

**KHOA KHOA HỌC MÁY TÍNH VÀ KỸ THUẬT**

**BÁO CÁO ĐỒ ÁN**

**MÔN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (OPP)**

**NĂM 2024-2025**

**TÊN CHỦ ĐỀ:**

**SPY AGENCY 2024**

**Sinh Viên:**

Đường Tri Nhân - 97482403193

**Lớp:** 24CNTT

**Giảng Viên:**

**Tiến SĨ Lê Ngọc Hiếu**

**THÁNG 5, 2024-2025**

Mục Lục

[Bảng Phân Công Nhiệm Vụ 4](#_Toc11817)

[Danh Mục Từ Viết Tắt 5](#_Toc14897)

[Phần Mở Đầu 6](#_Toc4301)

[1. Lý do chọn đề tài. 6](#_Toc19854)

[2. Mục đích nghiên cứu. 7](#_Toc898)

[3. Mục tiêu nghiên cứu 7](#_Toc21294)

[4. Mô tả tổng quát về hệ thống xây dựng 7](#_Toc11520)

[Danh Mục Từ Viết Tắt 10](#_Toc11447)

[PHẦN NỘI DUNG 11](#_Toc3833)

[Chương 1: Cơ sở lý thuyết và thực trạng nghiên cứu 11](#_Toc26925)

[1.1. Cơ sở khoa học 11](#_Toc10261)

[1.1.1. Tổng quan về lập trình hướng đối tượng 11](#_Toc20865)

[1.1.2. Công nghệ Java Swing trong phát triển ứng dụng desktop 12](#_Toc30403)

[1.1.3. Hệ quản trị cơ sở dữ liệu MySQL và ngôn ngữ SQL 13](#_Toc26274)

[1.1.4. Cấu trúc dữ liệu và Java Collections Framework 13](#_Toc27521)

[1.2. Thực trạng nghiên cứu 14](#_Toc17693)

[1.2.1. Thực trạng công nghệ hiện đại 14](#_Toc3732)

[1.2.2. Phân tích các ứng dụng và giải pháp quản lý – phân tích dữ liệu tương tự 15](#_Toc17022)

[Chương 2: Giải pháp đề xuất và phân tích thiết kế ứng dụng 17](#_Toc31809)

[2.1. Giải pháp đề xuất 17](#_Toc31518)

[2.1.1. Mô hình tổng quát 17](#_Toc13318)

[2.1.2. Yêu cầu chức năng 18](#_Toc31489)

[2.1.3. Yêu cầu phi chức năng 20](#_Toc23142)

[2.2.1. Kiến trúc hệ thống 21](#_Toc17129)

[2.2.2 Phân tích thiết kế giao diện người dùng 23](#_Toc19007)

[2.3 Thông tin giao diện........................................................................27](#_Toc13008)

[2.4 Cấu trúc cơ sở dữ liệu MySQL........ ....30](#_Toc31184)

[📑 2.4.1 Mô tả bảng dữ liệu 30](#_Toc28006)

[📑 2.4.2. Ràng buộc và Quan hệ 33](#_Toc6350)

[CHƯƠNG 3: CƠ CHẾ HOẠT ĐỘNG CỦA HỆ THỐNG 35](#_Toc9287)

[3.1 Cấu trúc tổng thể 35](#_Toc14394)

[3.2 Lớp Model 36](#_Toc31484)

[3.3 Lớp DAO (Data Access Object) 36](#_Toc4567)

[3.4 Lớp Controller 37](#_Toc24336)

[3.5 Lớp View (Java SWing) 37](#_Toc29815)

[3.6 Lớp Chart và ChartDAO 38](#_Toc26045)

[3.7 Cấu hình kết nối (Config) 38](#_Toc17264)

[3.8 Xử lý xác thực người dùng 39](#_Toc1013)

[CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ 39](#_Toc11002)

[4.1 Môi trường và Quy trình Thử nghiệm 39](#_Toc23517)

[4.2 Kết quả Kiểm thử 40](#_Toc30794)

[4.3 Đánh giá Sản phẩm 41](#_Toc29426)

[KẾT LUẬN 43](#_Toc15260)

[1. Xây dựng thành công hệ thống quản lý người dùng và xác thực (User Management & Authentication) 43](#_Toc32361)

[2. Phát triển module quản lý dữ liệu điệp viên và các thực thể liên quan (Data Management) 44](#_Toc19348)

[3. Triển khai chức năng trực quan hóa dữ liệu bằng biểu đồ (Data Visualization) 44](#_Toc8079)

[4. Ứng dụng hiệu quả các kỹ thuật Lập trình Hướng đối tượng (OOP) và công nghệ Java 44](#_Toc15514)

[5. Hoàn thiện cấu trúc dự án và hệ thống tài liệu 45](#_Toc30773)

[TÀI LIỆU THAM KHẢO 46](#_Toc13340)

**Bảng Phân Công Nhiệm Vụ**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Công Việc** | **Ghi Chú** |
| **1** | **Đường Tri Nhân 97482403193** | **Phụ trách toàn bộ đồ án** |  |

**Danh Mục Từ Viết Tắt**

|  |  |
| --- | --- |
| **Abbreviation** | **Explanation** |
| **DAO** | **Data Access Object** |
| **CRUD** | **Create, Read, Update, Delete** |
| **DB** | **Date Base** |
| **OOP** | **Object-Oriented Programming** |
| **UI** | **User Interface** |
| **MVC** | **Model - View - Controller** |
| **PK** | **Primary Key** |
| **JDBC** | **Java DataBase Connectivity** |

**Phần Mở Đầu**

1. **Lý do chọn đề tài.**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng phần mềm vào quản lý, lưu trữ và phân tích dữ liệu đang trở thành xu hướng tất yếu của nhiều lĩnh vực, đặc biệt là đối với những ngành nghề đòi hỏi tính bảo mật, chính xác và tốc độ xử lý cao. Trong đó, lĩnh vực tình báo là một ví dụ điển hình khi việc quản lý số lượng lớn các dữ liệu về điệp viên, nhiệm vụ, kỹ năng, tổ chức và các mối liên hệ phức tạp luôn là một bài toán khó và mang tính chất đặc thù.

Tuy nhiên, thực tế cho thấy hiện nay chưa có nhiều ứng dụng quản lý dữ liệu điệp viên đơn giản, dễ sử dụng mà vẫn đáp ứng đầy đủ các chức năng cần thiết cho việc quản lý, thống kê và trực quan hóa dữ liệu một cách trực tiếp trên giao diện desktop. Đa số các hệ thống hiện có thường phức tạp, khó tiếp cận hoặc thiếu các công cụ trực quan giúp người dùng theo dõi và đánh giá thông tin một cách nhanh chóng, hiệu quả.

Chính vì vậy, nhóm em lựa chọn thực hiện đề tài **“SpyAgency2024”** với mong muốn phát triển một phần mềm desktop quản lý dữ liệu tình báo có giao diện trực quan, dễ sử dụng, tích hợp đầy đủ các chức năng quản lý thông tin điệp viên, nhiệm vụ, kỹ năng, đội nhóm và các mối liên kết giữa các thành phần trong tổ chức tình báo. Ứng dụng không chỉ dừng lại ở việc lưu trữ, tra cứu thông tin mà còn hỗ trợ **thống kê và trực quan hóa dữ liệu** bằng biểu đồ thông qua thư viện **JFreeChart**, giúp người dùng dễ dàng theo dõi số liệu, xu hướng và tình hình hoạt động của các điệp viên cũng như toàn bộ hệ thống tình báo.

Đặc biệt, đề tài sử dụng **Java Swing** – một thư viện giao diện đồ họa mạnh mẽ và phổ biến của Java để xây dựng giao diện phần mềm, kết hợp với **MySQL** để lưu trữ dữ liệu và **OOP (Lập trình hướng đối tượng)** để thiết kế cấu trúc chương trình theo các lớp riêng biệt như **Model – DAO – Controller – Chart – ChartDAO – View – Config – Icon**. Qua đó, dự án giúp nhóm sinh viên rèn luyện kỹ năng thiết kế và xây dựng phần mềm theo mô hình nhiều lớp, đảm bảo tính mở rộng, dễ bảo trì và dễ nâng cấp trong tương lai.

Việc thực hiện đề tài này không những giúp nhóm củng cố và nâng cao kiến thức lập trình Java, kỹ năng xây dựng giao diện desktop bằng Java Swing, thao tác với cơ sở dữ liệu MySQL mà còn giúp nhóm thành thạo các kỹ thuật xử lý và trực quan hóa dữ liệu bằng biểu đồ, phục vụ cho công tác phân tích và quản lý dữ liệu thực tế. Đồng thời, sản phẩm của đề tài cũng có thể được ứng dụng vào thực tiễn hoặc làm nền tảng tham khảo để phát triển các phần mềm quản lý dữ liệu cho các tổ chức, doanh nghiệp hoặc lĩnh vực khác có nhu cầu tương tự.

Với những lý do trên, nhóm em tin rằng đề tài **“SpyAgency2024”** là một lựa chọn thiết thực, phù hợp với yêu cầu môn học và có giá trị ứng dụng thực tiễn cao.

1. **Mục đích nghiên cứu.**

Đề tài **“SpyAgency2024”** được thực hiện với mục đích xây dựng một ứng dụng desktop bằng **Java Swing** phục vụ cho việc **quản lý, lưu trữ và phân tích dữ liệu tình báo** một cách khoa học, hệ thống và hiệu quả. Phần mềm giúp đơn giản hóa quá trình quản lý thông tin về điệp viên, nhiệm vụ, kỹ năng và các tổ chức tình báo, đồng thời hỗ trợ người dùng thực hiện các thao tác thống kê, tìm kiếm và trực quan hóa dữ liệu nhanh chóng, trực quan thông qua các biểu đồ sinh động.

Thông qua đề tài này, nhóm sinh viên mong muốn **áp dụng những kiến thức lý thuyết đã học về lập trình hướng đối tượng (OOP), Java Swing và quản lý cơ sở dữ liệu MySQL vào một bài toán thực tế**, góp phần nâng cao khả năng thiết kế và phát triển phần mềm, đồng thời rèn luyện kỹ năng làm việc nhóm, tổ chức và triển khai dự án phần mềm hoàn chỉnh.

1. **Mục tiêu nghiên cứu**

**- Thiết kế và xây dựng hệ thống cơ sở dữ liệu** lưu trữ thông tin về điệp viên, nhiệm vụ, kỹ năng, tổ chức và các mối liên kết một cách logic, chặt chẽ và đảm bảo tính mở rộng.

- **Phát triển giao diện đồ họa desktop thân thiện với người dùng** bằng **Java Swing**, đảm bảo dễ sử dụng, trực quan, hỗ trợ đầy đủ các chức năng cần thiết cho việc quản lý dữ liệu.

- **Xây dựng hệ thống phân quyền người dùng** với chức năng đăng nhập bảo mật bằng mật khẩu băm hóa SHA-256.

- **Triển khai đầy đủ các chức năng quản lý dữ liệu** (CRUD: Thêm – Sửa – Xóa – Tìm kiếm) cho các bảng thông tin điệp viên, nhiệm vụ, kỹ năng, tổ chức, nhóm, và các bảng liên kết.

