**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI JAVA**

**CHỦ ĐỀ 04: QUẢN LÝ ĐIỂM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sinh viên thực hiện** | **Lớp** | **Khóa** | **Mã sinh viên** |
| **Nguyễn Văn Lâm** | **DCCNTT12.10.4** | **K12** | **20211166** |
| **Vi Văn Tuấn** | **DCCNTT12.10.4** | **K12** | **20211138** |
| **Lê Quý Mùi** | **DCCNTT12.10.4** | **K12** | **20211133** |

**Bắc Ninh, năm 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG VỚI JAVA**

**CHỦ ĐỀ 04: QUẢN LÝ ĐIỂM**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sinh viên thực hiện** | **Khóa** | **Lớp** | **Mã sinh viên** | **Điểm bằng số** | **Điểm bằng chữ** | **Ký tên** |
| **Nguyễn Văn Lâm** | **K12** | **DCCNTT12.10.4** | **20211166** |  |  |  |
| **Vi Văn Tuấn** | **K12** | **DCCNTT12.10.4** | **20211138** |  |  |  |
| **Lê Quý Mùi** | **K12** | **DCCNTT12.10.4** | **20211133** |  |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM 1**  *(Ký và ghi rõ họ tên)* | **CÁN BỘ CHẤM 2**  *(Ký và ghi rõ họ tên)* |

**Bắc Ninh, năm 2023**

Mục lục

[LỜI NÓI ĐẦU 3](#_Toc149221386)

[LỜI CẢM ƠN 4](#_Toc149221387)

[Chương I: Kế hoạch làm bài và giới thiệu về ngôn ngữ 5](#_Toc149221388)

[1. Khảo sát thực trạng 5](#_Toc149221389)

[2. Giới thiệu về ngôn ngữ JAVA 6](#_Toc149221390)

[2.1. Đặc điểm của ngôn ngữ lập trình Java 7](#_Toc149221391)

[2.2. Máy ảo Java (JVM – Java Virtual Machine) 9](#_Toc149221392)

[2.3. Các phiên bản của Java 11](#_Toc149221393)

[2.4. Các thành phần của Java SE Platform 11](#_Toc149221394)

[2.5. Khái niệm về lập trình hướng đối tượng trong java 12](#_Toc149221395)

[3. Giới thiệu về sql server 12](#_Toc149221396)

[a. Khái niệm SQL 12](#_Toc149221397)

[b. Lịch sử ra đời và các ấn bản của SQL server 13](#_Toc149221398)

[c. Các thành phần cơ bản trong SQL Server 13](#_Toc149221399)

[Chương II: Mô Hình Hệ Thống 17](#_Toc149221400)

[1. Biểu đồ phân rã chức năng 17](#_Toc149221401)

[2. Xây dựng sơ đồ user-case. 19](#_Toc149221402)

[3. Cơ Sở Dữ Liệu 23](#_Toc149221403)

[3.1 Các thực thể. 23](#_Toc149221404)

[3.2. Mô hình quan hệ. 26](#_Toc149221405)

[3.3 Chuẩn hóa 3NF 27](#_Toc149221406)

[4. Lược đồ quan hệ giữa các bảng 28](#_Toc149221407)

[5. Cấu trúc các bảng 29](#_Toc149221408)

[Chương III: Cài đặt chương trình 32](#_Toc149221409)

[1. Giao diện chương trình 32](#_Toc149221410)

[Kết Luận 40](#_Toc149221411)

[Tài liệu tham khảo 41](#_Toc149221412)

**DANH MỤC BẢNG BIỂU VÀ SƠ ĐỒ**

|  |  |  |
| --- | --- | --- |
| Số hiệu | Tên | Trang |
| Hình 1.1 | Biểu đồ phân rã chức năng | 16 |
| Hình 2.1 | Biểu đồ use-case | 18 |
| Hình 2.2 | Use-case xếp loại học tập | 19 |
| Hình 2.3 | Use-case quản lí lớp học | 20 |
| Hình 2.4 | Use-case quản lí sinh viên | 20 |
| Hình 2.5 | Use-case quản lí học phần | 21 |
| Hình 2.6 | Use-case quản lí điểm | 21 |
| Hình 3.1.1 | Thực thể lớp học | 22 |
| Hình 3.1.2 | Thực thể điểm | 23 |
| Hình 3.1.3 | Thực thể sinh viên | 23 |
| Hình 3.1.4 | Thực thể tài khoản | 24 |
| Hình 3.1.5 | Thực thể giảng viên | 24 |
| Hình 3.1.6 | Thực thể học phần | 25 |
| Hình 4 | Lược đồ quan hệ giữa các bảng | 27 |
| Bảng 5.1 | Bảng tài khoản | 28 |
| Bảng 5.2 | Bảng học phần | 28 |
| Bảng 5.3 | Bảng lớp học | 28 |
| Bảng 5.4 | Bảng sinh viên | 29 |
| Bảng 5.5 | Bảng điểm | 29 |
| Bảng 5.6 | Bảng giảng viên | 30 |
| Hình 3.1 | Giao diện trang đăng nhập | 31 |
| Hình 3.2 | Giao diện trang đăng ký | 31 |
| Hình 3.3 | Giao diện trang quên mật khẩu | 32 |
| Hình 3.4 | Giao diện trang chủ | 32 |
| Hình 3.5 | Giao diện trang sinh viên | 33 |
| Hình 3.6 | Giao diện khi thêm sinh viên | 34 |
| Hình 3.7 | Giao diện trang giảng viên | 34 |
| Hình 3.8 | Giao diện tìm kiếm giảng viên dựa theo mã lớp học | 34 |
| Hình 3.9 | Giao diện trang học phần | 35 |
| Hình 3.10 | Giao diện trang lớp học | 36 |
| Hình 3.11 | Giao diện trang điểm | 36 |
| Hình 3.12 | Giao diện trang tài khoản | 36 |
| Hình 3.13 | Giao diện trang chức năng hiển thị mật khẩu | 37 |
| Hình 3.14 | Giao diện trang chức năng đổi mật khẩu | 37 |
| Hình 3.15 | Giao diện khi đổi mật khẩu thành công | 28 |

