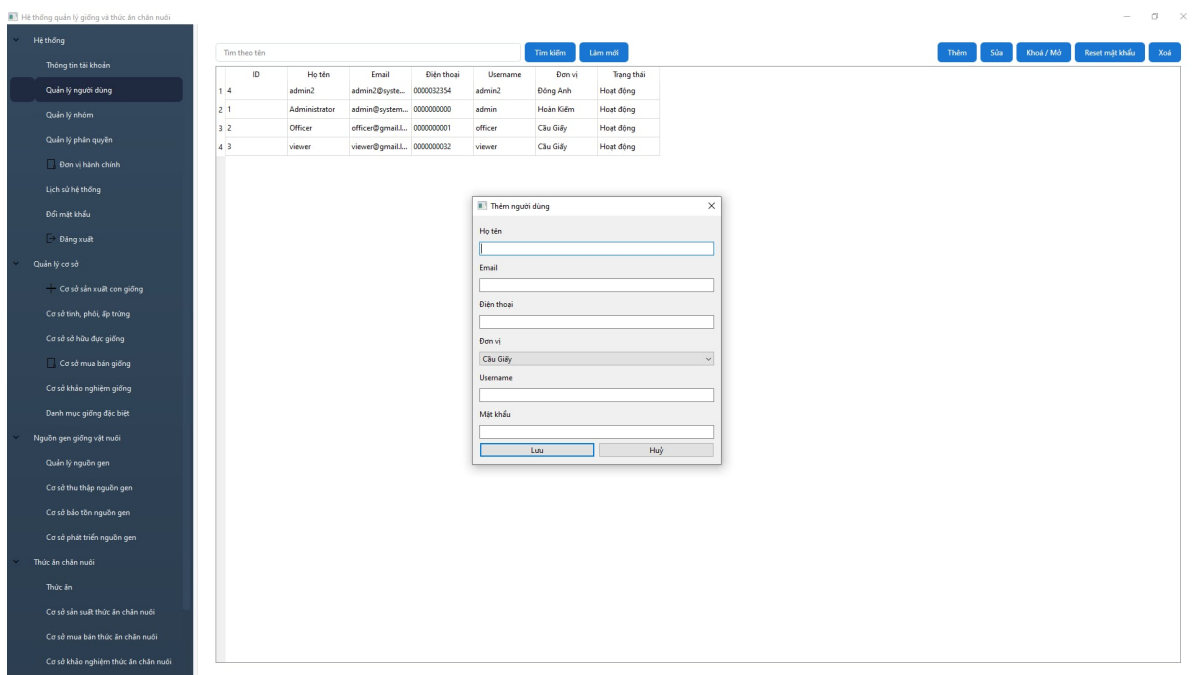
Hình 3.4.1: Giao diện đăng nhập hệ thống