- **Tích hợp tính năng thống kê và trực quan hóa dữ liệu** bằng thư viện **JFreeChart.**

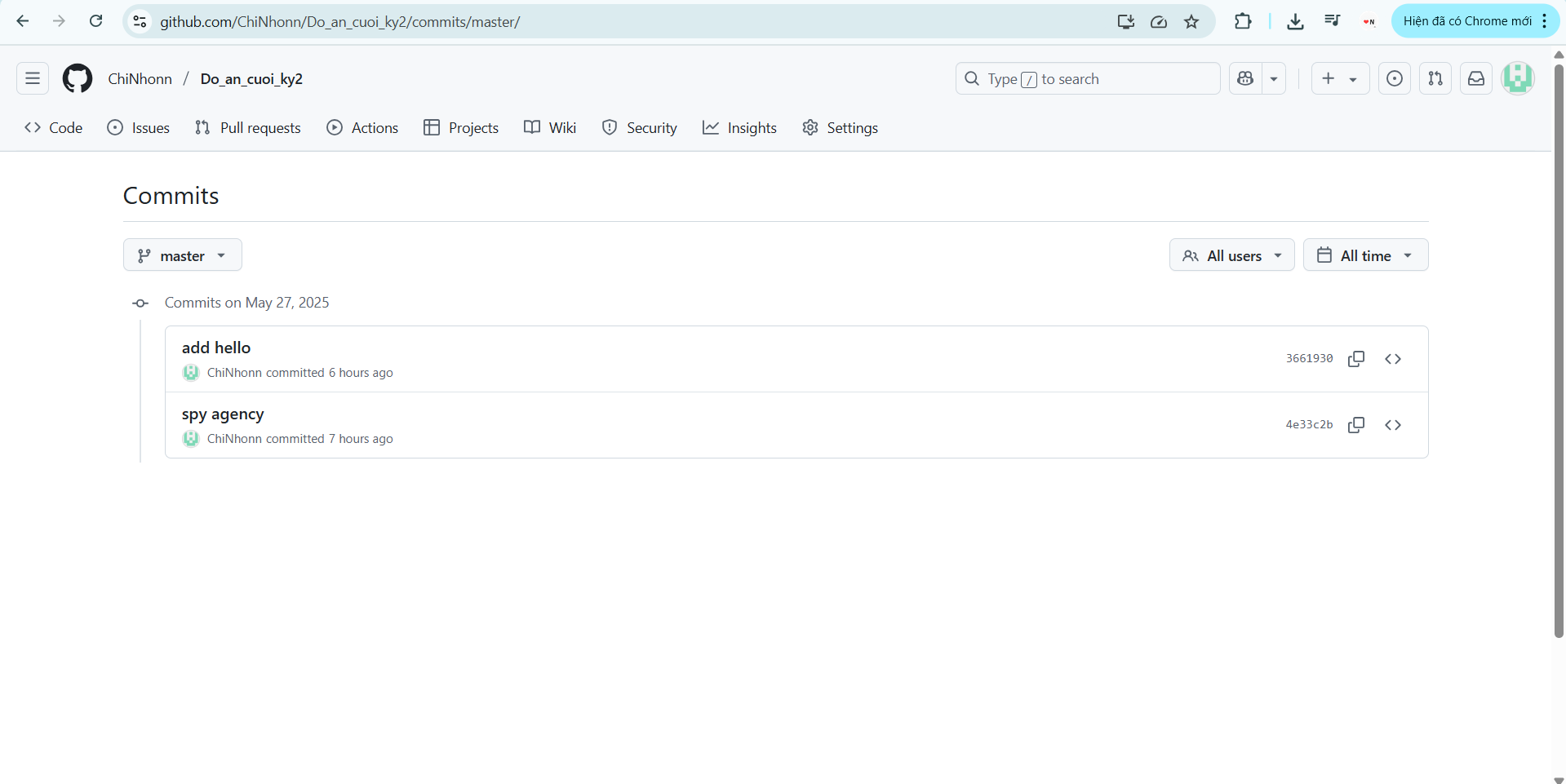
**- Tuân thủ nguyên lý lập trình hướng đối tượng (OOP)** và áp dụng mô hình phân lớp phần mềm hợp lý: **Model – DAO – Controller – Chart – ChartDAO – View – Config – Icon**. nhằm đảm bảo tính dễ bảo trì, dễ mở rộng và tái sử dụng cho hệ thống

1. **Mô tả tổng quát về hệ thống xây dựng**

Hệ thống **Spy Agency Management** được xây dựng dưới dạng một ứng dụng desktop sử dụng ngôn ngữ lập trình **Java**, với giao diện người dùng (**GUI**) thiết kế bằng **Java Swing** và cơ sở dữ liệu lưu trữ trên **MySQL**.  
Hệ thống này cho phép quản lý dữ liệu điệp viên, nhiệm vụ, nhóm, kỹ năng, ngoại ngữ, đơn vị liên kết và cấp độ bảo mật một cách hiệu quả và trực quan.

Các chức năng trực quan hóa số liệu phân tích thông qua **biểu đồ thống kê bằng thư viện JFreeChart** giúp ban lãnh đạo nhanh chóng nắm được tình hình phân công nhiệm vụ, số lượng điệp viên, mức độ an toàn, sự phân bổ nhân sự trong các chiến dịch tình báo, giảm thiểu sai sót, tránh gian lận, nâng cao hiệu quả công việc quản lý.

**Có sử dụng Github:**



**PHẦN NỘI DUNG**

**Chương 1: Cơ sở lý thuyết và thực trạng nghiên cứu**

### ****1.1. Cơ sở khoa học****

#### ****1.1.1. Tổng quan về lập trình hướng đối tượng****

**Lập trình hướng đối tượng (OOP)** là một phương pháp lập trình dựa trên khái niệm **“đối tượng”**, nơi các đối tượng này chứa dữ liệu (gọi là **thuộc tính**) và các hành động xử lý dữ liệu đó (gọi là **phương thức**). OOP mang lại cách tiếp cận trực quan, có cấu trúc và dễ quản lý, đặc biệt phù hợp với những hệ thống có dữ liệu phức tạp và nhiều đối tượng tương tác lẫn nhau.

Trong quá trình phát triển đề tài **SpyAgency2024**, nhóm đã áp dụng đầy đủ các nguyên lý cơ bản của lập trình hướng đối tượng để thiết kế và tổ chức mã nguồn một cách hợp lý, dễ mở rộng và bảo trì. Cụ thể:

**+ Tính đóng gói (Encapsulation):**  
Đảm bảo các dữ liệu quan trọng của đối tượng được bảo vệ khỏi các tác động không mong muốn từ bên ngoài. Các lớp **Model** như Agent, Mission, Skill, Team... sẽ khai báo các thuộc tính ở mức **private**, và cung cấp các phương thức **getter/setter** công khai để truy cập dữ liệu một cách an toàn.

**+ Tính kế thừa (Inheritance):**  
Cho phép các lớp con kế thừa các thuộc tính và phương thức chung từ lớp cha, giúp giảm thiểu việc lặp lại mã nguồn và thiết lập mối quan hệ phân cấp rõ ràng giữa các đối tượng. Ví dụ, trong đề tài có thể xây dựng một lớp BaseDAO chứa các phương thức thao tác cơ bản với cơ sở dữ liệu, và các lớp DAO như AgentDAO, MissionDAO kế thừa lại để tái sử dụng và mở rộng.

**+ Tính đa hình (Polymorphism):**  
Giúp các đối tượng khác nhau có thể sử dụng chung một giao diện xử lý nhưng thực hiện theo cách riêng của mình. Tính năng này được thể hiện thông qua cơ chế **ghi đè phương thức (overriding)** và **nạp chồng phương thức (overloading)**

**+ Tính trừu tượng (Abstraction):**  
Cho phép lập trình viên tập trung vào những thành phần cốt lõi, che giấu các chi tiết triển khai phức tạp phía sau. Các lớp trừu tượng hoặc giao diện sẽ định nghĩa những hành vi cần có, còn chi tiết thực thi sẽ do các lớp con thực hiện. Điều này giúp đơn giản hóa hệ thống, tăng khả năng mở rộng và linh hoạt trong phát triển phần mềm.

Việc áp dụng đầy đủ các nguyên lý của lập trình hướng đối tượng vào đề tài **SpyAgency2024** không chỉ giúp mã nguồn được tổ chức rõ ràng, dễ đọc, dễ bảo trì mà còn tạo nền tảng vững chắc để nhóm có thể phát triển thêm nhiều chức năng mới trong tương lai mà không ảnh hưởng đến cấu trúc hệ thống hiện có.

**1.1.2. Công nghệ Java Swing trong phát triển ứng dụng desktop**

**Java Swing** là một thư viện giao diện người dùng mạnh mẽ thuộc Java Standard Edition, được sử dụng rộng rãi để phát triển các ứng dụng desktop. Swing cung cấp tập hợp đa dạng các thành phần giao diện (components) như button, label, textbox, table, panel… cùng với khả năng tùy biến giao diện, xử lý sự kiện linh hoạt, giúp lập trình viên xây dựng các phần mềm có giao diện thân thiện, dễ sử dụng.

Trong đề tài **SpyAgency2024**, Java Swing đóng vai trò là công nghệ chính để thiết kế và phát triển giao diện cho phần mềm quản lý điệp viên. Các đặc điểm nổi bật được áp dụng trong quá trình phát triển gồm:

**+ Bộ thành phần giao diện phong phú:** Swing cung cấp đầy đủ các thành phần như JButton, JLabel, JTextField, JComboBox, JTable, JScrollPane… giúp nhóm thiết kế các giao diện nhập liệu, hiển thị bảng dữ liệu và bảng điều khiển một cách linh hoạt và hiệu quả.

**+ Quản lý giao diện theo Layout:** Swing cho phép sắp xếp các thành phần giao diện theo nhiều dạng bố cục như BorderLayout, FlowLayout, GridLayout, giúp thiết kế các cửa sổ làm việc rõ ràng, gọn gàng.

**+ Xử lý sự kiện (Event Handling):** Ứng dụng tận dụng cơ chế bắt sự kiện trong Swing để xử lý các thao tác của người dùng như nhấn nút, chọn bảng, tìm kiếm dữ liệu, tạo biểu đồ…

**+ Tùy biến giao diện:** Java Swing cho phép thay đổi màu sắc, font chữ, viền, biểu tượng… giúp tạo nên giao diện đồng bộ, dễ sử dụng và trực quan.

**+ Tích hợp với thư viện trực quan hóa dữ liệu:** Mặc dù không có SceneBuilder như JavaFX, nhưng Swing hoàn toàn tương thích với các thư viện như **JFreeChart**, giúp nhúng biểu đồ vào ứng dụng một cách dễ dàng.

#### ****1.1.3. Hệ quản trị cơ sở dữ liệu MySQL và ngôn ngữ SQL****

Trong đề tài **SpyAgency2024**, hệ thống sử dụng **MySQL** làm hệ quản trị cơ sở dữ liệu chính, được triển khai thông qua **XAMPP** — một phần mềm giả lập máy chủ phổ biến, tích hợp Apache, MySQL, PHP và phpMyAdmin giúp dễ dàng cài đặt và quản lý cơ sở dữ liệu ngay trên máy tính cá nhân.

**+ Quản lý dữ liệu dạng bảng:** Dữ liệu của hệ thống được tổ chức và lưu trữ trong các bảng (table), mỗi bảng gồm nhiều hàng (record) và cột (field). Các bảng được liên kết với nhau bằng **khóa chính (Primary Key)** và **khóa ngoại (Foreign Key)** để đảm bảo tính toàn vẹn và quan hệ logic giữa các thực thể như điệp viên, nhiệm vụ, kỹ năng, tổ chức.

**+ Truy vấn dữ liệu bằng SQL:** Hệ thống sử dụng ngôn ngữ SQL để thực hiện các thao tác với cơ sở dữ liệu như thêm mới (INSERT), sửa (UPDATE), xóa (DELETE), tìm kiếm (SELECT) và thống kê dữ liệu từ các bảng.

**+ Quản lý cơ sở dữ liệu bằng phpMyAdmin (XAMPP):** Cùng với MySQL Workbench, nhóm sử dụng **phpMyAdmin** tích hợp trong XAMPP để tạo bảng, chỉnh sửa dữ liệu và kiểm tra nhanh các câu lệnh SQL trong quá trình phát triển.

**+ Kết nối với Java thông qua JDBC:** Để phần mềm Java Swing giao tiếp với cơ sở dữ liệu, nhóm sử dụng **JDBC (Java Database Connectivity)** — một API chuẩn của Java giúp kết nối và thực thi các câu lệnh SQL. Lớp kết nối được tổ chức riêng giúp quản lý và tái sử dụng trong toàn bộ dự án.

Việc sử dụng **XAMPP** mang lại nhiều lợi ích như dễ cài đặt, dễ quản lý cơ sở dữ liệu trên localhost và phù hợp với quy mô dự án desktop nội bộ như **SpyAgency2024**.

#### ****1.1.4. Cấu trúc dữ liệu và Java Collections Framework****

Trong quá trình phát triển ứng dụng **SpyAgency2024**, việc lựa chọn các cấu trúc dữ liệu phù hợp đóng vai trò quan trọng giúp tối ưu hóa hiệu suất và đơn giản hóa việc xử lý dữ liệu. Dự án đã tận dụng **JFreeChart -** là thư viện Java mã nguồn mở hỗ trợ tạo các loại biểu đồ 2D chuyên nghiệp. Trong đề tài **SpyAgency2024**, nhóm sử dụng JFreeChart để trực quan hóa số liệu tình báo và kết quả thống kê.