# LỜI NÓI ĐẦU

Cùng với sự phát triển không ngừng về kỹ thuật máy tính và mạng điện tử, công nghệ thông tin cũng đạt được rất nhiều những thành tựu to lớn, góp công rất lớn trong công cuộc xây dựng và phát triển xã hội của nhân loại. Ngày nay, Công Nghệ Thông Tin có những bước phát triển mạnh mẽ, ứng dụng trong nhiều lĩnh vực đời sống và hiệu quả mang lại là không thể phụ nhận, giúp giảm thiểu tối đa các khâu làm việc thủ công kém hiệu quả chính xác của con người trước kia. Máy tính điện tử trở nên phổ biến và gần gũi với mọi người, việc tiếp cận với Hệ thống Thông tin quản lý trở thành mục tiêu hàng đầu của các tổ chức, doanh nghiệp cho mục đích quản thông tin của mình.

Qua yêu cầu thực tế đó và được sự hướng dẫn, giúp đỡ của thầy giáo Đinh Công Tùng nhóm em đã chọn đề tài “Xây dựng ứng dụng quản lý điểm sinh viên”. Ứng dụng đóng vai trò quan trọng trong hỗ trợ quản lý điểm của sinh viên cũng như giúpgiảng viên có thể tra cứu điểm và cập nhật điểm sinh viên một cách nhanh chóng và chính xác.

# LỜI CẢM ƠN

Thông qua môn học Chuyên đề cơ sở, nhóm chúng em đã có cơ hội hoàn thành báo cáo với đề tài “Xây dựng ứng dụng quản lý điểm sinh viên”. Với lòng biết ơn sâu sắc nhất, nhóm em xin gửi lời cám ơn chân thành tới toàn thể thầy cô khoa Công nghệ thông tin những người đã dạy dỗ và truyền đạt những kiến thức quý báu giúp chúng em hoàn thành đề tài này.

Đặc biệt, chúng em xin bày tỏ lòng cảm ơn sâu sắc nhất tới thầy Đinh Công Tùng vì đã tận tình hướng dẫn nhóm em trong suốt quá trình. Với sự chỉ bảo của thầy, nhóm em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu từ đó đạt được kết quả tốt nhất.

*Một lần nữa nhóm em xin chân thành cảm ơn !*

# Chương I: Kế hoạch làm bài và giới thiệu về ngôn ngữ

## Khảo sát thực trạng

Trường đại học Công Nghệ Đông Á là trường mới thành lập. Do vậy, các khâu quản lí mang nặng tính thủ công. Mô hình hoạt động của hệ thống quản lí điểm được mô tả qua sơ đồ sau:

A diagram of a company

Description automatically generated

* Quy trình thống kê:

A diagram of a flowchart

Description automatically generated

Đánh giá:

Quy trình thủ công dựa trên sức người là chính đã bộc lộ những nhược điểm sau:

* Việc lưu trữ các thông tin như: thông tin sinh viên, điểm,…. Đều tiến hành thủ công bằng sổ sách và các chứng từ với một số lượng lớn và lưu trữ nhiều năm, chính vì vậy đã gây ra nhiều khó khăn cho công tác quản lí, tốn nhiều thời gian và công sức cho người trực tiếp điều hành
* Khi lưu trữ thủ công khiến việc sửa đổi trở nên khó khăn
* Tốn nhiều nhân lực

## 2. Giới thiệu về ngôn ngữ JAVA

Java là gì?

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

### 2.1. Đặc điểm của ngôn ngữ lập trình Java

Tương tự C++, hướng đối tượng hoàn toàn

Trong quá trình tạo ra một ngôn ngữ mới phục vụ cho mục đích chạy được trên nhiều nền tảng, các kỹ sư của Sun MicroSystem muốn tạo ra một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Vì vậy họ đã sử dụng lại các cú pháp của C và C++.