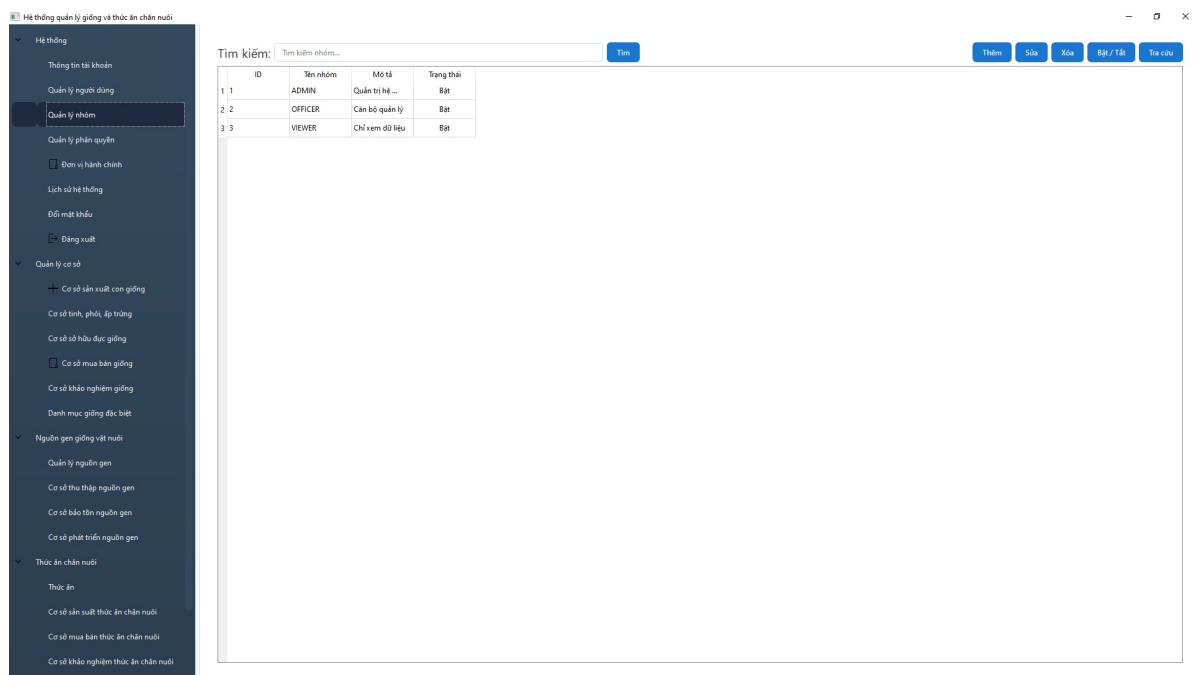
Hình 3.4.2: Giao diện chương trình



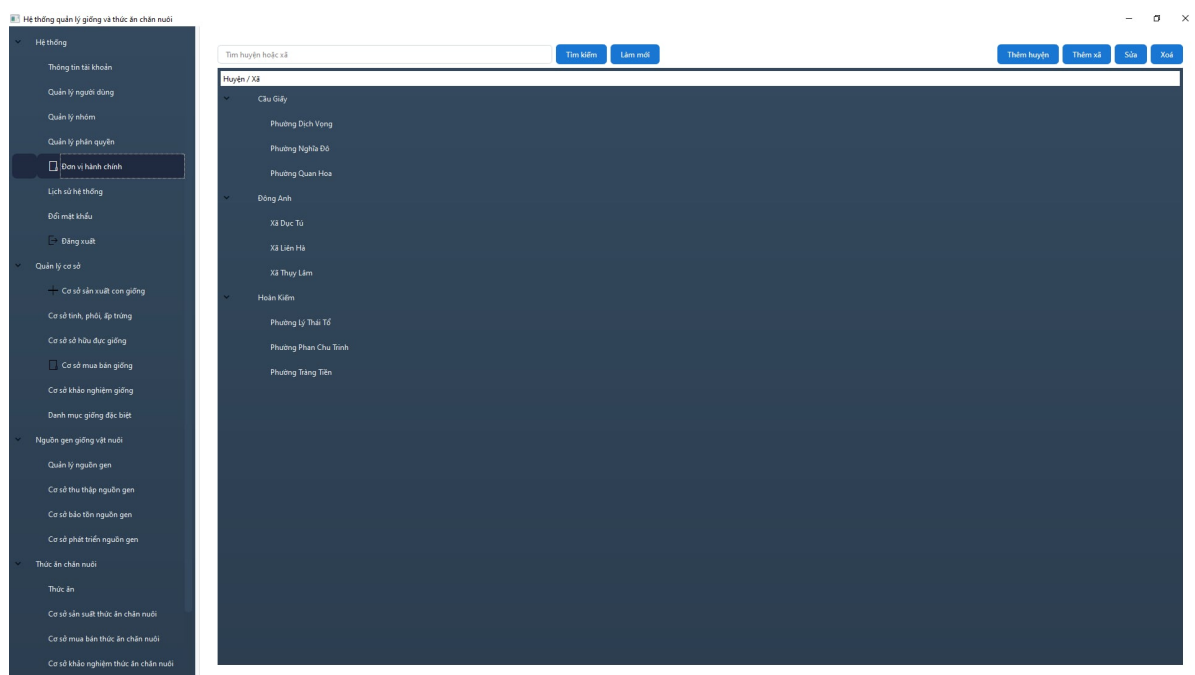
Hình 3.4.3: Thông tin tài khoản



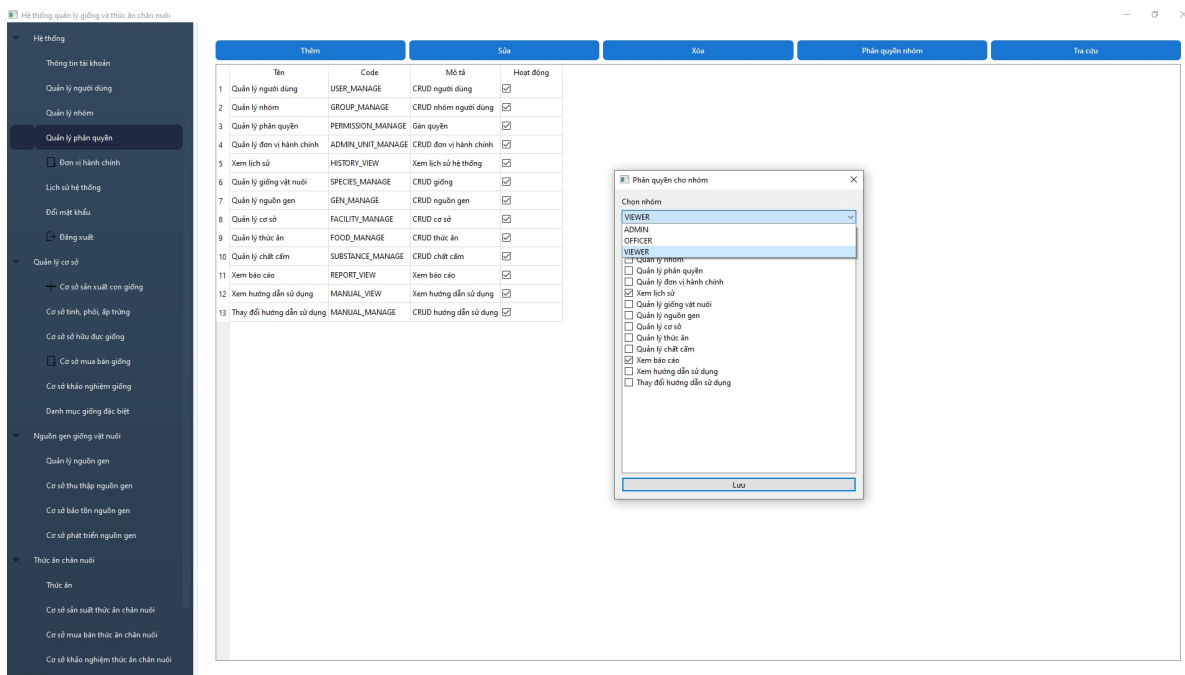
Hình 3.4.4: Quản lý tài khoản người dùng