**Các loại biểu đồ được sử dụng:**

**+ Biểu đồ cột (Bar Chart):** Thể hiện top điệp viên có nhiều nhiệm vụ nhất.

**+ Biểu đồ tròn (Pie Chart):** Thống kê tỷ lệ quốc tịch của điệp viên.

**+ Biểu đồ đường (Line Chart):** Theo dõi số lượng nhiệm vụ qua từng năm.

**Dataset:** Dữ liệu cho biểu đồ được tổ chức dưới dạng DefaultCategoryDataset cho biểu đồ cột/đường và DefaultPieDataset cho biểu đồ tròn, giúp trực quan hóa thông tin một cách hiệu quả.

**Tùy chỉnh biểu đồ:** JFreeChart hỗ trợ tùy chỉnh tiêu đề, nhãn trục, màu sắc, font chữ và kích thước giúp tạo nên các biểu đồ đẹp mắt, rõ ràng.

**Tích hợp với Java Swing:** Các biểu đồ được nhúng trực tiếp vào Swing qua **ChartPanel**, cho phép hiển thị biểu đồ cùng với các bảng dữ liệu, form quản lý trong cùng một cửa sổ làm việc.

**Nguyên tắc trực quan hóa hiệu quả:** Nhóm chú trọng việc chọn loại biểu đồ phù hợp với từng loại dữ liệu, sử dụng màu sắc dễ phân biệt và bố cục hợp lý nhằm giúp người dùng dễ dàng nắm bắt thông tin mà không bị rối mắt.

### ****1.2. Thực trạng nghiên cứu****

#### ****1.2.1. Thực trạng công nghệ hiện đại****

Trong thời đại chuyển đổi số mạnh mẽ hiện nay, công nghệ thông tin đã và đang phát triển không ngừng, mang lại nhiều công cụ và nền tảng hỗ trợ đắc lực cho việc thu thập, quản lý, phân tích và trực quan hóa dữ liệu trong nhiều lĩnh vực, đặc biệt là lĩnh vực an ninh và tình báo. Các công nghệ và nền tảng đáng chú ý bao gồm:

**Nền tảng phát triển ứng dụng desktop:**  
**Java**, đặc biệt với thư viện **Java Swing**, vẫn là một công cụ phổ biến và ổn định trong việc xây dựng các ứng dụng desktop nhờ khả năng xử lý dữ liệu tốt, giao diện thân thiện và dễ tích hợp với các thư viện bên ngoài. Ngoài ra, các nền tảng như **.NET (C#)** hay **Python (Tkinter, PyQt)** cũng được sử dụng nhiều trong phát triển phần mềm quản lý dữ liệu.

**Hệ quản trị cơ sở dữ liệu:**  
**MySQL**, **PostgreSQL** và **SQL Server** tiếp tục là những lựa chọn phổ biến cho việc lưu trữ dữ liệu. Bên cạnh đó, các hệ quản trị cơ sở dữ liệu NoSQL như **MongoDB** cũng dần được áp dụng vào các hệ thống cần xử lý dữ liệu linh hoạt và phi cấu trúc.

**Thư viện trực quan hóa dữ liệu:**  
Ngoài **JFreeChart** dành cho Java, các thư viện như **Matplotlib**, **Seaborn** (Python), **D3.js** (JavaScript), cùng với các nền tảng phân tích dữ liệu và BI chuyên sâu như **Power BI**, **Tableau** cũng được sử dụng rộng rãi để trực quan hóa và phân tích dữ liệu. Tuy nhiên, đa phần các công cụ này thường phục vụ mục đích phân tích tổng quát, ít khi được triển khai cho các ứng dụng tình báo chuyên biệt.

**Công cụ phát triển và quản lý dự án:**  
Các môi trường lập trình hiện đại như **NetBeans**, **IntelliJ IDEA**, **Eclipse** hỗ trợ tốt cho phát triển ứng dụng Java Swing. Ngoài ra, công cụ quản lý cơ sở dữ liệu như **phpMyAdmin** (trong **XAMPP**) và **MySQL Workbench** giúp thiết kế và kiểm tra cơ sở dữ liệu trực quan. Việc quản lý mã nguồn bằng **Git** và xây dựng dự án với **Maven** cũng trở thành tiêu chuẩn phổ biến.

**Nhận định:**  
Với sự phát triển mạnh mẽ của các công nghệ hiện đại hiện nay, việc xây dựng một phần mềm quản lý và phân tích dữ liệu tình báo chuyên sâu như **SpyAgency2024** là hoàn toàn khả thi. Tuy nhiên, bài toán đặt ra là lựa chọn đúng công nghệ và tích hợp các thành phần một cách tối ưu để đáp ứng đầy đủ nhu cầu nghiệp vụ và dễ dàng mở rộng trong tương lai.

#### ****1.2.2. Phân tích các ứng dụng và giải pháp quản lý – phân tích dữ liệu tương tự****

Hiện nay, một số hệ thống và ứng dụng đã được phát triển phục vụ cho công tác quản lý và phân tích dữ liệu tình báo hoặc dữ liệu tổ chức nhân sự tương tự, tuy nhiên vẫn còn tồn tại những hạn chế nhất định:

**Các phần mềm quản lý nhân sự hoặc tổ chức chung:**  
Có nhiều phần mềm hỗ trợ quản lý nhân sự, tổ chức, tuy nhiên các phần mềm này thường chỉ phục vụ nhu cầu quản lý thông tin cá nhân, công việc mà chưa có khả năng thống kê, phân tích hoặc trực quan hóa dữ liệu chuyên sâu phục vụ cho lĩnh vực tình báo.

**Ứng dụng di động hoặc web-based:**  
Một số ứng dụng dạng mobile hoặc web hỗ trợ lưu trữ và cập nhật thông tin về nhân sự, tuy nhiên không nhiều hệ thống tập trung vào thống kê dữ liệu nhiệm vụ và mối quan hệ giữa các cá nhân trong tổ chức tình báo, cũng như thiếu tính năng trực quan hóa dữ liệu để hỗ trợ phân tích.

**Công cụ Business Intelligence (BI):**  
Các phần mềm như **Power BI**, **Tableau** hay **Google Data Studio** có thể kết nối và phân tích dữ liệu từ nhiều nguồn khác nhau. Tuy nhiên, chúng đòi hỏi người sử dụng phải có kiến thức về BI và kỹ năng kỹ thuật để cấu hình kết nối và thiết kế dashboard. Đồng thời, đây là những công cụ phân tích tổng quát, chưa được “đóng gói” sẵn cho các tổ chức tình báo hay bộ phận an ninh nội bộ.

**Các dự án mã nguồn mở:**  
Một số cộng đồng lập trình hoặc các nhóm nghiên cứu có phát triển các dự án quản lý thông tin nhân sự hoặc dữ liệu tình báo ở quy mô nhỏ. Tuy nhiên, các dự án này thường mang tính thử nghiệm hoặc học thuật, chưa được tối ưu hóa giao diện và thiếu tính năng trực quan hóa dữ liệu trực tiếp trong phần mềm desktop.

**Tổng kết:**  
 Từ thực trạng trên cho thấy, chưa có nhiều giải pháp quản lý và phân tích dữ liệu tình báo thực sự hoàn chỉnh và thân thiện với người dùng phổ thông. Chính vì vậy, đề tài **SpyAgency2024** hướng tới xây dựng một ứng dụng desktop đơn giản, dễ sử dụng, tích hợp đầy đủ tính năng quản lý dữ liệu điệp viên, nhiệm vụ và trực quan hóa số liệu thông qua biểu đồ ngay trong phần mềm, đồng thời đảm bảo khả năng mở rộng và bảo mật dữ liệu nội bộ.

## ****Chương 2: Giải pháp đề xuất và phân tích thiết kế ứng dụng****

### ****2.1. Giải pháp đề xuất****

Ứng dụng **“SpyAgency2024”** được đề xuất xây dựng dưới dạng **phần mềm desktop**, sử dụng **ngôn ngữ lập trình Java** với thư viện **Java Swing** để thiết kế giao diện người dùng (GUI). Dữ liệu về điệp viên, nhiệm vụ, kỹ năng, tổ chức và các mối liên kết sẽ được lưu trữ và quản lý trong cơ sở dữ liệu **MySQL**. Cơ sở dữ liệu được triển khai thông qua **XAMPP** và quản lý bằng công cụ **phpMyAdmin**.

Chức năng thống kê và trực quan hóa dữ liệu tình báo sẽ được thực hiện bằng thư viện **JFreeChart**, tích hợp trực tiếp vào ứng dụng Swing thông qua **ChartPanel** tối ưu hóa quy trình build.

#### ****2.1.1. Mô hình tổng quát****

Ứng dụng được thiết kế theo kiến trúc **nhiều lớp**, bám sát mô hình **Model – View – Controller (MVC)** để đảm bảo tính module hóa, dễ bảo trì và mở rộng:

**Lớp Giao diện (View – Java Swing):**  
 Chịu trách nhiệm hiển thị thông tin và tiếp nhận các thao tác từ người dùng thông qua giao diện đồ họa được thiết kế bằng thư viện Java Swing. Các thành phần giao diện sử dụng gồm JButton, JTextField, JTable, JComboBox, JLabel… Các biểu đồ được tích hợp trực tiếp vào cửa sổ giao diện bằng **ChartPanel** của thư viện **JFreeChart**.

**Lớp Điều khiển (Controller):**  
 Là cầu nối giữa giao diện và các lớp xử lý dữ liệu. Các controller chịu trách nhiệm tiếp nhận và xử lý các sự kiện do người dùng thực hiện, đồng thời gọi các phương thức trong DAO hoặc lớp Chart để lấy dữ liệu, xử lý kết quả và cập nhật lại View tương ứng.

**Lớp Dữ liệu (Model):**  
 Đại diện cho các thực thể dữ liệu trong ứng dụng như Agent, Mission, Skill, Team… Các lớp này sử dụng các thuộc tính ở mức private và cung cấp phương thức getter/setter công khai để đảm bảo tính đóng gói và bảo vệ dữ liệu.

**Lớp Truy cập Dữ liệu (DAO – Data Access Object):**  
 Thực hiện các thao tác trực tiếp với cơ sở dữ liệu thông qua **JDBC**. Lớp DAO cung cấp các phương thức CRUD (Create, Read, Update, Delete), cũng như các truy vấn tìm kiếm hoặc thống kê cơ bản. Việc phân tách lớp DAO giúp giảm sự phụ thuộc giữa phần xử lý nghiệp vụ và phần truy cập dữ liệu.

**Lớp Biểu đồ (Chart và ChartDAO):**

**+ ChartDAO:** Đảm nhiệm việc truy vấn dữ liệu thống kê đặc thù từ cơ sở dữ liệu, phục vụ riêng cho các biểu đồ như thống kê số nhiệm vụ theo năm, phân bổ điệp viên theo quốc tịch…

**+ Chart:** Tiếp nhận dữ liệu từ ChartDAO, xử lý và hiển thị biểu đồ bằng **JFreeChart** dưới dạng biểu đồ cột, biểu đồ tròn, biểu đồ đường và nhúng trực tiếp vào giao diện Swing.

**Lớp Cấu hình (Config):**  
 Chứa các thông số kết nối cơ sở dữ liệu như URL, tên người dùng, mật khẩu, và cung cấp phương thức thiết lập kết nối thông qua **JDBC**. Việc tách riêng lớp Config giúp hệ thống dễ dàng quản lý và chỉnh sửa cấu hình kết nối khi cần thiết.

**Lớp Icon:**  
 Lưu trữ và quản lý các tệp hình ảnh biểu tượng sử dụng trong giao diện người dùng, đảm bảo giao diện trực quan, đẹp mắt và nhất quán về mặt hình ảnh.

**Tóm lại**, mô hình tổng quát của ứng dụng được thiết kế theo hướng **module hóa**, đảm bảo từng lớp thực hiện đúng trách nhiệm của mình, giúp giảm thiểu sự phụ thuộc giữa các thành phần, tăng tính tái sử dụng, dễ bảo trì và mở rộng trong tương lai.