Tuy nhiên, trong Java thao tác với con trỏ bị lược bỏ nhằm đảo bảo tính an toàn và dễ sử dụng hơn. Các thao tác overload, goto hay các cấu trúc như struct và union cũng được loại bỏ khỏi Java.

Độc lập phần cứng và hệ điều hành

Một chương trình viết bằng ngôn ngữ Java có thể chạy tốt ở nhiều môi trường khác nhau. Gọi là khả năng “cross-platform”. Khả năng độc lập phần cứng và hệ điều hành được thể hiện ở 2 cấp độ là cấp độ mã nguồn và cấp độ nhị phân.

Ở cấp độ mã nguồn: Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.

Ở cấp độ nhị phân: Một mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có Java Virtual Machine để thông dịch đoạn mã này.

Ngôn ngữ thông dịch

Ngôn ngữ lập trình thường được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch.

* Thông dịch (Interpreter) : Nó dịch từng lệnh rồi chạy từng lệnh, lần sau muốn chạy lại thì phải dịch lại.
* Biên dịch (Compiler): Code sau khi được biên dịch sẽ tạo ra 1 file thường là .exe, và file .exe này có thể đem sử dụng lại không cần biên dịch nữa.

Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch. Cụ thể như sau:

Khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã byte code. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã byte code này thành machine code (hay native code) khi nhận được yêu cầu chạy chương trình. A diagram of a computer process

Description automatically generated

Ưu điểm : Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện là JVM có hỗ trợ chạy trên nền tảng này.

Nhược điểm : Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong một mức chấp nhận được).

Cơ chế thu gom rác tự động:

Khi tạo ra các đối tượng trong Java, JRE sẽ tự động cấp phát không gian bộ nhớ cho các đối tượng ở trên heap.

Với ngôn ngữ như C C++, bạn sẽ phải yêu cầu hủy vùng nhớ mà bạn đã cấp phát, để tránh việc thất thoát vùng nhớ. Tuy nhiên vì một lý do nào đó, bạn không hủy một vài vùng nhớ, dẫn đến việc thất thoát và làm giảm hiệu năng chương trình.

Ngôn ngữ lập trình Java hỗ trợ cho bạn điều đó, nghĩa là bạn không phải tự gọi hủy các vùng nhớ. Bộ thu dọn rác của Java sẽ theo vết các tài nguyên đã được cấp. Khi không có tham chiếu nào đến vùng nhớ, bộ thu dọn rác sẽ tiến hành thu hồi vùng nhớ đã được cấp phát.

Đa luồng:

Java hỗ trợ lập trình đa tiến trình (multithread) để thực thi các công việc đồng thời. Đồng thời cũng cung cấp giải pháp đồng bộ giữa các tiến trình (giải pháp sử dụng priority…).

Tính an toàn và bảo mật

* Tính an toàn
* Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu.
* Dữ liệu phải được khai báo tường minh.
* Không sử dụng con trỏ và các phép toán với con trỏ.
* Java kiểm soát chặt chẽ việc truy nhập đến mảng, chuỗi. Không cho phép sử dụng các kỹ thuật tràn. Do đó các truy nhập sẽ không vượt quá kích thước của mảng hoặc chuỗi.
* Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động.
* Cơ chế xử lý lỗi giúp việc xử lý và phục hồi lỗi dễ dàng hơn.
* Tính bảo mật
* Java cung cấp một môi trường quản lý chương trình với nhiều mức khác nhau.
* Mức 1 : Chỉ có thể truy xuất dữ liệu cũng như phương phức thông qua giao diện mà lớp cung cấp.
* Mức 2 : Trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch.
* Mức 3 : Trình thông dịch sẽ kiểm tra mã byte code xem các đoạn mã này có đảm bảo được các quy định, quy tắc trước khi thực thi.
* Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

### 2.2. Máy ảo Java (JVM – Java Virtual Machine)

Để đảm bảo tính đa nền, Java sử dụng cơ chế Máy ảo của Java.

ByteCode là ngôn ngữ máy của Máy ảo Java tương tự như các lệnh nhị phân của các máy tính thực.

Một chương trình sau khi được viết bằng ngôn ngữ Java (có phần mở rộng là .java) phải được biên dịch thành tập tin thực thi được trên máy ảo Java (có phần mở rộng là .class). Tập tin thực thi này chứa các chỉ thị dưới dạng mã Bytecode mà máy ảo Java hiểu được phải làm gì.

Khi thực hiện một chương trình, máy ảo Java lần lượt thông dịch các chỉ thị dưới dạng Bytecode thành các chỉ thị dạng nhị phân của máy tính thực và thực thi thực sự chúng trên máy tính thực (còn gọi là khả năng khả chuyển).

Máy ảo thực tế đó là một chương trình thông dịch. Vì thế các hệ điều hành khác nhau sẽ có các máy ảo khác nhau. Để thực thi một ứng dụng của Java trên một hệ điều hành cụ thể, cần phải cài đặt máy ảo tương ứng cho hệ điều hành đó.