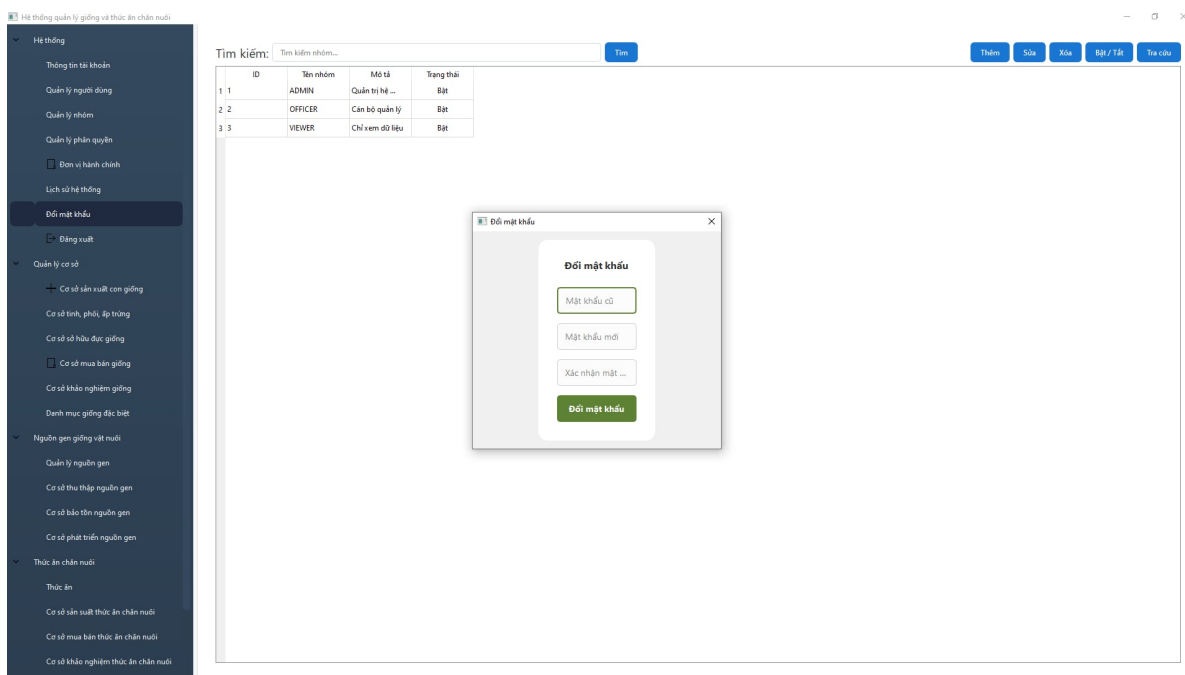
Hình 3.4.5: Quản lý nhóm người dùng



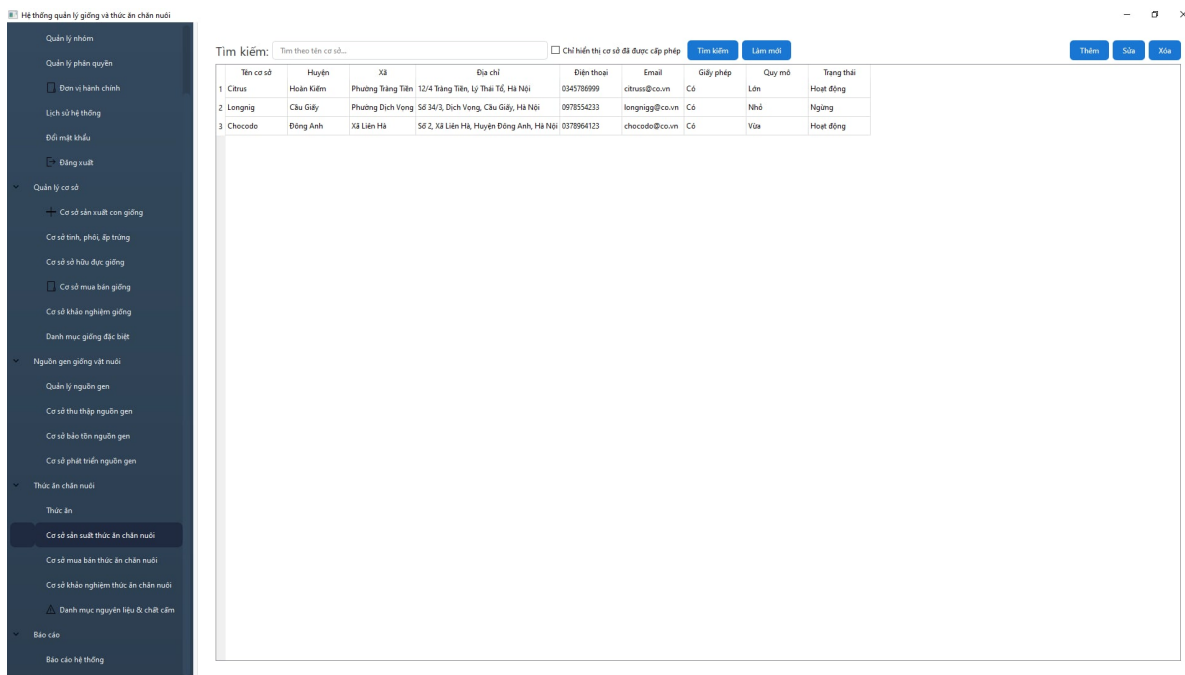
Hình 3.4.6: Quản lý CSDL đơn vị hành chính