#### ****2.1.2. Yêu cầu chức năng****

##### ****1. Quản lý người dùng và xác thực****

**Đăng nhập/Đăng xuất:**

+ Hệ thống cung cấp giao diện đăng nhập để người dùng nhập tên đăng nhập và mật khẩu.

+ Xác thực thông tin người dùng dựa trên dữ liệu lưu trữ trong bảng Account của hệ thống cơ sở dữ liệu MySQL.

+ Mật khẩu được mã hóa bằng thuật toán **SHA-256** trước khi lưu trữ.

+ Hiển thị thông báo lỗi khi đăng nhập thất bại, hỗ trợ cơ chế khóa tài khoản khi phát hiện đăng nhập sai nhiều lần liên tiếp.

+ Lưu trữ thông tin người dùng phiên hiện tại (session) sau khi đăng nhập thành công, đồng thời điều hướng đến giao diện **Dashboard**.

+ Hỗ trợ chức năng đăng xuất.

**Quản lý hồ sơ người dùng (User Profile):**

+ Hiển thị thông tin cơ bản (họ tên, email, vai trò) của người dùng đang đăng nhập.

+ Cho phép người dùng thay đổi mật khẩu của chính mình.

**Quản lý tài khoản người dùng :**

+ Admin được phép xem danh sách tài khoản, thêm mới, chỉnh sửa thông tin (email, vai trò) hoặc xóa người dùng khỏi hệ thống.

##### ****2. Quản lý dữ liệu điệp viên và nhiệm vụ****

**Hiển thị danh sách điệp viên, nhiệm vụ, kỹ năng và tổ chức:**

Hiển thị dữ liệu từ các bảng dưới dạng bảng JTable với các cột tương ứng.

Cho phép sắp xếp và lọc dữ liệu theo các trường thông tin cụ thể như tên điệp viên, quốc tịch, hoặc năm sinh.

**Thêm mới bản ghi:**

Cung cấp form nhập thông tin mới cho các bảng điệp viên, nhiệm vụ, kỹ năng và tổ chức.

Kiểm tra tính hợp lệ của dữ liệu đầu vào trước khi lưu vào CSDL.

**Cập nhật bản ghi:**

Cho phép người dùng chọn một bản ghi để chỉnh sửa và lưu lại dữ liệu vào CSDL.

**Xóa bản ghi:**

Cho phép người dùng xóa bản ghi, hiển thị xác nhận trước khi thực hiện.

##### ****3. Bảng điều khiển và trực quan hóa dữ liệu****

**Hiển thị biểu đồ thống kê sử dụng JFreeChart:**

**Biểu đồ cột (Bar Chart):**

Thống kê top N điệp viên thực hiện nhiều nhiệm vụ nhất.

Thống kê số nhiệm vụ theo từng năm.

**Biểu đồ tròn (Pie Chart):**

Thể hiện tỷ lệ quốc tịch hoặc giới tính của các điệp viên trong tổ chức.

**Biểu đồ đường (Line Chart):**

Biểu diễn xu hướng số nhiệm vụ được thực hiện qua các năm.

##### ****4. Các chức năng bổ sung****

**Quản lý mối quan hệ giữa các bảng (Agent – Mission – Team – Skill):**

Quản lý các bảng liên kết như AgentMission, AgentSkill, TeamRel, cho phép thêm, sửa, xóa quan hệ giữa các đối tượng.

#### ****2.1.3. Yêu cầu phi chức năng****

##### ****1. Tính bảo mật (Security):****

Mật khẩu người dùng được băm SHA-256 và lưu trữ an toàn.

Ngăn chặn truy cập trái phép vào các chức năng quản lý và dữ liệu quan trọng.

##### ****2. Tính dễ sử dụng (Usability):****

Giao diện thiết kế thân thiện, dễ thao tác với người dùng.

Các nút chức năng và thông báo hiển thị rõ ràng, trực quan.

Hệ thống phản hồi nhanh, đảm bảo thao tác mượt mà.

##### ****3. Tính ổn định và tin cậy (Reliability & Robustness):****

Ứng dụng hoạt động ổn định, không bị treo khi thao tác dữ liệu số lượng lớn.

Xử lý ngoại lệ hiệu quả, thông báo lỗi rõ ràng và thân thiện.

Đảm bảo tính toàn vẹn dữ liệu khi thực hiện các thao tác thêm, sửa, xóa.

##### ****4. Tính dễ bảo trì và mở rộng (Maintainability & Extensibility):****

Mã nguồn được tổ chức rõ ràng theo mô hình module hóa: Model – DAO – Controller – Chart – View.

Tuân thủ các nguyên tắc lập trình hướng đối tượng (OOP) và quy ước đặt tên.

Các đoạn mã phức tạp được chú thích đầy đủ.

Dễ dàng bổ sung, nâng cấp các module hoặc thêm chức năng mới mà không ảnh hưởng đến hệ thống hiện có.

##### ****5. Hiệu năng (Performance):****

Các truy vấn CSDL được tối ưu hóa để đảm bảo tốc độ xử lý nhanh.

Việc hiển thị bảng dữ liệu và biểu đồ với số lượng lớn bản ghi không làm ảnh hưởng đến hiệu năng tổng thể của ứng dụng.

#### ****2.2.1. Kiến trúc hệ thống****

Ứng dụng **SpyAgency2024** được thiết kế và xây dựng dựa trên kiến trúc **phân lớp (layered architecture)** theo hướng tiếp cận mô hình **Model – View – Controller (MVC)** kết hợp với các lớp hỗ trợ chuyên biệt. Kiến trúc này giúp phân tách rõ ràng giữa giao diện, xử lý nghiệp vụ và truy cập dữ liệu, đảm bảo tính module hóa, dễ bảo trì và mở rộng khi phát triển phần mềm.

Các thành phần chính trong kiến trúc hệ thống bao gồm:

##### ****Lớp Giao diện người dùng (Presentation Layer – View)****

**Xây dựng bằng Java Swing**, sử dụng các thành phần giao diện như JButton, JTextField, JLabel, JTable, JComboBox, ChartPanel (JFreeChart) để hiển thị thông tin và tiếp nhận thao tác từ người dùng.

Các lớp View chịu trách nhiệm hiển thị dữ liệu và truyền sự kiện người dùng đến Controller xử lý.

Lớp Util chứa các hàm tiện ích dùng chung cho toàn bộ ứng dụng, ví dụ như hàm băm mã hóa mật khẩu, xử lý định dạng dữ liệu, hoặc các hàm hỗ trợ khác. Điển hình là hàm băm mật khẩu sử dụng thuật toán SHA-256 nhằm bảo vệ thông tin đăng nhập người dùng trước khi lưu trữ hoặc so sánh với dữ liệu trong CSDL.Việc mã hóa mật khẩu tại lớp Util giúp tách biệt logic bảo mật khỏi các tầng khác, tăng tính tái sử dụng và bảo trì mã nguồn.

Ví dụ: AgentManagementPanel.java, MissionDashboardPanel.java, AffiliationManagementPanel.java.

##### ****Lớp Điều khiển (Controller Layer)****

Quản lý việc xử lý các sự kiện từ giao diện người dùng, tiếp nhận dữ liệu từ View, gửi yêu cầu đến DAO hoặc ChartDAO, xử lý dữ liệu và cập nhật lại View.

Các controller tổ chức theo từng module chức năng như AgentController.java, MissionController.java, DashboardController.java,..

##### ****Lớp Truy cập Dữ liệu (Data Access Layer – DAO)****

Chịu trách nhiệm truy vấn trực tiếp vào cơ sở dữ liệu MySQL thông qua JDBC.

Thực hiện các thao tác CRUD (Create, Read, Update, Delete) và tìm kiếm dữ liệu cho từng bảng như AgentDAO.java, MissionDAO.java, AffiliationDAO.java,..

Lớp DBConnection trong package Config quản lý việc tạo và đóng kết nối CSDL.

##### ****Lớp Dữ liệu (Data Model Layer – Model)****

Chứa các lớp đối tượng (POJO) biểu diễn các thực thể dữ liệu như Agent.java, Mission.java, Skill.java, Team.java,..

Mỗi lớp chứa các thuộc tính private, và các phương thức getter/setter public, phục vụ cho việc trao đổi dữ liệu giữa các tầng của ứng dụng.

##### ****Lớp Biểu đồ và Thống kê (Chart & ChartDAO)****

**ChartDAO:** Thực hiện các truy vấn đặc thù để lấy dữ liệu thống kê phục vụ cho biểu đồ (ví dụ: top điệp viên nhiều nhiệm vụ nhất, phân bổ quốc tịch…).

**Chart:** Nhận dữ liệu từ ChartDAO, xử lý và dựng biểu đồ trực tiếp bằng thư viện **JFreeChart**, sau đó nhúng vào Swing thông qua ChartPanel.

Các loại biểu đồ gồm: Biểu đồ cột, biểu đồ tròn, biểu đồ đường.

##### ****Lớp Cấu hình (Config)****

Chứa các lớp và file cấu hình hệ thống như DBConnection.java để quản lý thông số và khởi tạo kết nối CSDL.

Áp dụng mẫu thiết kế **Singleton** nhằm đảm bảo chỉ có một kết nối duy nhất tại một thời điểm.

**Lớp Icon**

Lưu trữ và quản lý các tệp hình ảnh icon dùng trong giao diện người dùng, đảm bảo tính trực quan và nhất quán về mặt hình ảnh của ứng dụng.

****Tổng kết:****

Kiến trúc hệ thống **SpyAgency2024** được phân tách rõ ràng theo mô hình **MVC mở rộng**, bổ sung các lớp **Chart**, **ChartDAO**, **Config** và **Icon** phù hợp với yêu cầu nghiệp vụ và tính năng trực quan hóa dữ liệu trong ứng dụng quản lý tình báo. Cách tổ chức này đảm bảo:

+ Dễ bảo trì, sửa lỗi, nâng cấp

+ Dễ mở rộng tính năng mới

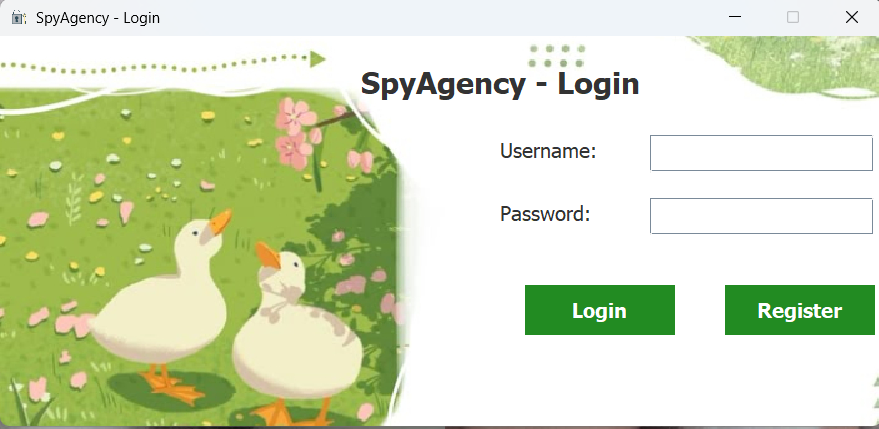
+ Tăng tính ổn định, bảo mật và khả năng tái sử dụng mã nguồn

**2.2.2 Phân tích thiết kế giao diện người dùng**

Giao diện người dùng của ứng dụng **SpyAgency2024** được phát triển dựa trên nền tảng **Java Swing**, kết hợp các thành phần giao diện như JButton, JLabel, JTextField, JTable, JComboBox và ChartPanel (từ thư viện **JFreeChart**) nhằm phục vụ cho việc hiển thị dữ liệu và thao tác trực tiếp với người dùng.

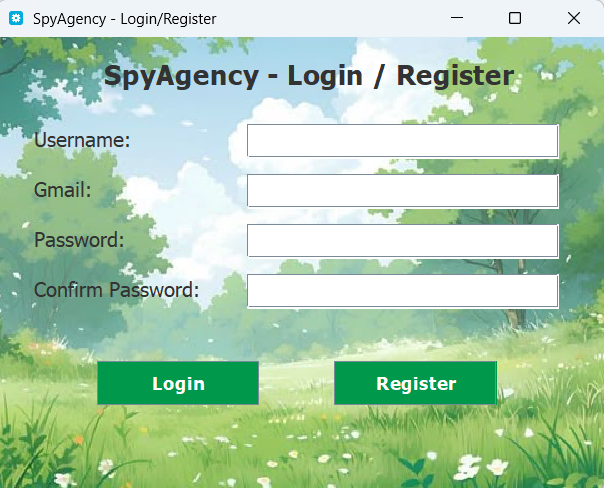
Mục tiêu của thiết kế là tạo nên một giao diện **khoa học, trực quan và dễ thao tác**, đáp ứng tốt yêu cầu quản lý và phân tích dữ liệu tình báo.