JVM cung cấp môi trường thực thi cho chương trình Java (còn gọi là khả năng độc lập với nền).

Có nhiều JVM cho các nền tảng khác nhau chẳng hạn như: Windows, Liux, và Mac.

A diagram of a java code

Description automatically generated

Các loại ứng dụng được phát triển sử dụng Java:

**A diagram of a software application

Description automatically generated**

### 2.3. Các phiên bản của Java

A diagram of different types of electronics

Description automatically generated

Java Standard Edition (Java SE) – Là một nền tảng cơ bản cho phép phát triển giao diện điều khiển, các ứng dụng mạng và các ứng dụng dạng Win Form.

Java Enterprise Edition (Java EE) – Được xây dựng trên nền tảng Java SE, giúp phát triển các ứng dụng web, các ứng dụng ở cấp doanh nghiệp, …

Java Mobile Edition (Java ME) – Là một nền tảng cho phép phát triển các ứng dụng nhúng vào các thiết bị điện tử như mobile,…

### 2.4. Các thành phần của Java SE Platform

Gồm 2 thành phần:

* JRE (Java Runtime Environment): cung cấp JVM (Java Virtual Machine) và thư viện được sử dụng để chạy chương trình Java.
* JDK (Java Development Kit): được biết đến như bộ cung cụ phát triển Java, bao gồm: trình biên dịch và trình gỡ rối được sử dụng để phát triển các ứng dụng Java.

A table with different colored squares

Description automatically generated with medium confidence

### 2.5. Khái niệm về lập trình hướng đối tượng trong java

Lập trình hướng đối tượng (Object Oriented Programing – OOP) là một phương pháp để thiết kế một chương trình bởi sử dụng các lớp và các đối tượng.

Java là một ngôn ngữ lập trình hướng đối tượng vì vậy nó cũng hỗ trợ các đặc tính của lập trình hướng đối tượng:

* Đa hình (Polymorphism)
* Thừa kế (Inheritance)
* Đóng gói (Encapsulation)
* Trừu tượng (Abstraction)

## 3. Giới thiệu về sql server

### a. Khái niệm SQL

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

### b. Lịch sử ra đời và các ấn bản của SQL server

+ Lịch sử ra đời SQL server

Năm 1989, phiên bản đầu tiên của SQL Server 1.0 ra đời được dùng cho các hệ điều hành 16 bit và được phát triển cho tới ngày nay.

Cho tới khi SQL Server ra phiên bản 6.5 thì được thị trường chấp nhận rộng rãi. Một đột phá cải tiến cho SQL Server 7.0 khi được Microsoft viết lại một engine hoàn toàn mới. Đến khi SQL Server từ phiên bản 7.0 cải tiến lên 8.0 chủ yếu phát triển về tính năng thiết kế website.

Cho đến ngày nay thì phiên bản mới nhất đó là SQL Server 2016 hỗ trợ bộ vi xử lý 64 bit ra đời vào ngày 1 tháng 6 năm 2016.

+ Một vài ấn bản SQL Server

Enterprise: là một ấn bản chứa tất cả các đặc điểm nổ bật của SQL Server như: các công cụ cho tạo và quản lý phân cụm SQL Server, nhân bộ máy cơ sở dữ liệu và một số dịch vụ đi kèm. Nó có thể đánh địa chỉ 12 terabytes và quản lý cơ sở dữ liệu lên tới 524 petabytes.

Standard: Ấn bản này có thể chạy tốt trên hệ thống lên tới 4 CPU và 2 GB RAM rất thích hợp cho các dịch vụ thiết kế web vừa và nhỏ.

Developer: Ấn bản này giới hạn số lượng người kết nối với server nhưng có đầy đủ các tính năng của Enterprise Edition. Đây là phiên bản được sử dụng cho kiểm tra và phát triển ứng dụng phù hợp cho các cá nhân trong lĩnh vực web như: freelancer Việt Nam, website spa,…

Workgroup: ấn bản SQL Server này có các chức năng lõi cơ sở dữ liệu nhưng không đi kèm các dịch vụ. Ở phiên bản 2012 không có ấn bản này.

Express: Ấn bản này dễ dàng sử dụng và quản trị cơ sở dữ liệu đơn giản.

### c. Các thành phần cơ bản trong SQL Server

- Các thành cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service,…

- Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

+ Database Engine: Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho ệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.

+ Integration Services: là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu. Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access,… và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.

+ Analysis Services: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.

+ Notification Services: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người dăng ký sử dụng trên nhiều loại thiết bị khác nhau.

+ Reporting Services: là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

+ Full Text Search Service: là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.

+ Service Broker: là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

Cấu trúc của SQL Server

Sơ đồ dưới đây minh họa cấu trúc của SQL Server:

A computer screen shot of a computer

Description automatically generated SQL Server bao gồm 2 thành phần chính: Database Engine và SQLOS.