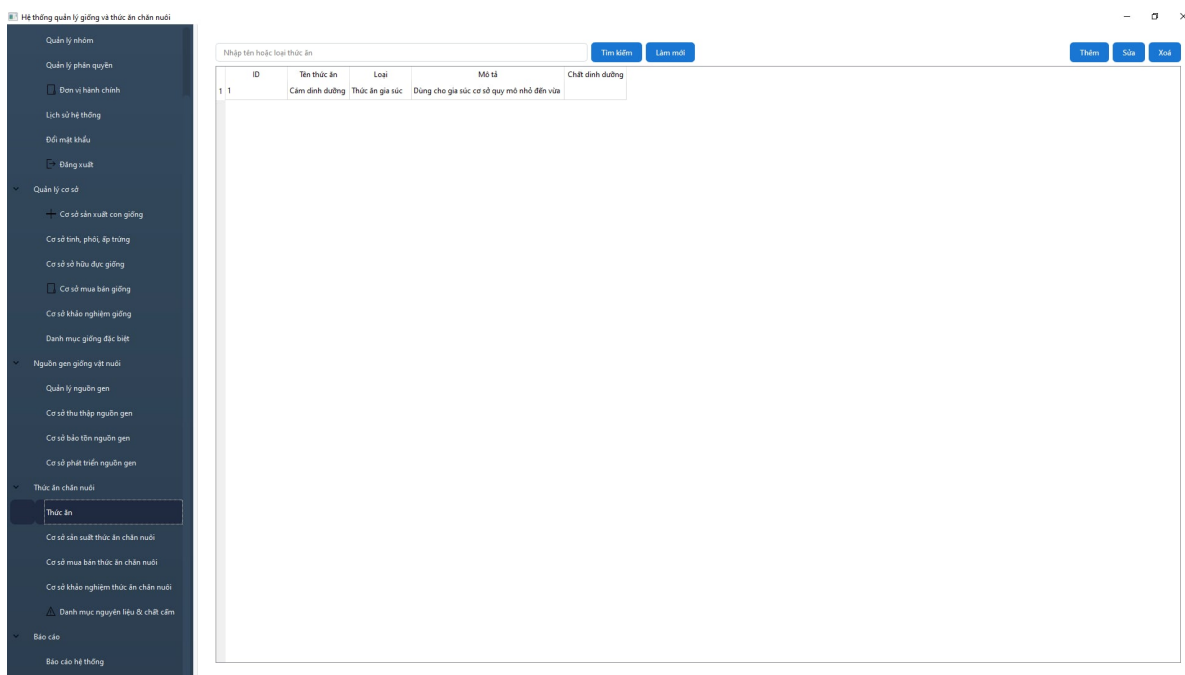
Hình 3.4.7: Phân quyền người dùng



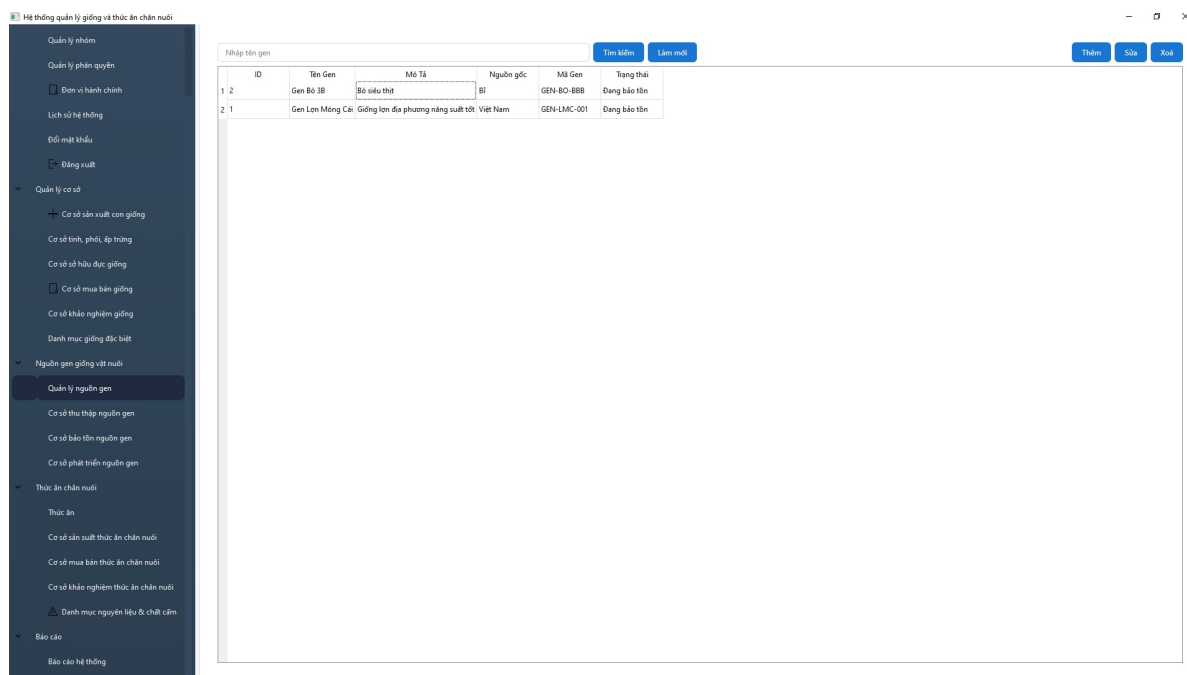