**Màn Hình đăng nhập:**



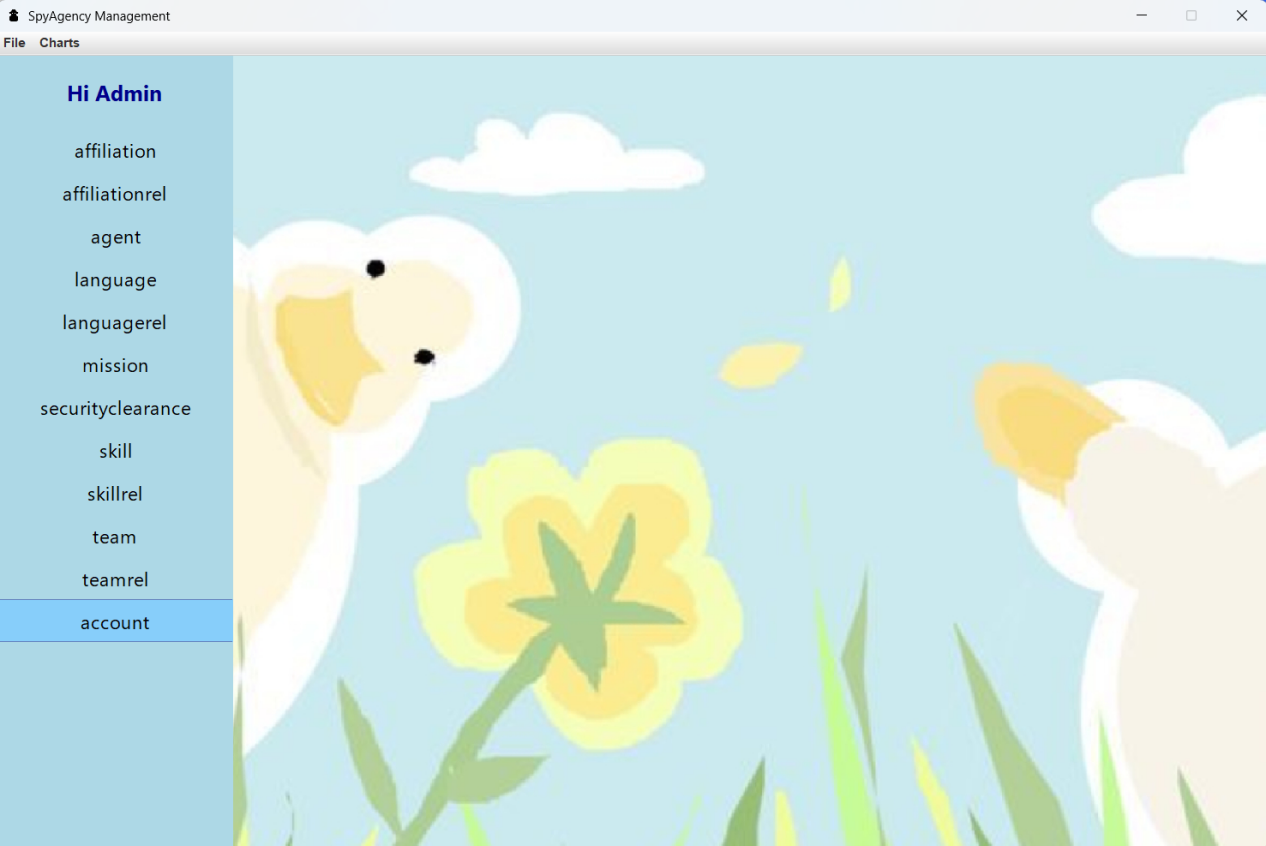
*Hình 1*

**Màn hình đăng ký tài khoản:**



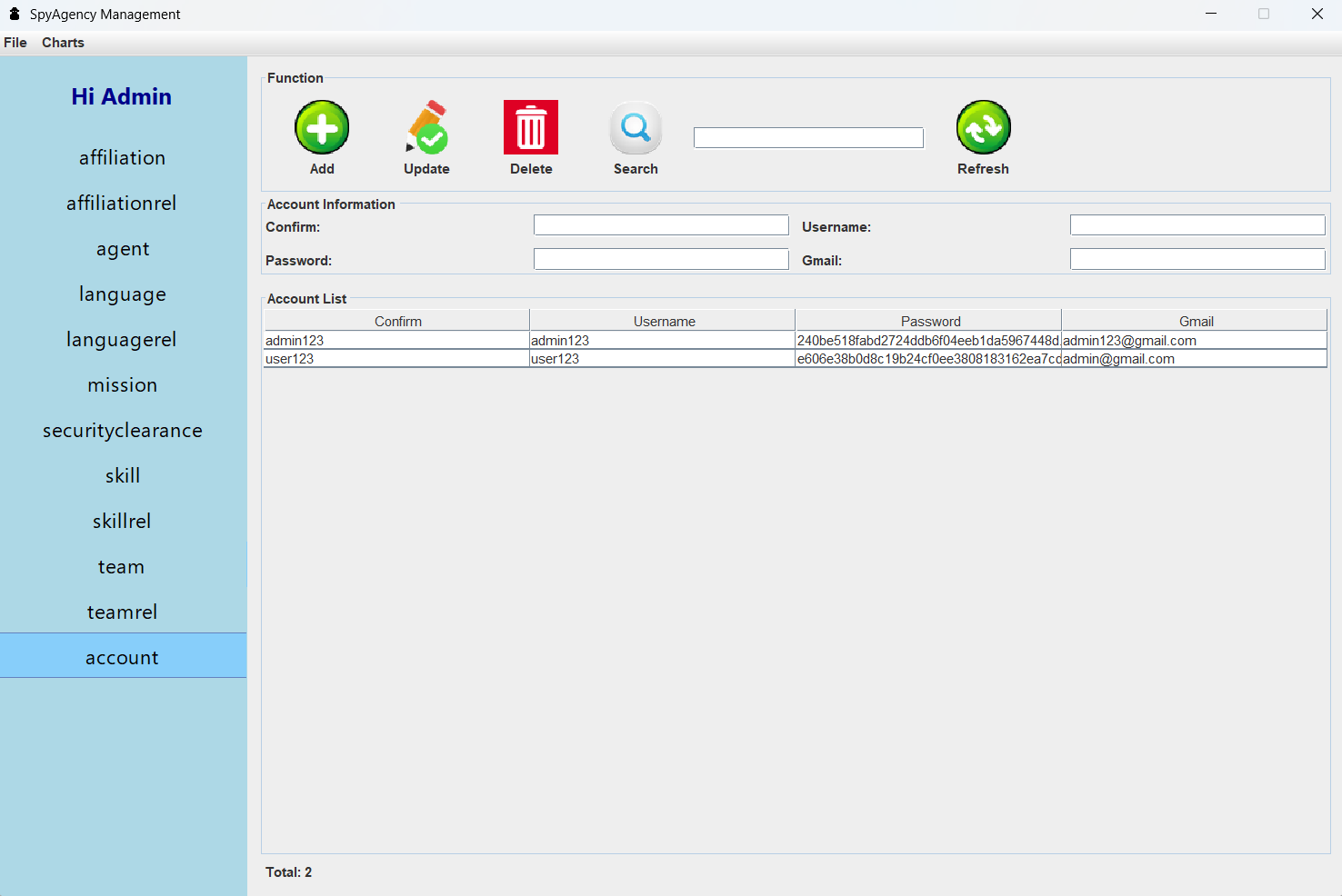
*Hình 2*

**Bảng điều khiển chính:**



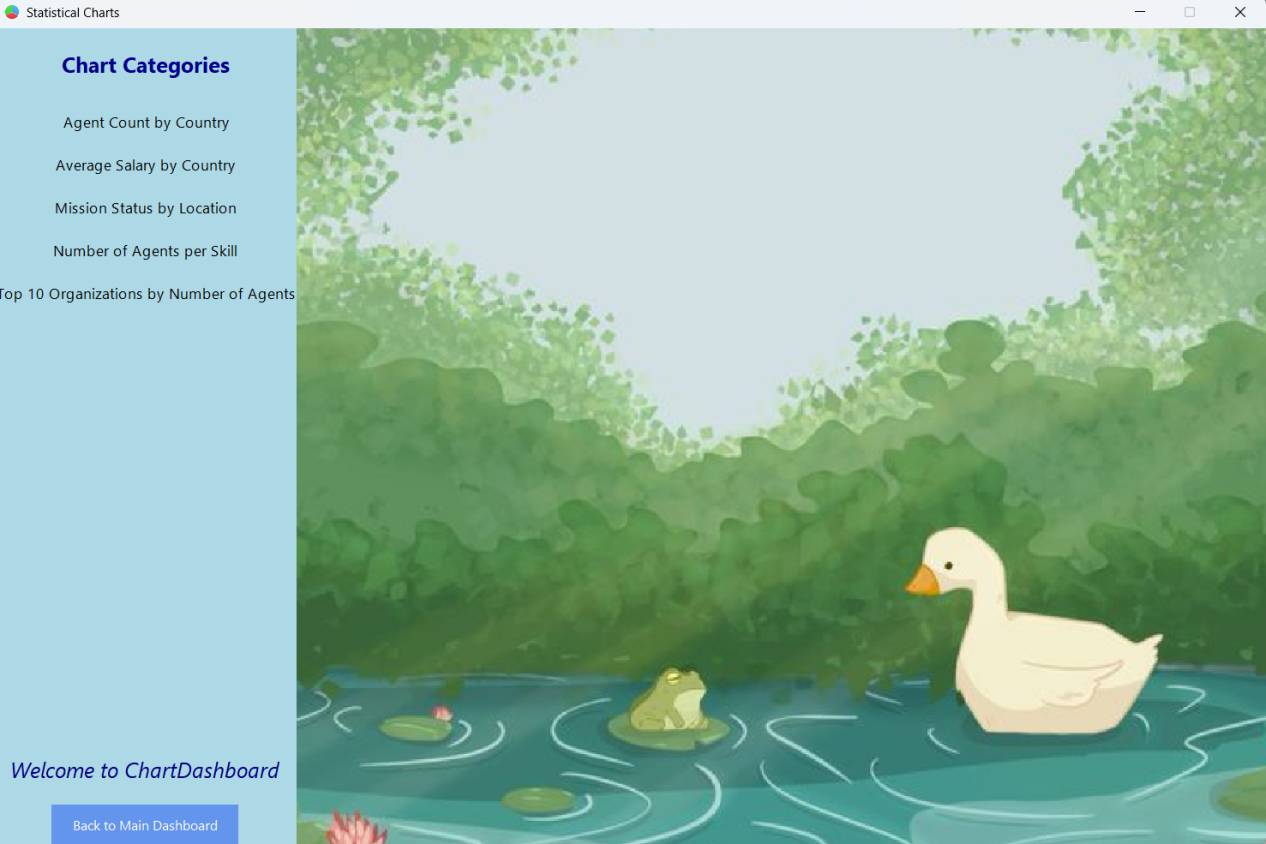
*Hình 3*

**Màn hình giao diện quản lý người dùng :**



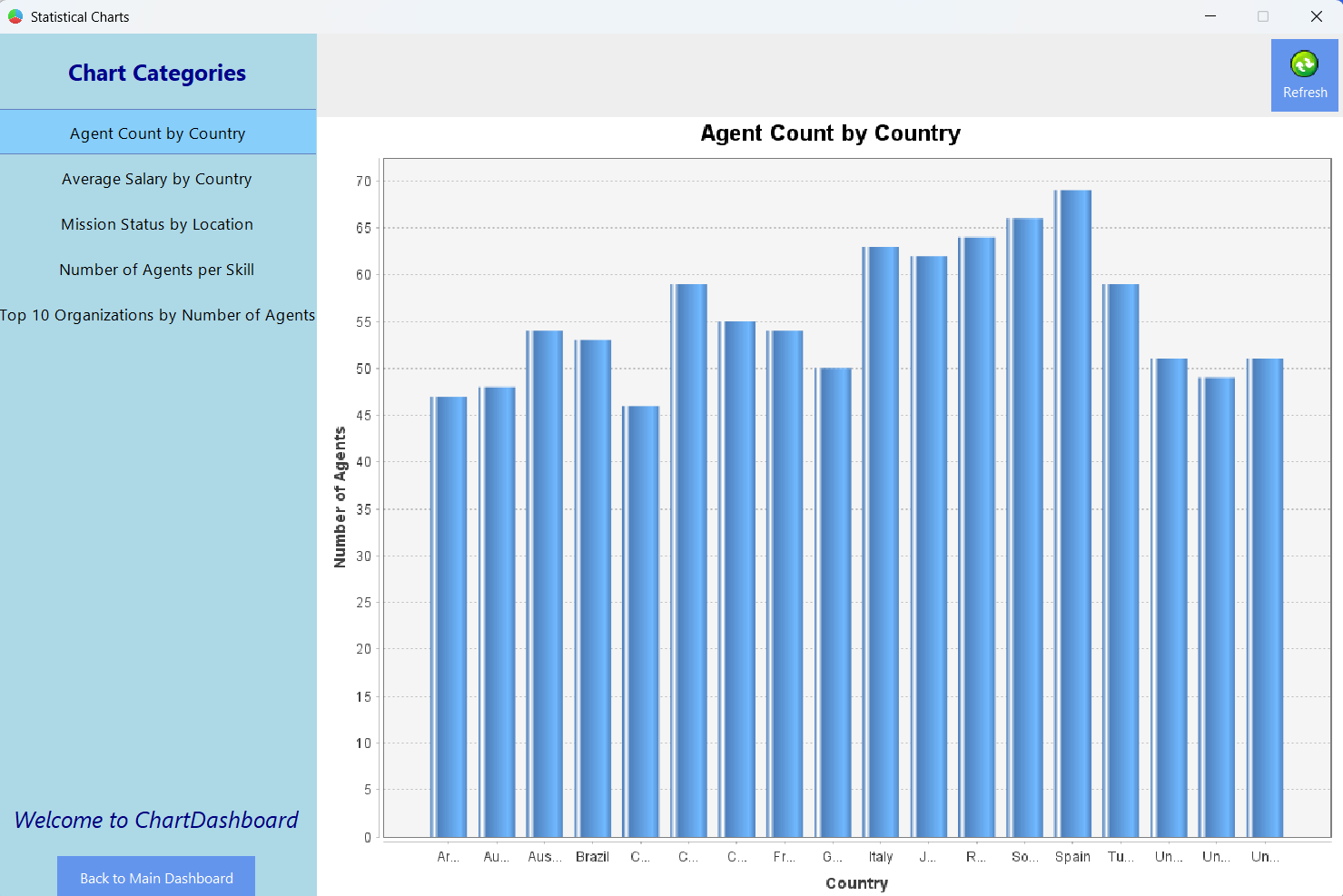
*Hình 4*

**Màn hình giao diện chính của biểu đồ:**



*Hình 5*

**Màn hình xem biểu đồ :**



*Hình 6*

**2.3 Thông tin giao diện.**

Mô tả quy trình người dùng đăng nhập vào hệ thống quản lý Cơ quan Tình báo (Spy Agency) và cách hệ thống hiển thị giao diện chính sau khi đăng nhập thành công

**- Khởi động ứng dụng:**

+ Giao diện đăng nhập được hiển thị đầu tiên, do lớp **LoginInterface2.java** điều khiển.

**- Người dùng tiến hành đăng nhập:**

+ Nhân viên tình báo nhập **tên người dùng** và **mật khẩu** vào các trường nhập liệu (JTextField và JPasswordField).

+ Nhấn vào nút **"Login"** nút này được biểu diễn bằng một biểu tượng phong cách phần mềm quản lý hiện đại.

**- Xử lý đăng nhập:**

+ Sự kiện nhấn nút được xử lý thông qua action listener

+ Hệ thống gọi phương thức checkLogin(username, password) để xác thực thông tin.

Lớp checkLogin sẽ tương tác với AccountDAO để kiểm tra thông tin tài khoản trong cơ sở dữ liệu MySQL.

Mật khẩu được đối chiếu sau khi băm (hash) bằng thuật toán SHA-256 để đảm bảo tính bảo mật

**Các trường hợp xử lý:**

Đăng nhập thành công:

Cửa sổ đăng nhập đóng lại.

Ứng dụng chuyển sang hiển thị DashboardPanel,đây là class điều khiển trực tiếp trang chủ chính

Thông điệp chào mừng hiển thị ở trong Dashboard: Hi Admin

Đăng nhập thất bại:

Hiển thị thông báo lỗi ngay trên giao diện: "Incorrect username or password”.

**- Giao diện Trang chủ (Dashboard):**

Giao diện chính được chia thành:

**Thanh điều hướng bên trái (Sidebar)** với các mục chức năng được thiết kế phù hợp với vai trò người dùng.

Còn bên phải là Hình nền làm cho giao diện thêm màu sắc, tạo cảm giác cho người dùng được thoải mái.

Các nút khác bao gồm:

**Logout: đưa ra màn hình đăng nhập khi muốn sang tài khoản khác .**

Exit: Thoát khỏi ứng dụng nhanh chóng, tiết kiệm thời gian.

Còn các chức năng đặc biệt khác ở cột bên trai của màn hình chính đó là mỗi nút khi ấn vào đều sẽ hiện ra một trang khác nhau, mỗi trang đầy đủ các chức năng phù hợp với ứng dụng quản lý. Các chức năng bao gồm:

- Thêm mới (Add or Create):

+ Người dùng nhập thông tin đầy đủ vào các ô JTextField

+ Nhấn Nút ADD

+ Tự động dữ liệu sẽ được lưu vào trong danh sách và cũng sẽ xuất hiện trong bảng CSDL thông qua sự kết nối trước đó.

- Cập nhập (Update or Edit):

+ Sau khi ấn vào cột thông tin bên dưới ta có thể chỉnh sửa theo ý mình

+ Nhấn nút Update

+ Tự động thông tin mới sẽ được cập nhập và bảng CSDL cũng sẽ được cập nhập theo.

- Xóa (Delete):

+ Sau khi ấn vào cột thông tin bên dưới ta có thể chỉnh sửa theo ý mình

+ Nhấn nút Delete

+ Hộp Thoại xác nhập rằng bạn có muốn xóa sẽ xuất hiện.

+ Ấn xác nhập thì thông tin bên dưới sẽ được xóa và bảng CSDL cũng sẽ được cập nhập dữ liệu bị xóa.

- Làm mới(refresh):

+ Khi ấn vào toàn bộ thông tin hiểu thị sẽ được làm mới, và ô thông tin đang có dữ liệu trước đó sẽ mất và quay lại các ô trắng

- Tìm kiếm (search):

+ Có ô thông tin riêng để người dùng có thể nhập thông tin mình muốn tìm.

+ Khi ấn vào “search”

+ Những thông tin liên quan đến từ khóa đó sẽ xuất hiện

Bên cạch các trang quản lý đó còn có hệ thống quản lý cũng tương tự nhưng đặc biệt hơn ở chỗ, đó là thông tin của hàm băm sẽ được hiện bên dưới cho người dùng nhìn được và cũng như điều chỉnh thông tin thông qua các chức năng CRUD.

**Thống kê và biểu đồ:**

Về giao diện, bên ChartDashboard cũng mang hình nền mang phong cách bắt mắt và dễ gần cho người sử dụng, tạo cảm giác dễ sử dụng. Không những vậy mà cột bên tay trái cũng có các nút ấn, khi ta ấn vào sẽ thấy được các biểu đồ hiện ra, tuy không thể cho người dùng tự tạo biểu đồ của riêng mình nhưng biểu đồ được tạo ra trông rất bắt mắt nhằm để cho người dùng có thể nhìn tồng quan nhất có thể, mọi dữ liệu đều kết nối từ MySQL khi ta thêm bớt, xóa sửa thì đều sẽ ảnh hưởng đến biểu đồ vì bên cạnh biểu đồ có một nút ấn cho phép làm mới biểu đồ, luôn cập nhập cho biểu đồ thông tin mới nhất, làm cho người dùng nắm bắt thông tin nhanh nhất, tạo điều kiện phát triển, thúc đẩy quá trình cho các lĩnh vực khác