Database Engine

* Thành phần cốt lõi của SQL Server là Database Engine. Nó bao gồm Relation Engine giúp xử lý các truy vấn và Storage Engine giúp quản lý các tệp cơ sở dữ liệu, các trang, chỉ mục,…
* Dưới Database Engine là hệ điều hành SQL Server hay SQLOS (SQL Server Operating System). SQLOS giúp quản lý bộ nhớ và I/O, lên lịch nhiệm vụ và khóa dữ liệu để tránh các xung đột xảy ra khi update.

SQL Server cung cấp dịch vụ và công cụ gì?

* Microsoft cung cấp công cụ, dịch vụ quản lý dữ liệu (data management) và BI.
* Với quản lý dữ liệu, SQL Server cung cấp các dịch vụ như: SQL Server Integration Services (SSIS), SQL Server Data Quality Services và SQL Server Master Data Services. Để phát triển cơ sở dữ liệu, ta có thể sử dụng công cụ SQL Server Data. Để quản lý, triển khai và giám sát cơ sở dữ liệu ta có SQL Server Management Studio (SSMS).
* Với việc phân tích dữ liệu, SQL Server cung cấp SQL Server Analysis Services (SSAS), SQL Server Reporting Services (SSRS) và The Machine Learning Services (R Services).

Các ấn bản SQL Server

SQL Server có bốn phiên bản chính với những dịch vụ khác nhau:

* Developer: sử dụng cho việc phát triển và thử nghiệm cơ sở dữ liệu.
* Expression: dành cho lượng cơ sở dữ liệu nhỏ, dung lượng lưu trữ trên đĩa không quá 10GB.
* Enterprise: sử dụng cho những ứng dụng lớn hơn và quan trọng hơn, phiên bản này bao gồm tất cả các tính năng của SQL Server.
* Standard: ở ấn bản này chứa một phần tính năng của ấn bản Enterprise và giới hạn về cấu hình số lượng lõi bộ xử lý và bộ nhớ trên máy chủ.

SQL Server giúp bạn làm việc dễ dàng hơn như thế nào?

* Xét qua những ưu điểm cũng như thuận lợi của SQL Server và ta có thể hiểu nó giúp cho công việc của Developer dễ dàng hơn như thế nào.
* Giao diện SQL Server rất dễ dàng để các Backend Developer tập trung nhiều vào việc chăm sóc dữ liệu hơn là việc nó hoạt động như thế nào.
* Tích hợp với giao diện người dùng: SQL Server được tích hợp với ứng dụng giao diện người dùng, thường là các ứng dụng web để cung cấp cơ chế thay đổi dữ liệu động.
* Vì là sản phẩm của Microsoft nên việc tích hợp framework .Net sẽ dễ dàng hơn do chúng có cùng một tổ chức.
* So với các phương tiện lưu trữ dữ liệu khác như Excel, văn bản,… cơ sở dữ liệu luôn được ưa chuộng hơn chủ yếu do khả năng lưu trữ lớn, tính bảo mật cao và sức mạnh xử lý dữ liệu.

# Chương II: Mô Hình Hệ Thống

## 1. Biểu đồ phân rã chức năng

A black and white rectangular object with text

Description automatically generated

*Hình 1.1: Biểu đồ phân rã chức năng*

**Mô tả:**

Hệ thống:

* Đăng nhập: Kiểm tra thông tin đăng nhập có tồn tại hay không, đăng nhập hệ thống
* Đăng ký: Đăng kí thông tin tài khoản cá nhân để đăng nhập hệ thống dưới quyền Sinh Viên
* Đăng xuất: Dùng để thoát hệ thống sau khi hoàn thành tác vụ

Quản lý tài khoản :

* Đổi mật khẩu: người dùng có thể mật khẩu cho tài khoản của mình

Quản lý sinh viên:

* Thêm: Cho phép người QL thêm một sinh viên mới vào hệ thống
* Sửa: Chỉnh sửa thông tin trên tài khoản của sinh viên
* Xóa sinh viên: Xoá một hoặc nhiều sinh viên ra khỏi hệ thống
* Danh sách sinh viên: Xuất ra một loạt danh sách sinh viên có trên hệ thống

Quản lý giảng viên:

* Thêm: Thêm giảng viên mới vào hệ thống
* Sửa: Sửa thông tin giảng viên
* Xoá giảng viên: Xoá một hoặc nhiều giảng viên ra khỏi hệ thống
* Danh sách giảng viên: Xuất ra một loạt danh sách giảng viên có trên hệ thống

Quản lý điểm:

* Thêm: Nhập điểm cho actor sinh viên
* Sửa: Cho phép người QL có thể sửa hoặc thay đổi điểm số trên hệ thống của sinh viên khi xảy ra nhầm lẫn
* Xóa bảng điểm: Xoá một hoặc nhiều bảng điểm sinh viên
* Danh sách bảng điểm: Xuất ra một loạt danh sách điểm của sinh viên

Quản lý môn học:

* Thêm: cho phép người QL thêm một môn học mới lên hệ thống
* Sửa: Sửa đổi một số môn học đã tồn tại trên hệ thống
* Xóa môn học: Xoá một hoặc nhiều môn học trên hệ thống
* Danh sách bảng môn học: Xuất ra một loạt danh sách môn học

Tìm kiếm:

* Tìm kiếm sinh viên: Dùng để tìm kiếm sinh viên trong số danh sách sinh viên đã tồn tại trên hệ thống
* Tìm kiếm môn học: Dùng để tìm kiếm môn học đã tồn tại trên hệ thống

Báo cáo và thống kê :

* Báo cáo danh sách sinh viên: Đưa ra báo cáo danh sách sinh viên, tổng số sinh viên, có thể theo lớp hoặc khoa.
* Báo cáo danh sách bảng điểm: Đưa ra báo cáo điểm của sinh viên, tổng số điểm của sinh viên.