Hình 3.4.8: Đổi mật khẩu tài khoản



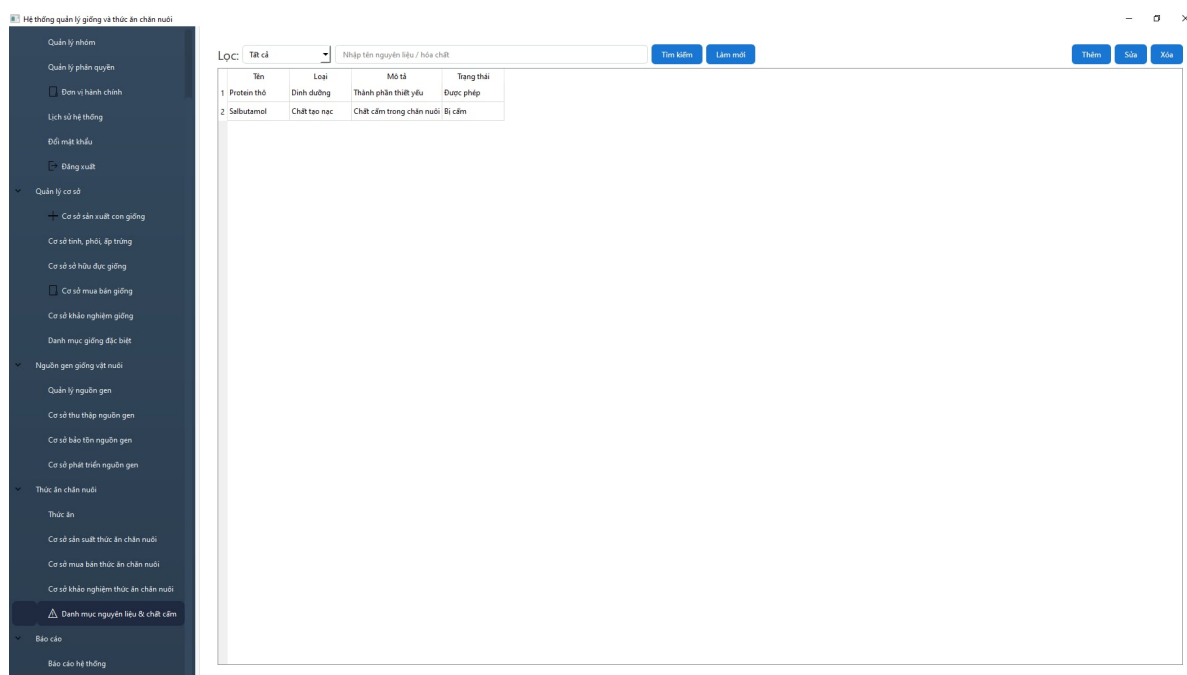
Hình 3.4.9: Quản lý CSDL cơ sở sản xuất thức ăn chăn nuôi