**2.4 Cấu trúc cơ sở dữ liệu MySQL**

#### 📑 ****2.4.1 Mô tả bảng dữ liệu****

**- Bảng** account

**Mục đích:** Quản lý thông tin tài khoản người dùng

**Các cột:**

id (INT, PK, AI): Mã định danh tài khoản

username (VARCHAR(50), NOT NULL, UNIQUE): Tên đăng nhập

password (VARCHAR(255), NOT NULL): Mật khẩu (được mã hóa)

Confirm (VARCHAR(20), NOT NULL): Nhập lại mật khẩu

****- Bảng** agent**

****Mục đích:**** Quản lý thông tin điệp viên.

****Các cột:****

agent\_id (INT, PK, AI): Mã định danh điệp viên

First\_name (VARCHAR(255), NOT NULL): Tên điệp viên

Last\_name (VARCHAR(255), NOT NULL): Tên điệp viên

address (VARCHAR(255)): Địa chỉ

city (VARCHAR(100)): Thành phố

country (VARCHAR(100)): Quốc gia

salary (DECIMAL(15,2)): Mức lương

****- Bảng** affiliation**

**Mục đích:** Quản lý thông tin tổ chức điệp viên.

**Các cột:**

Affiliation\_id (INT, PK, AI): Mã định danh tổ chức

Affiliation\_name (VARCHAR(255), NOT NULL): Tên tổ chức

description (TEXT): Mô tả tổ chức

**- **Bảng** mission**

**Mục đích:** Quản lý thông tin nhiệm vụ.

**Các cột:**

mission\_id (INT, PK, AI): Mã nhiệm vụ

mission\_name (VARCHAR(255), NOT NULL): Tên nhiệm vụ

location (VARCHAR(255)): Địa điểm

agent\_id (INT, FK): Mã điệp viên tham gia

access\_id (INT, FK): Mã cấp độ bảo mật (tham chiếu bảng securityclearance)

team\_id (INT, FK): Mã đội thực hiện

mission\_status (VARCHAR(50)): Trạng thái nhiệm vụ

****- Bảng** skill**

**Mục đích:** Quản lý danh mục kỹ năng điệp viên.

**Các cột:**

id (INT, PK, AI): Mã kỹ năng

name (VARCHAR(255), NOT NULL): Tên kỹ năng

****Bảng** language**

**Mục đích:** Quản lý danh mục ngôn ngữ.

**Các cột:**

language\_id (INT, PK, AI): Mã ngôn ngữ

language\_name (VARCHAR(255), NOT NULL): Tên ngôn ngữ

****- Bảng** securityclearance**

**Mục đích:** Quản lý mức độ bảo mật của điệp viên.

**Các cột:**

sc\_id (INT, PK, AI): Mã mức bảo mật

sc\_level (VARCHAR(50), NOT NULL): Tên mức bảo mật

description (TEXT): Mô tả độ khó

****- Bảng** team**

**Mục đích:** Quản lý các nhóm điệp viên.

**Các cột:**

id (INT, PK, AI): Mã nhóm

name (VARCHAR(255), NOT NULL): Tên nhóm

meeting\_frequency (VARCHAR(50)): Tần suất họp đội

****- Bảng** affiliationrel**

**Mục đích:** Lưu thông tin mối quan hệ giữa điệp viên và tổ chức.

**Các cột:**

affiliation\_strength (VARCHAR(50)): Mức độ liên kết

agent\_id (INT, PK, FK): Mã điệp viên

affiliation\_id (INT, FK): Mã tổ chức

****- Bảng** skillrel**

**Mục đích:** Lưu thông tin mối quan hệ giữa điệp viên và kỹ năng.

**Các cột:**

agent\_id (INT, FK): Mã điệp viên

skill\_id (INT, FK): Mã kỹ năng

****- Bảng** languagerel**

**Mục đích:** Lưu thông tin mối quan hệ giữa điệp viên và ngôn ngữ.

**Các cột:**

agent\_id (INT, FK): Mã điệp viên

language\_id (INT, FK): Mã ngôn ngữ

****- Bảng** teamrel**

**Mục đích:** Lưu thông tin mối quan hệ giữa điệp viên và nhóm.

**Các cột:**

agent\_id (INT, FK): Mã điệp viên

team\_id (INT, FK): Mã nhóm

#### 📑 ****2.4.2. Ràng buộc và Quan hệ****

**Ràng buộc:**

**Primary Key:**  
Mỗi bảng đều có cột id làm khoá chính (PK).

**Foreign Key:**

agent.security\_clearance\_id → securityclearance.id

affiliationrel.agent\_id → agent.id

affiliationrel.affiliation\_id → affiliation.id

skillrel.agent\_id → agent.id

skillrel.skill\_id → skill.id

languagerel.agent\_id → agent.id

languagerel.language\_id → language.id

teamrel.agent\_id → agent.id

teamrel.team\_id → team.id

**Unique:**

account.username

**Not Null:**

account.username, account.password, account.role, agent.name, affiliation.name, mission.name, skill.name, language.name, securityclearance.level, team.name