## 2. Xây dựng sơ đồ user-case.

Biểu đồ gồm 2 tác nhân chính :

A screenshot of a computer

Description automatically generated

*Hình 2.1 : Biểu đồ use-case*

**Mô tả:**

Actor Hệ thống:

* Đăng nhập
* QL lớp học
* QL Sinh viên
* QL học phần
* QL điểm
* QL giảng viên

Actor Giảng Viên:

* Đăng nhập
* Đăng kí
* Đổi mật khẩu
* Thêm thông tin
* Sửa thông tin
* Xóa thông tin
* Tìm kiếm

Use case chi tiết cho xếp loại học tập:

A diagram of a network

Description automatically generated

*Hình 2.2: Use-case xếp loại học tập*

Trong chức năng này, nhân viên quản lí có thể phải tương tác với các giao diện:

* Đăng nhập -> thống nhất với UC đăng nhập
* Xem thống kê loại học lực -> đề xuất UC xem TK loại học lực
* Xem thống kê các sinh viên của 1 loại học lực -> đề xuất UC xem TK sinh viên của loại học lực.
* Xem kết quả của một sinh viên -> đề xuất UC xem điểm các môn học của sinh viên
* Xem chi tiết một môn học của sinh viên -> đề xuất UC xem điểm chi tiết 1 môn học của sinh viên.
* Các giao diện thống kê lần lượt theo dạng: giao diện sau là tùy chọn từ giao diện trước. Do đó, chúng có quan hệ mở rộng lần lượt cái sau từ cái trước.

Như vậy, biểu đồ UC chi tiết cho chức năng thống kê loại học lực được trình bày như trong hình vẽ. Trong đó, các UC chi tiết được mô tả như sau:

* Xem TK loại học lực: UC này cho phép NVQL xem thống kê số lượng sinh viên trong mỗi loại học lực của một kì học
* Xem TK sinh viên của một loại học lực: UC này cho phép NVQL xem TK kết quả các sinh viên của một loại học lực
* Xem TK điểm của một sinh viên: UC này cho phép NVQL xem kết quả các môn học của một sinh viên trong một kì học
* Xem kết quả một một học của một sinh viên: UC này cho phép NVQL xem kết quả chi tiết một môn học của một sinh viên.

Use case chi tiết cho quản lí lớp học:

A diagram of a diagram

Description automatically generated

*Hình 2.3: Use-case quản lí lớp học*

Use case chi tiết cho quản lí sinh viên:

A diagram of a computer

Description automatically generated

*Hình 2.4: Use-case quản lí sinh viên*

Use case chi tiết quản lí học phần:

A diagram of a computer

Description automatically generated

*Hình 2.5: Use-case quản lí học phần*

Use case chi tiết quản lí điểm:

A diagram of a diagram

Description automatically generated

*Hình 2.6: Use-case quản lí điểm*

## 3. Cơ Sở Dữ Liệu

### 3.1 Các thực thể.

*A diagram of a structure

Description automatically generated*

*Hình 3.1.1: Thực thể lớp học*

A diagram of a company

Description automatically generated

*Hình 3.1.2: Thực thể điểm*

*A diagram of a company

Description automatically generated*

*Hình 3.1.3: Thực thể sinh viên*

*A diagram with text and circles

Description automatically generated with medium confidence*

*Hình 3.1.4: Thực thể tài khoản*

*A diagram of a network

Description automatically generated*

*Hình 3.1.5: Thực thể giảng viên*

*A diagram of a company

Description automatically generated*

*Hình 3.1.6: Thực thể học phần*

### 3.2. Mô hình quan hệ.

Lophoc(maLH, tenLH)

Diem(*idD*, diemGK, diemCC, diem CK, diemTK,maSV,maHP,maLH)

Sinhvien(*maSV*, hoVaTen, gioiTinh, diaChi, ngaySinh, idHP, idLH)

GiangVien(*maGV*, hoVaTen, gioiTinh, diaChi, ngaySinh,maLH,maHP)

Taikhoan(tenDangNhap,matKhau,xacnhanmk,email,MaGV)