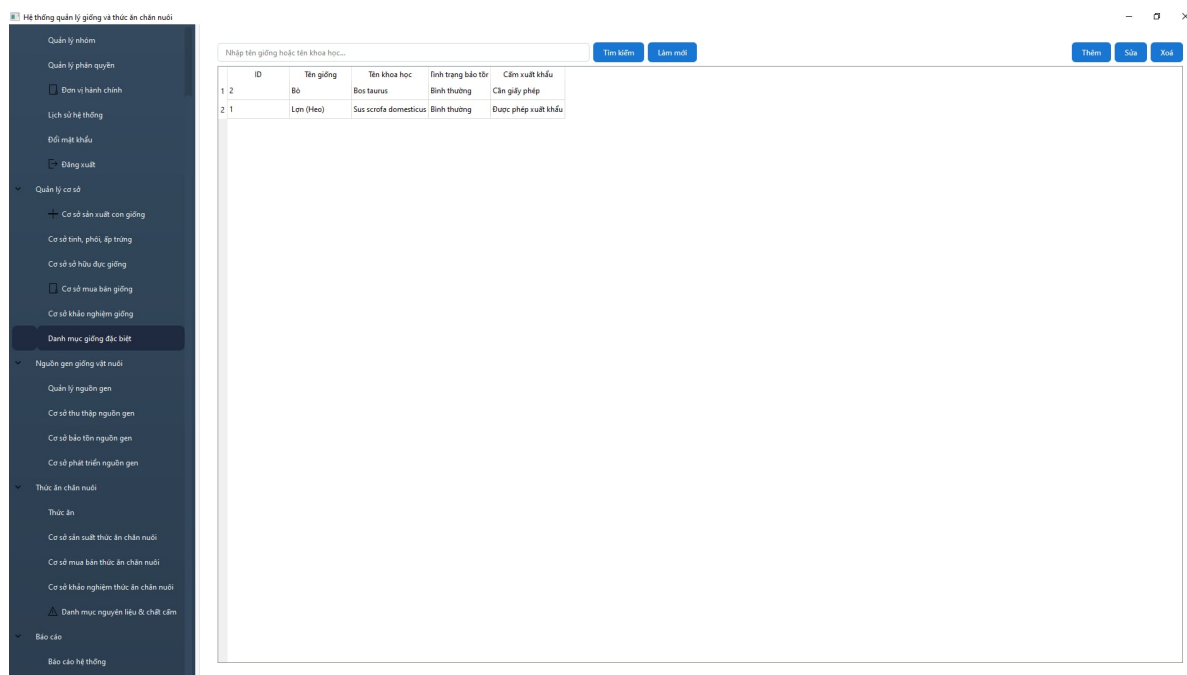
Hình 3.4.10: Quản lý CSDL thức ăn



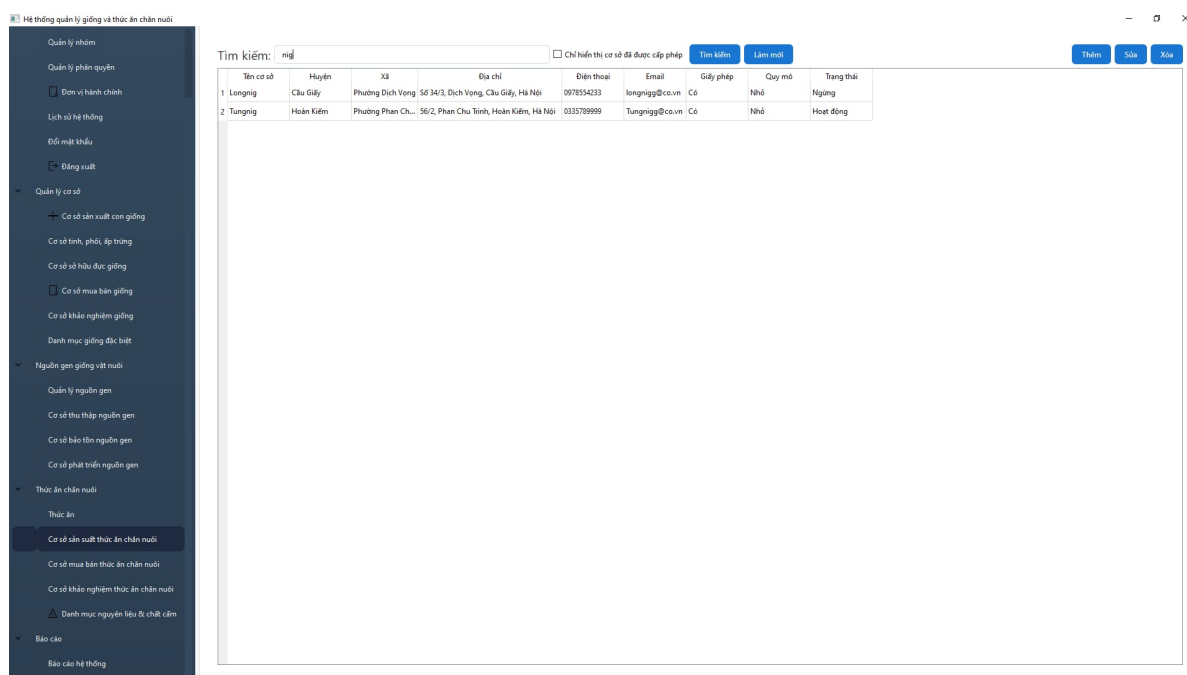
Hình 3.4.11: Quản lý CSDL gen



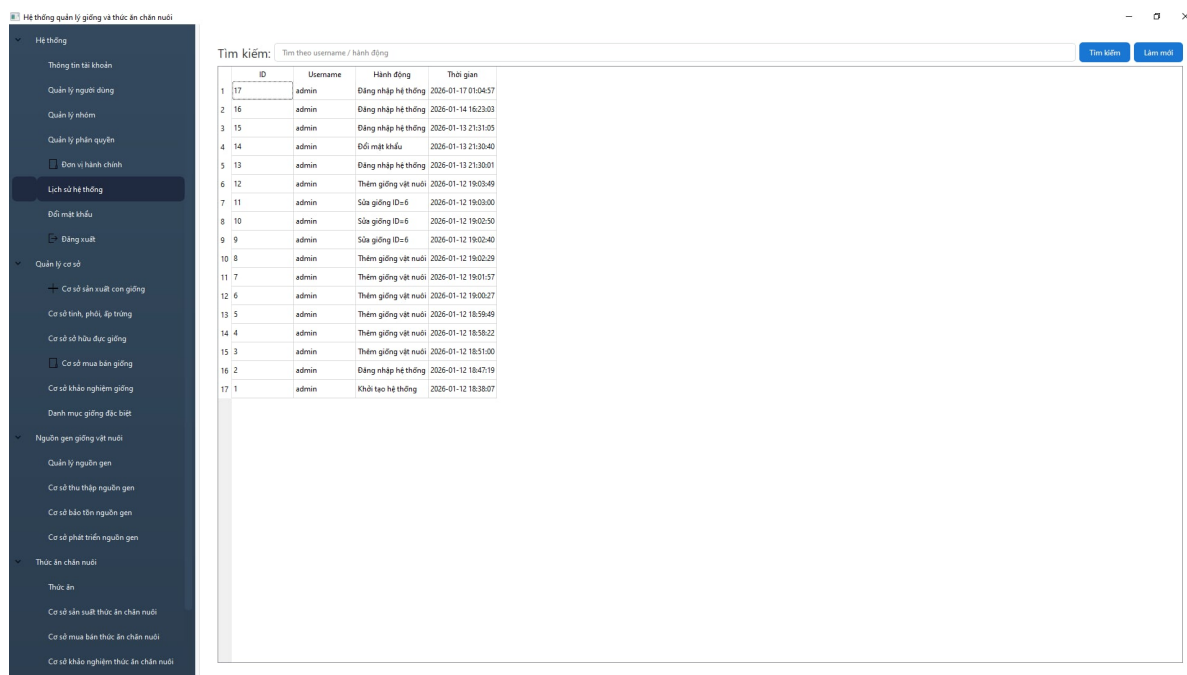
Hình 3.4.12: Quản lý CSDL nguyên liệu và chất cấm