**Quan hệ giữa các bảng:**

account độc lập

agent — securityclearance: Quan hệ 1-n

agent — affiliationrel: Quan hệ 1-n

agent — skillrel: Quan hệ 1-n

agent — languagerel: Quan hệ 1-n

agent — teamrel: Quan hệ 1-n

affiliationrel — affiliation: Quan hệ n-1

skillrel — skill: Quan hệ n-1

languagerel — language: Quan hệ n-1

teamrel — team: Quan hệ n-1

## 

## CHƯƠNG 3: CƠ CHẾ HOẠT ĐỘNG CỦA HỆ THỐNG

### ****3.1 Cấu trúc tổng thể****

Hệ thống phần mềm **Spy Agency Management** được thiết kế và xây dựng trên nền tảng ngôn ngữ lập trình **Java**, kết hợp cơ sở dữ liệu **MySQL** để lưu trữ và quản lý dữ liệu tập trung. Dự án được tổ chức theo mô hình **phân lớp (Layered Architecture)**, kết hợp chặt chẽ với **mô hình thiết kế phần mềm MVC (Model-View-Controller)** và **DAO Pattern (Data Access Object)** nhằm tách biệt các chức năng nghiệp vụ, giao diện và xử lý dữ liệu.

Tổng thể hệ thống được chia thành các **package** (gói) riêng biệt, mỗi gói đảm nhiệm một vai trò, một chức năng rõ ràng như quản lý model, view, controller, DAO, config và chart. Việc tổ chức như vậy giúp mã nguồn được quản lý khoa học, dễ mở rộng, bảo trì và nâng cấp về sau.

Cấu trúc hệ thống bao gồm:

**Model:** Các lớp biểu diễn đối tượng dữ liệu.

**DAO:** Các lớp thao tác dữ liệu với MySQL.

**Controller:** Xử lý sự kiện và điều phối giữa View và DAO.

**View:** Các lớp giao diện người dùng với Java Swing.

**Chart & ChartDAO:** Xử lý và trực quan hóa dữ liệu biểu đồ.

**Config:** Cấu hình hệ thống và kết nối CSDL.

**Icon:** Quản lý hình ảnh biểu tượng và tài nguyên giao diện.

Thiết kế này đảm bảo hệ thống dễ dàng phát triển thêm tính năng mới như thống kê, phân quyền nâng cao hay cảnh báo mà không ảnh hưởng đến cấu trúc cũ.

### ****3.2 Lớp Model****

**Lớp Model** chịu trách nhiệm biểu diễn các thực thể (entity) thực tế trong hệ thống phần mềm, chẳng hạn như **Agent**, **Mission**, **Affiliation**, **Account**, **Skill**, **Team**,... Mỗi lớp Model chính là một **Plain Old Java Object (POJO)** với các thuộc tính private và phương thức getter, setter public.

Việc xây dựng Model theo chuẩn POJO giúp:

Đảm bảo tính đóng gói (Encapsulation).

Dễ dàng ánh xạ dữ liệu với các bảng tương ứng trong cơ sở dữ liệu.

Tạo nền tảng để chuyển đổi dữ liệu linh hoạt giữa các lớp DAO, Controller và View.

Cách tổ chức này giúp dữ liệu giữa chương trình và cơ sở dữ liệu luôn đồng nhất, thuận tiện trong việc thêm mới, cập nhật và xóa dữ liệu.

**3.3 Lớp DAO (Data Access Object)**

**Lớp **DAO**** đóng vai trò trung gian giữa hệ thống và cơ sở dữ liệu, chuyên xử lý các thao tác với dữ liệu như **thêm, xóa, sửa, tìm kiếm**. Các lớp DAO như **AgentDAO**, **AffiliationDAO**, **MissionDAO** sử dụng **JDBC** để thực thi các câu lệnh SQL trực tiếp và trả kết quả về cho Controller xử lý tiếp.

Ưu điểm:

Tách biệt truy vấn SQL khỏi Controller, tăng tính bảo trì.

Dễ dàng thay đổi cơ sở dữ liệu hoặc cấu trúc bảng mà không ảnh hưởng đến các tầng khác.

Tái sử dụng phương thức, tránh trùng lặp mã.

DAO cũng giúp xử lý ngoại lệ và quản lý kết nối hiệu quả, tránh tình trạng rò rỉ tài nguyên.

### ****3.4 Lớp Controller****

**Controller** đóng vai trò điều phối giữa **View** và **DAO**, tiếp nhận các sự kiện người dùng từ giao diện và gọi các phương thức DAO để thực thi nghiệp vụ, sau đó trả lại dữ liệu cho View hiển thị.

Ví dụ:

Khi người dùng nhấn nút **“Thêm trong Agent”**, Controller sẽ thu thập dữ liệu từ form nhập liệu, tạo đối tượng Agent mới, gọi AgentDAO.insertAgent(agent) và sau đó cập nhật lại bảng dữ liệu hiển thị.

Ưu điểm:

Tách rời logic xử lý nghiệp vụ khỏi giao diện.

Dễ dàng kiểm soát luồng xử lý và mở rộng chức năng khi cần.

Ngoài ra, Controller cũng thực hiện kiểm tra dữ liệu hợp lệ trước khi gửi về DAO nhằm đảm bảo tính toàn vẹn dữ liệu của hệ thống.

### ****3.5 Lớp View (Java SWing)****

**View** trong hệ thống được thiết kế bằng **Java Swing**, gồm các **JPanel** riêng biệt như **AgentManagementJPanel**, **MissionManagementJPanel**, **AffiliationManagementJPanel**… Mỗi View có giao diện hiện đại, bố cục rõ ràng, gồm các thành phần:

**Label** hiển thị tiêu đề và mô tả.

**TextField** nhập liệu.

**ComboBox** chọn lựa nhanh.

**Button** thực hiện tác vụ.

**JTable** hiển thị dữ liệu dạng bảng.

**Icon** và hiệu ứng hỗ trợ trực quan.

Các thành phần giao diện được bố trí hợp lý, kết hợp icon đẹp mắt và màu sắc tối ưu giúp người dùng thao tác nhanh, dễ sử dụng và tránh nhầm lẫn.

### ****3.6 Lớp Chart và ChartDAO****

Hệ thống sử dụng **thư viện JFreeChart** để trực quan hóa dữ liệu thông qua các biểu đồ **hình tròn, cột, đường**. Lớp **Chart** chịu trách nhiệm cấu hình và hiển thị biểu đồ, còn **ChartDAO** lấy dữ liệu từ MySQL theo truy vấn phù hợp.

Ví dụ:

Đếm số lượng Agent theo quốc tịch bằng phương thức AgentChartDAO.countByNationality() và trả dataset về cho biểu đồ tròn.

Thống kê số nhiệm vụ theo trạng thái (thành công, thất bại) để biểu diễn bằng biểu đồ cột.

Ưu điểm:

Giúp nhà quản lý theo dõi số liệu trực quan.

Phân tích xu hướng, đánh giá hiệu quả tổ chức dễ dàng.

### ****3.7 Cấu hình kết nối (Config)****

Lớp **Config** quản lý các thông số cấu hình kết nối cơ sở dữ liệu như:

Địa chỉ máy chủ.

Tên cơ sở dữ liệu.

Tên đăng nhập.

Mật khẩu.

Cổng kết nối.

Lớp **DBConnection** được thiết kế theo **mẫu Singleton** đảm bảo chỉ tạo một kết nối duy nhất trong toàn hệ thống, giúp tiết kiệm tài nguyên và tránh lỗi khi nhiều thao tác truy xuất đồng thời.

Ưu điểm:

Dễ dàng điều chỉnh thông số khi triển khai thực tế.

Bảo mật thông tin kết nối.

Tránh lặp lại đoạn mã thiết lập kết nối trong mỗi DAO.

### ****3.8 Xử lý xác thực người dùng****

Hệ thống có cơ chế xác thực tài khoản người dùng tại màn hình đăng nhập. Quá trình xác thực gồm:

Nhập tên tài khoản và mật khẩu.

Mã hóa mật khẩu bằng thuật toán **SHA-256**.

So sánh mật khẩu mã hóa với dữ liệu trong bảng **Account**.

## CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

### 4.1 Môi trường và Quy trình Thử nghiệm

Để đảm bảo tính chính xác và độ tin cậy của hệ thống, quá trình kiểm thử được tiến hành trong môi trường phần cứng và phần mềm cụ thể, phù hợp với điều kiện vận hành thực tế của các ứng dụng quản lý thông tin hiện nay. Việc lựa chọn môi trường thử nghiệm hiện đại giúp đảm bảo hệ thống có thể đáp ứng tốt về hiệu suất và tính ổn định trong quá trình vận hành.

**Cấu hình môi trường kiểm thử như sau:**

**Hệ điều hành:** Windows 11 Pro 64-bit

**Phiên bản JDK:** Java Development Kit 17 (LTS)

**Hệ quản trị cơ sở dữ liệu:** MySQL Community Server 8.0

**Công cụ phát triển:** IntelliJ IDEA phiên bản 2024.1 Ultimate

**Thư viện biểu đồ:** JFreeChart phiên bản 1.5.4

**Dữ liệu thử nghiệm** được xây dựng với hơn **1000 bản ghi**, phân bổ đều vào các bảng dữ liệu quan trọng của hệ thống như **Agent**, **Mission**, **Affiliation**, **Affiliationrel**, **Skill**, **Skillrel**, **Team**, **Teamrel**, **Securityclearance** và **Account**. Các bản ghi này được tạo giả lập dựa trên kịch bản thực tế, đảm bảo độ đa dạng và tính phức tạp nhất định trong dữ liệu nhằm kiểm tra năng lực xử lý của hệ thống.

**Quy trình thử nghiệm** được thực hiện theo các bước chuẩn hóa gồm:

**Kiểm thử chức năng đăng nhập:** Thực hiện với nhiều tài khoản khác nhau để kiểm tra khả năng phân quyền truy cập, xác thực thông tin và điều hướng giao diện phù hợp.

**Kiểm thử chức năng quản lý dữ liệu:** Tiến hành các thao tác thêm mới, chỉnh sửa, xóa, tìm kiếm trên từng bảng dữ liệu, đảm bảo tính toàn vẹn và đồng bộ dữ liệu giữa giao diện và cơ sở dữ liệu.

**Kiểm thử biểu đồ thống kê:** Tạo biểu đồ với các dạng thống kê khác nhau như biểu đồ cột, biểu đồ tròn, biểu đồ đường… Kiểm tra khả năng hiển thị chính xác dữ liệu và tính năng cập nhật động khi dữ liệu thay đổi.

**Kiểm thử hiệu năng:** Thực hiện các thao tác liên tục với lượng dữ liệu lớn để đánh giá khả năng phản hồi của ứng dụng, xử lý đồng thời và kiểm tra phát sinh lỗi hoặc hiện tượng treo hệ thống.

Quá trình kiểm thử được thực hiện trong nhiều phiên, với các kịch bản lỗi giả lập như đăng nhập sai, thêm dữ liệu trùng khóa chính, xóa bản ghi đang được tham chiếu… nhằm đánh giá tính ổn định và khả năng xử lý ngoại lệ của hệ thống.