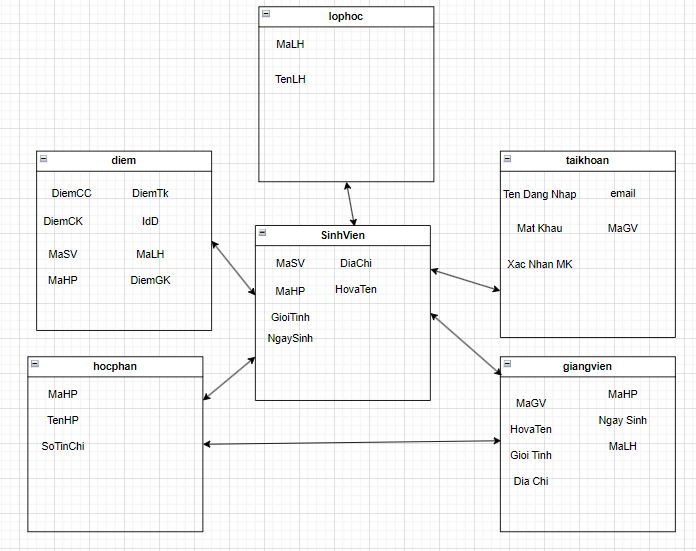
Hocphan(maHP, tenHP, soTinChi)

### 3.3 Chuẩn hóa 3NF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Các thuộc tính | 1nf | 2nf | 3nf | Ten bang |
| maLH, tenLH,  *idD*, diemGK, diemCC, diemCK,diemTK,maSV,maHP,MaLH  *maSV*,hoVaTen,gioiTinh,diaChi,ngaySinh,idHP,idLH,  tenDangNhap, matKhau  *maGV*, hoVaTen, gioiTinh, diaChi, ngaySinh, idLH  maHP,tenHP,soTinChi | maLH, tenLH  *idD*, diemGK, diemCC, diem CK, diemTK, |  |  |  |
| *maLH* |  | Lophoc |
| diemCC, diem CK, diemTK, | diemCC, diem CK, diemTK, | Diem |
|  | maHP, tenHP,soTinChi | maHP, tenHP,soTinChi | Hocphan |
| maSV,hoVaTen, gioiTinh, diaChi, ngaySinh, | maSV,hoVaTen,maLH,maHP | maSV,hoVaTen | Sinhvien |
|  | tenDangNhap,matKhau  *maGV*, email | tenDangNhap,matKhau,xacnhanmk | tenDangNhap,matKhau | Taikhoan |
|  |  | *maGV*, hoVaTen, gioiTinh | *maGV*, hoVaTen, | GiangVien |

## 4. Lược đồ quan hệ giữa các bảng

Trên cơ sở phân tích yêu cầu hệ thống ở chương 1, ta thấy hệ thống có các bảng được kết nối với nhau theo hình sau

****

*Hình 4: Lược đồ quan hệ giữa các bảng*

## 5. Cấu trúc các bảng

#### 5.1 Bảng taikhoan

Bảng taikhoan dùng để lưu trữ thông tin tài khoản gồm các trường được cho trong bảng sau:

*Bảng 5.1. Bảng tài khoản*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | tenDangNhap | Varchar | Tên đăng nhập |
| 2 | matKhau | Varchar | Mật khẩu |
| 3 | xacnhanMK | Varchar | Xác nhận mật khẩu |
| 4 | email | Varchar | Email |
| 5 | maGV | Varchar | Mã Giảng Viên |

#### 5.2 Bảng hocphan

Bảng hocphan dùng để lưu trữ dữ liệu về các học phần, bao gồm các trường được cho trong bảng sau:

*Bảng 5.2. Bảng học phần*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | maHP | Varchar | Mã học phần |
| 2 | tenHP | Varchar | Tên học phần |
| 3 | soTinChi | Int | Số tín chỉ |

#### 5.3 Bảng lophoc

Bảng lophoc dùng để lữu trữ dữ liệu về lớp học học, bao gồm các trường được cho trong bảng sau:

*Bảng 5.3. Bảng lớp học*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | maLH | Varchar | Mã lớp học |
| 2 | tenLH | Varchar | Tên lớp học |

#### 5.4 Bảng sinhvien

Bảng sinhvien dùng để lữu trữ dữ liệu về sinh viên, bao gồm các trường được cho trong bảng sau:

*Bảng 5.4. Bảng sinh viên*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | maSV | Varchar | Mã sinh viên |
| 2 | hoVaTen | Varchar | Họ và tên |
| 3 | gioiTinh | Varchar | Giới tính |
| 4 | diaChi | Varchar | Địa chỉ |
| 5 | ngaySinh | date | Ngày sinh |
| 6 | maLH | Varchar | Mã lớp học |
| 7 | maHP | Varchar | Mã học phần |

#### 5.5 Bảng diem

Bảng diem dùng để lữu trữ dữ liệu về điểm, bao gồm các trường được cho trong bảng sau:

*Bảng 5.5. Bảng điểm*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | idD | Int | Id điểm |
| 2 | diemGK | Double | Điểm giữa kì |
| 3 | diemCC | Double | Điểm chuyên cần |
| 4 | diemCK | Double | Điểm cuối kì |
| 5 | diemTK | Double | Điểm tổng kết |
| 6 | maSV | Int | Mã sinh viên |
| 7 | maHP | Int | Mã học phần |
| 8 | maLH | Int | Mã lớp học |