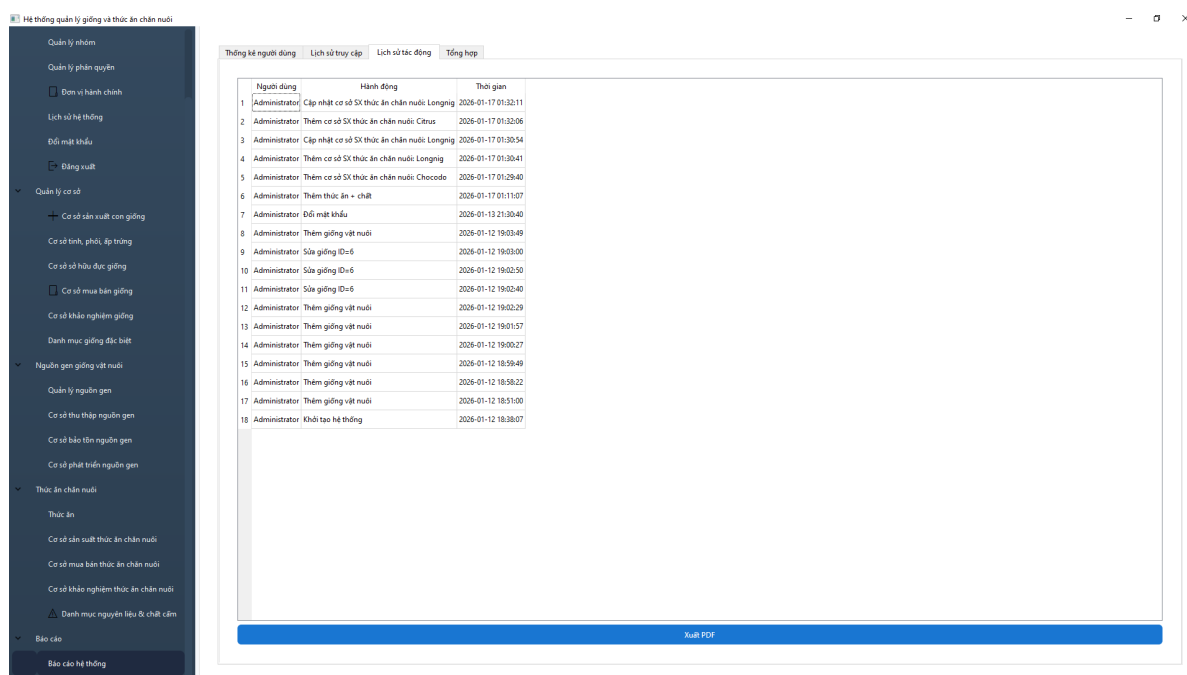
Hình 3.4.13: Quản lý CSDL giống vật nuôi



Hình 3.4.14: Chức năng tìm kiếm



Hình 3.4.15: Lịch sử hệ thống

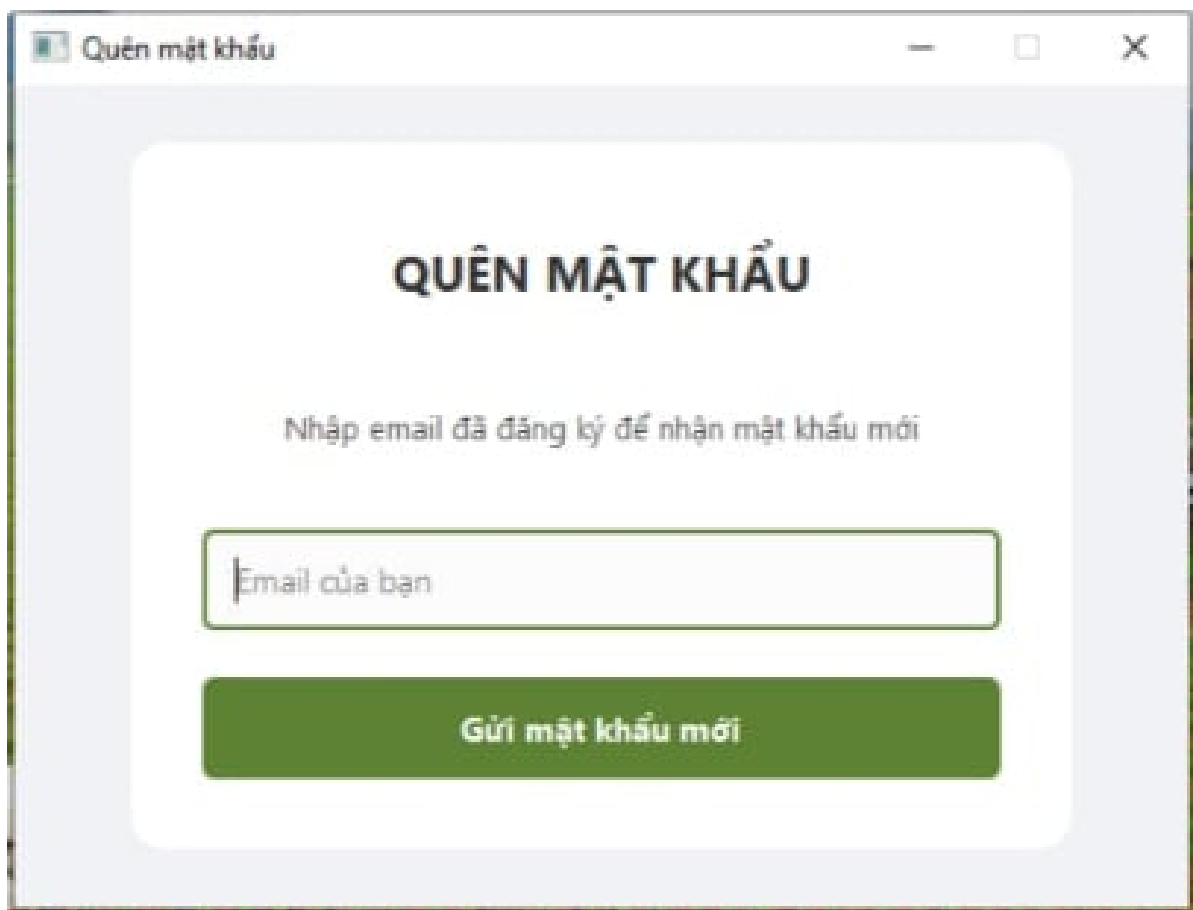


Hình 3.4.16: Thống kê báo cáo

Ngày xuất: 2026-01-17 01:37:10

Thông tin	Giá trị
Tổng số người dùng	4
Tổng số hành động	24
Tổng số hành động tạo/sửa/xóa	18