### 4.2 Kết quả Kiểm thử

Sau khi tiến hành kiểm thử toàn bộ hệ thống theo kế hoạch đã thiết lập, kết quả đạt được như sau:

**Chức năng đăng nhập:**  
Hệ thống thực hiện kiểm tra thông tin đăng nhập chính xác, đảm bảo các trường hợp tài khoản không tồn tại, sai mật khẩu hoặc quyền hạn không phù hợp đều được xử lý đúng. Khi đăng nhập thành công, hệ thống hiển thị giao diện tương ứng với vai trò của người dùng, giúp phân tách rõ ràng giữa các chức năng quản trị và chức năng nhân viên.

**Chức năng quản lý dữ liệu:**  
Các chức năng thêm mới, chỉnh sửa, xóa và tìm kiếm bản ghi trên từng bảng hoạt động đúng như yêu cầu. Giao diện cập nhật tức thì, đảm bảo dữ liệu trong cơ sở dữ liệu được đồng bộ với thông tin hiển thị. Hệ thống cũng hỗ trợ tìm kiếm nhanh theo từ khóa và theo từng tiêu chí cụ thể, giúp người dùng thao tác thuận tiện và tiết kiệm thời gian.

**Chức năng biểu đồ thống kê:**  
Hệ thống cho phép tạo biểu đồ trực quan với dữ liệu chính xác, hỗ trợ nhiều dạng biểu đồ khác nhau. Biểu đồ có khả năng cập nhật động khi dữ liệu thay đổi mà không cần khởi động lại chương trình. Tốc độ render biểu đồ nhanh, màu sắc rõ ràng, các nhãn và thông tin hiển thị đầy đủ, hỗ trợ tốt cho công tác theo dõi và phân tích dữ liệu.

**Hiệu năng xử lý:**  
Khi thử nghiệm với hơn 1000 bản ghi và thực hiện các thao tác liên tục trong thời gian dài, hệ thống vận hành ổn định, không phát sinh lỗi runtime hay hiện tượng treo ứng dụng. Tốc độ xử lý các thao tác thêm, sửa, xóa và tìm kiếm đạt mức tốt, thời gian phản hồi trung bình dưới 1 giây, đảm bảo đáp ứng tốt nhu cầu sử dụng thực tế.

Ngoài ra, hệ thống cũng được thử nghiệm với các kịch bản lỗi giả lập như nhập dữ liệu trùng khóa chính, xóa bản ghi đang được tham chiếu bởi bảng khác, đăng nhập nhiều lần liên tiếp với tài khoản sai… Tất cả các trường hợp ngoại lệ đều được xử lý đúng theo logic và thông báo rõ ràng đến người dùng.

### 4.3 Đánh giá Sản phẩm

Sau quá trình kiểm thử toàn diện, hệ thống cho thấy khả năng đáp ứng tốt các yêu cầu chức năng đặt ra ban đầu. Có thể đánh giá sản phẩm dựa trên một số tiêu chí sau:

**Ưu điểm:**

**Áp dụng mô hình MVC chuẩn hóa:**  
Hệ thống được xây dựng dựa trên kiến trúc MVC, giúp phân tách rõ ràng các thành phần **Model**, **View**, **Controller**. Việc này không chỉ giúp tăng tính bảo trì và khả năng mở rộng hệ thống mà còn đảm bảo quá trình phát triển sau này trở nên thuận tiện và dễ kiểm soát.

**Giao diện hiện đại, thân thiện:**  
Giao diện người dùng được thiết kế hiện đại, màu sắc hài hòa, các chức năng bố trí hợp lý. Người dùng dễ dàng tiếp cận và thực hiện các thao tác quản lý, giảm thiểu tối đa thời gian làm quen hệ thống.

**Hệ thống biểu đồ trực quan, hỗ trợ phân tích:**  
Việc tích hợp biểu đồ giúp hệ thống không chỉ đơn thuần là một công cụ quản lý dữ liệu mà còn hỗ trợ phân tích và trực quan hóa số liệu, phục vụ tốt cho công tác báo cáo và ra quyết định.

**Hiệu năng ổn định với dữ liệu lớn:**  
Hệ thống xử lý mượt mà với hơn 1000 bản ghi, đảm bảo tốc độ phản hồi tốt, không phát sinh lỗi, kể cả khi thực hiện nhiều thao tác đồng thời.

**Hạn chế:**

**Chưa hỗ trợ đa ngôn ngữ:**  
Hiện tại giao diện hệ thống chỉ hỗ trợ Tiếng Anh. Điều này chưa đáp ứng được nhu cầu sử dụng của các tổ chức quốc tế hoặc các đơn vị đa quốc gia. Trong tương lai, hệ thống cần được bổ sung tính năng lựa chọn ngôn ngữ hoặc phiên bản song ngữ để mở rộng đối tượng người dùng.

**Chưa tích hợp các chức năng phân tích nâng cao:**  
Hệ thống mới chỉ dừng lại ở mức thống kê và hiển thị số liệu dạng biểu đồ. Các chức năng phân tích nâng cao như **dự đoán xu hướng hoạt động của đặc vụ**, **phân tích mối liên hệ giữa các đặc vụ và nhiệm vụ**, hay **đánh giá hiệu quả của các đội nhóm** vẫn chưa được triển khai. Đây là các tính năng tiềm năng cần được nghiên cứu và phát triển trong các phiên bản kế tiếp để gia tăng giá trị sử dụng cho sản phẩm.

**Tổng kết:**  
Qua quá trình thử nghiệm và đánh giá, có thể khẳng định hệ thống đạt yêu cầu về tính năng, hiệu năng và giao diện người dùng. Đây là nền tảng vững chắc để tiếp tục phát triển các tính năng mới, mở rộng khả năng phân tích và tích hợp với các hệ thống khác trong tương lai.

# ****KẾT LUẬN****

Sau một quá trình nghiên cứu, thiết kế và phát triển nghiêm túc, đồ án **“SpyAgency2024 – Hệ thống Quản lý và Phân tích Dữ liệu Điệp Viên”** đã hoàn thành tốt các mục tiêu đề ra, xây dựng thành công một phần mềm desktop hỗ trợ quản lý thông tin điệp viên, nhiệm vụ và các thông tin liên quan với tính năng trực quan hóa dữ liệu mạnh mẽ, giao diện thân thiện và đảm bảo tính bảo mật. Những kết quả nổi bật mà đồ án đã đạt được gồm:

## ****1. Xây dựng thành công hệ thống quản lý người dùng và xác thực (User Management & Authentication)****

Ứng dụng đã tích hợp đầy đủ chức năng đăng nhập và đăng xuất, quản lý phiên làm việc người dùng một cách an toàn. Mật khẩu người dùng được mã hóa bằng thuật toán **SHA-256** trước khi lưu vào cơ sở dữ liệu MySQL, đảm bảo tính bảo mật. Ngoài ra, phần mềm cung cấp chức năng cập nhật thông tin cá nhân cho từng người dùng và xây dựng giao diện riêng cho **Quản trị viên (Admin)**, cho phép quản lý toàn bộ tài khoản người dùng, phân quyền thao tác, chỉnh sửa và vô hiệu hóa tài khoản khi cần thiết, giúp đảm bảo quyền truy cập dữ liệu một cách hợp lý và an toàn.

## ****2. Phát triển module quản lý dữ liệu điệp viên và các thực thể liên quan (Data Management)****

Hệ thống đã thiết kế giao diện thân thiện, khoa học, cho phép người dùng thực hiện đầy đủ các thao tác CRUD (Create, Read, Update, Delete) trên các bảng dữ liệu chính như **Agent**, **Mission**, **Affiliation**, **Skill**, **Language**, **Team**, và các bảng quan hệ như **AffiliationRel**, **LanguageRel**, **SkillRel**, **TeamRel**. Cơ sở dữ liệu MySQL được xây dựng với cấu trúc chuẩn hóa, đảm bảo tính toàn vẹn dữ liệu và tính mở rộng cao.

Việc tích hợp công cụ **MySQL** hỗ trợ kiểm tra dữ liệu nhanh chóng và thuận tiện. Dữ liệu mẫu được xây dựng bám sát tình huống thực tế của một tổ chức điệp viên, với đầy đủ thông tin về điệp viên, nhiệm vụ, tổ chức, kỹ năng và phân quyền nhằm giúp việc kiểm tra, quản lý dữ liệu trở nên trực quan và dễ dàng.

## ****3. Triển khai chức năng trực quan hóa dữ liệu bằng biểu đồ (Data Visualization)****

Đồ án tích hợp thành công thư viện **JFreeChart** để trực quan hóa các số liệu liên quan đến điệp viên và nhiệm vụ dưới dạng biểu đồ sinh động. Các dạng biểu đồ đã thực hiện gồm:

**Bar Chart**: Thống kê số lượng điệp viên theo quốc tịch hoặc theo từng team.

**Pie Chart**: Tỷ lệ phân bổ điệp viên theo từng cấp độ bảo mật (Security Clearance).

**Line Chart**: Thể hiện xu hướng hoàn thành nhiệm vụ theo từng tháng hoặc năm.

Người dùng có thể lựa chọn tham số như quốc gia, team, hoặc mốc thời gian để hiển thị biểu đồ theo nhu cầu phân tích cụ thể, giúp việc giám sát và quản lý hoạt động điệp viên trở nên hiệu quả hơn.

## ****4. Ứng dụng hiệu quả các kỹ thuật Lập trình Hướng đối tượng (OOP) và công nghệ Java****

Trong quá trình thiết kế và lập trình, đồ án đã vận dụng triệt để các nguyên tắc **OOP** như đóng gói, kế thừa và đa hình vào hệ thống các lớp đối tượng **POJO** (Agent, Mission, Affiliation, Team, Skill, Language), các lớp **DAO** (AgentDAO, MissionDAO, AffiliationDAO, …) để giao tiếp với cơ sở dữ liệu và các lớp **Service** chuyên xử lý logic nghiệp vụ như AuthService, ChartService, MissionService.

Ngoài ra, dự án còn áp dụng các công nghệ và công cụ hiện đại:

**Java Swing**: Xây dựng giao diện người dùng dạng desktop trực quan và dễ thao tác.

## ****5. Hoàn thiện cấu trúc dự án và hệ thống tài liệu****

Toàn bộ mã nguồn, tài nguyên (icon, ảnh, CSS), file cấu hình FXML và các script SQL tạo bảng, chèn dữ liệu mẫu đều được tổ chức khoa học, rõ ràng theo chuẩn dự án thực tế. Ngoài ra, đồ án cũng hoàn thiện bộ **tài liệu kỹ thuật**, **hướng dẫn sử dụng**, và **báo cáo phân tích thiết kế** giúp việc triển khai, vận hành và mở rộng phần mềm sau này thuận lợi hơn.

**Hướng phát triển**

Mặc dù đã đạt được nhiều kết quả khả quan, phần mềm **SpyAgency2024** vẫn còn một số hướng phát triển tiềm năng trong tương lai:

Bổ sung chức năng **thông báo và cảnh báo sự kiện bảo mật**, theo dõi hoạt động bất thường của các tài khoản điệp viên.

Phát triển module **thống kê và phân tích AI**, dự đoán mức độ rủi ro khi phân công nhiệm vụ.

Thiết kế **dashboard tổng quan realtime** trực quan hóa toàn bộ số liệu điệp viên, nhiệm vụ, và thông tin tổ chức.

Tích hợp **hệ thống đa ngôn ngữ** để phục vụ người dùng quốc tế.

Với những kết quả đạt được, có thể khẳng định đồ án **“SpyAgency2024”** không chỉ giúp người thực hiện củng cố, vận dụng hiệu quả các kiến thức về lập trình Java, cơ sở dữ liệu và trực quan hóa số liệu, mà còn xây dựng được một phần mềm thực tế có tính ứng dụng cao. Đây là nền tảng thuận lợi để tiếp tục nghiên cứu, mở rộng các tính năng nâng cao trong lĩnh vực quản lý và phân tích dữ liệu tình báo, phục vụ cho công tác giám sát, quản trị và phân tích thông tin trong các tổ chức an ninh.

# TÀI LIỆU THAM KHẢO

1. Slide bài giảng Lập Trình Hướng Đối Tượng của Hệ thống thầy Lê Ngọc Hiếu
2. JFreeChart Project. JFreeChart Developer Guide. Truy cập tại: <http://www.jfree.org/jfreechart/.>
3. git clone <https://github.com/hgbaodev/QuanLyKhoDienThoai.git>.