#### 5.6 Bảng giangvien

Bảng giangvien dùng để lữu trữ dữ liệu về sinh viên, bao gồm các trường được cho trong bảng sau:

*Bảng 5.6. Bảng giảng viên*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | maGV | Varchar | Mã sinh viên |
| 2 | hoVaTen | Varchar | Họ và tên |
| 3 | gioiTinh | Varchar | Giới tính |
| 4 | diaChi | Varchar | Địa chỉ |
| 5 | ngaySinh | date | Ngày sinh |
| 6 | maHP | Varchar | Mã học phần |
| 7 | maLH | Varchar | Mã lớp học |

# Chương III: Cài đặt chương trình

## Giao diện chương trình

A screenshot of a computer screen

Description automatically generated

*Hình 3.1: Giao diện trang đăng nhập*

A screenshot of a computer

Description automatically generated

*Hình 3.2: Giao diện trang đăng ký*

**A screenshot of a computer

Description automatically generated**

*Hình 3.3: Giao diện trang quên mật khẩu*

A screenshot of a computer

Description automatically generated

*Hình 3.4: Giao diện trang chủ*

A screenshot of a computer

Description automatically generated

*Hình 3.5: Giao diện trang sinh viên*

A screenshot of a computer

Description automatically generated

*Hình 3.6: Giao diện khi thêm sinh viên*

**A screenshot of a computer

Description automatically generated**

*Hình 3.7: Giao diện trang giảng viên*

A screenshot of a computer

Description automatically generated

*Hình 3.8: Giao diện tìm kiếm giảng viên dựa theo mã lớp học*

**A screenshot of a computer

Description automatically generated**

*Hình 3.9: Giao diện trang học phần*

A screenshot of a computer

Description automatically generated

*Hình 3.10: Giao diện trang lớp học*

A screenshot of a computer

Description automatically generated

*Hình 3.11: Giao diện trang điểm*

A screenshot of a login form

Description automatically generated

*Hình 3.12: Giao diện trang tài khoản*

A screenshot of a computer

Description automatically generated

*Hình 3.13: Giao diện trang chức năng hiển thị mật khẩu*

A screenshot of a computer

Description automatically generated

*Hình 3.14: Giao diện trang chức năng đổi mật khẩu*

**A screenshot of a computer

Description automatically generated**

*Hình 3.15: Giao diện khi đổi mật khẩu thành công*

# Kết Luận

**I. Kết luận**

* Đứng trước xu hướng phát triển của công nghệ thông tin như vũ bão, việc tin học hoá quản lý hành chính là vô cùng quan trọng và cần thiết. Nó sẽ giúp cho các trường học, trung tâm rèn luyện,… giải quyết được các vấn đề nhanh chóng, hiệu quả và tối ưu nhất. Ứng dụng cơ sở dữ liệu đã giải quyết được vấn đề đó.
* Với đề tài: Ứng đụng quản lý điểm sinh viên, tuy chương trình cài đặt chưa được hoàn thiện các chứng năng, song nếu có thời gian phát triển và hoàn thiện hơn thì chương trình này sẽ có ích rất nhiều giúp cho các nhân viên của các trường đại học, trung tâm,… trong việc quản lý học sinh, sinh viên một cách thuận tiên, dễ dàng và triệt để hơn rất nhiều

**II. Hạn chế**

* Do thời gian thực hiện tương đối hạn chế nên chương trình không thể tránh được những thiếu sót nhất định. Bên cạnh đó chương tình còn có một số chức năng chưa đủ thời gian để thực hiện kịp. Đồng thời có một số form chưa hoàn thiện đầy đủ và chính xác các ràng buộc về dữ liệu vẫn chưa đầy đủ lắm. Hơn nữa trong quá trình vừa viết chương trình vừa tìm hiểu về ngôn nữ nên chưa thể tránh khỏi những thiếu sót

**III. Hướng phát triển**

* Để chương trình có thể hoạt động hiệu quả hơn cần dữ liệu phải được lưu trữ bằng hệ quản trị cơ sở dữ liệu có tính bảo mật hơn. Cần có sự kết hợp xong song giữa quản lý nhân viên và học sinh một cách chặt chẽ hơn để việc theo dõi và quản lý của các trường học được tốt hơn.
* Nếu có thể đưa chương trình quản lý các thông tin về học sinh lên các trang web để giúp cho phụ huynh của học sinh có sự thuận lợi hơn để theo dõi được quả trình học tập của con em cho dù đang ngồi ở nhà hay ở bất kỳ nơi nào.

# Tài liệu tham khảo

Trần Xuân Thanh, IT3219 - Lập trình hướng đối tượng, elearning.eaut

<http://elearning.eaut.edu.vn/course/view.php?id=42>

“Giải ngố” về SQL Server – Hệ quản trị cơ sở dữ liệu tốt nhất, FPT smart cloud

<https://microsoft.fptcloud.com/kien-thuc/giai-ngo-ve-sql-server/>