Hình 3.4.17: File báo cáo được xuất



The image shows a web browser window with the title 'Quên mật khẩu'. The main heading is 'QUÊN MẬT KHẨU'. Below it is a subheading 'Nhập email đã đăng ký để nhận mật khẩu mới'. There is a text input field with the placeholder text 'Email của bạn'. Below the input field is a green button with the text 'Gửi mật khẩu mới'.

Quên mật khẩu

QUÊN MẬT KHẨU

Nhập email đã đăng ký để nhận mật khẩu mới

Gửi mật khẩu mới

Hình 3.4.18: Quên mật khẩu

3.5. Kết luận chương

Chương 3 đã tổng kết toàn bộ quá trình hiện thực hóa hệ thống từ thiết kế dữ liệu đến triển khai ứng dụng. Kết quả đạt được bao gồm:

Cơ sở dữ liệu: Xây dựng thành công cơ sở dữ liệu MySQL chuẩn hóa, đảm bảo tính toàn vẹn cho dữ liệu về giống loài, phả hệ và lịch sử chăn nuôi.

Kiến trúc phần mềm: Ứng dụng mô hình MVC trên nền tảng Python và PyQt6 giúp mã nguồn mạch lạc, tách biệt rõ ràng giữa giao diện và xử lý, thuận tiện cho việc bảo trì và mở rộng.

Sản phẩm hoàn thiện: Phần mềm hoạt động ổn định, đáp ứng tốt các nghiệp vụ cốt lõi như quản lý hồ sơ vật nuôi, hệ thống báo cáo thống kê chi tiết,..

KẾT LUẬN

Trong quá trình thực hiện bài tập lớn, chúng em đã có cơ hội tiếp thu và tích lũy thêm nhiều kiến thức mới, đồng thời rèn luyện những kỹ năng quan trọng như làm việc nhóm, tư duy logic và giải quyết vấn đề. Dù hệ thống mà nhóm phát triển vẫn còn một số hạn chế, nhưng trải nghiệm này đã giúp chúng em hiểu sâu hơn về cấu trúc, nguyên lý hoạt động của lập trình phần mềm, cũng như nắm vững hơn ngôn ngữ và lập trình MySQL Server. Qua đó, tư duy lập trình của chúng em cũng được nâng cao một cách đáng kể.

Không chỉ dừng lại ở những kiến thức trong phạm vi môn học, bài tập lớn còn mở ra cho chúng em cơ hội khám phá và học hỏi nhiều kỹ năng mới. Chúng em đã cùng nhau nghiên cứu chủ đề, học cách xây dựng thư viện, sử dụng thành thạo các công cụ như Visual Studio Code, và phối hợp nhịp nhàng để hoàn thành dự án một cách hiệu quả. Những trải nghiệm này không chỉ giúp chúng em củng cố kiến thức chuyên môn mà còn rèn luyện khả năng làm việc nhóm và quản lý thời gian.

Cuối cùng, nhóm em xin gửi lời cảm ơn chân thành đến thầy Vũ Song Tùng vì sự hướng dẫn tận tình và sự hỗ trợ quý báu trong suốt quá trình thực hiện bài tập lớn. Do kiến thức và kinh nghiệm còn hạn chế, bài làm của chúng em chắc chắn không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và những góp ý quý báu từ thầy để chúng em có thể hoàn thiện bản thân và sản phẩm của mình hơn nữa trong tương lai.

Tài liệu tham khảo

Tài liệu

- [1] Python Software Foundation. *Python 3.12 Documentation*. [Trực tuyến].
Địa chỉ: <https://docs.python.org/3/>
- [2] Riverbank Computing. *PyQt6 Reference Guide*. [Trực tuyến].
Địa chỉ: <https://www.riverbankcomputing.com/static/Docs/PyQt6/>
- [3] Qt Group. *Qt for Python (PySide6) and Qt 6 Documentation*. [Trực tuyến].
Địa chỉ: <https://doc.qt.io/>
- [4] Oracle. *MySQL 8.0 Reference Manual*. [Trực tuyến].
Địa chỉ: <https://dev.mysql.com/doc/refman/8.0/en/>
- [5] MySQL. *MySQL Connector/Python Developer Guide*. [Trực tuyến].
Địa chỉ: <https://dev.mysql.com/doc/connector-python/en/>
- [6] Martin Fitzpatrick. *Create GUI Applications with Python & Qt6 (PyQt6 Edition)*. 5th Edition, 2021.
- [7] Mark Lutz. *Learning Python*. 5th Edition. O'Reilly Media, 2013.
- [8] Paul DuBois. *MySQL*. 5th Edition. Addison-Wesley Professional, 2013.
- [9] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [10] PyPi - The Python Package Index. *Thư viện mysql-connector-python*. [Trực tuyến].
Địa chỉ: <https://pypi.org/project/mysql-connector-python/>
- [11] PyPi - The Python Package Index. *Thư viện PyQt6*. [Trực tuyến].
Địa chỉ: <https://pypi.org/project/PyQt6/>