**1.1 Giới thiệu đề tài**

**Tên sản phẩm:** *“Phần mềm Quản lý sinh viên”*

Việc quản lý thông tin sinh viên trong các trường học luôn là một vấn đề được quan tâm. Hệ thống quản lý truyền thống dựa trên sổ sách hoặc bảng tính có thể gây khó khăn trong việc xử lý dữ liệu lớn, dễ gặp sai sót và mất thời gian khi cần tìm kiếm hay thống kê. Chính vì vậy, nhóm đã thực hiện đề tài *“Phần mềm Quản lý sinh viên”* nhằm xây dựng một công cụ số hóa quy trình quản lý sinh viên, giúp tiết kiệm thời gian, tăng độ chính xác và tính hiệu quả.

* **Mục tiêu chính:**
  + Xây dựng một hệ thống quản lý sinh viên có khả năng lưu trữ, chỉnh sửa, tìm kiếm và thống kê dữ liệu sinh viên nhanh chóng và chính xác.
  + Tích hợp các chức năng đánh giá rèn luyện, quản lý ký túc xá và báo cáo thống kê, hỗ trợ tối đa cho cán bộ quản lý của trường.
* **Các tính năng nổi bật:**
  + **Quản lý thông tin cá nhân của sinh viên:**
    - Lưu trữ thông tin như mã sinh viên, họ tên, ngày sinh, lớp học.
    - Cập nhật và tìm kiếm thông tin theo mã sinh viên hoặc các tiêu chí khác.
  + **Quản lý ký túc xá:**
    - Phân chia phòng ở và kiểm soát danh sách sinh viên đăng ký.
    - Quản lý chi phí, nợ tiền phòng ký túc xá.
  + **Đánh giá rèn luyện:**
    - Hỗ trợ cán bộ và sinh viên nhập điểm rèn luyện.
    - Tự động tổng hợp và xếp loại.
  + **Thống kê và báo cáo:**
    - Lập báo cáo theo lớp, khoa hoặc toàn trường.
    - Xuất dữ liệu dưới dạng bảng Excel hoặc PDF.
* **Phạm vi áp dụng:**
  + Sản phẩm này được thiết kế dành cho **Đại học Công nghiệp Hà Nội**, nhưng các chức năng có thể tùy chỉnh để phù hợp với yêu cầu của các trường học hoặc tổ chức khác.
* **Hình thức sản phẩm:**
  + Ứng dụng desktop được phát triển trên nền tảng Java, với giao diện thân thiện, đơn giản, dễ sử dụng.
  + Sử dụng cơ sở dữ liệu SQL Server để lưu trữ thông tin tập trung, đảm bảo dữ liệu an toàn và dễ dàng truy xuất.
* **Kết quả mong đợi:**
  + Xây dựng một sản phẩm hoàn chỉnh với đầy đủ chức năng nêu trên.
  + Đảm bảo phần mềm dễ dàng cài đặt và sử dụng, có thể mở rộng trong tương lai.
  + Tăng hiệu quả quản lý sinh viên, giảm thiểu sai sót và tiết kiệm thời gian.

**1.2 Kiến thức cần chuẩn bị**

Để thực hiện thành công đề tài này, nhóm cần trang bị các kiến thức lý thuyết và kỹ năng thực hành liên quan đến lập trình Java, cơ sở dữ liệu, thiết kế hệ thống phần mềm và sử dụng các công cụ hỗ trợ. Dưới đây là nội dung chi tiết:

**1.2.1 Lập trình hướng đối tượng (OOP)**

**Lập trình hướng đối tượng** là một trong những kỹ thuật lập trình quan trọng nhất khi phát triển phần mềm với Java. Kỹ thuật này giúp tổ chức mã nguồn theo cách dễ hiểu và dễ bảo trì, đồng thời tận dụng khả năng tái sử dụng mã để xây dựng các hệ thống phức tạp.

1. **Các khái niệm cơ bản:**
   * **Lớp (Class):**
     + Là bản thiết kế của đối tượng, chứa các thuộc tính và phương thức.
     + Ví dụ: Lớp Student có các thuộc tính như studentID, name, dob, và các phương thức như addStudent(), deleteStudent().
   * **Đối tượng (Object):**
     + Là một thể hiện cụ thể của lớp. Ví dụ, Student student1 = new Student();.
     + Đối tượng được tạo để lưu trữ thông tin cụ thể và thực hiện hành động cụ thể.
   * **Thuộc tính (Attribute):**
     + Đại diện cho dữ liệu bên trong đối tượng, ví dụ như String name, int age.
   * **Phương thức (Method):**
     + Đại diện cho các hành động của đối tượng, ví dụ void displayInfo() để in thông tin sinh viên.
2. **Các tính chất cơ bản của OOP:**
   * **Tính đóng gói (Encapsulation):**
     + Đóng gói dữ liệu và phương thức bên trong lớp để bảo vệ chúng khỏi truy cập trái phép.
     + Sử dụng các từ khóa như private, protected, public để kiểm soát quyền truy cập.
   * **Tính kế thừa (Inheritance):**
     + Một lớp con có thể kế thừa thuộc tính và phương thức của lớp cha.
     + Ví dụ: class GraduateStudent extends Student { ... }.
   * **Tính đa hình (Polymorphism):**
     + Một đối tượng hoặc phương thức có thể có nhiều hình thái khác nhau.
     + Gồm ghi đè phương thức (@Override) và nạp chồng phương thức (Overloading).
   * **Tính trừu tượng (Abstraction):**
     + Che giấu các chi tiết phức tạp, chỉ hiển thị các chức năng chính cho người dùng.
     + Ví dụ: Sử dụng lớp trừu tượng (abstract class) hoặc giao diện (interface).
3. **Các kỹ thuật nâng cao:**
   * **Constructor (Phương thức khởi tạo):**
     + Tự động được gọi khi tạo đối tượng, giúp khởi tạo giá trị ban đầu.
   * **Static Method:**
     + Phương thức không phụ thuộc vào đối tượng cụ thể, ví dụ: Math.sqrt().

**1.2.2 Java Collection Framework**

Java Collection Framework là tập hợp các cấu trúc dữ liệu và thuật toán để quản lý dữ liệu hiệu quả hơn.

1. **Các cấu trúc dữ liệu quan trọng:**
   * **List:**
     + Lưu trữ danh sách sinh viên với thứ tự, ví dụ: ArrayList<Student>.
   * **Set:**
     + Lưu dữ liệu không trùng lặp, ví dụ: HashSet.
   * **Map:**
     + Lưu trữ dữ liệu dưới dạng cặp khóa - giá trị, ví dụ: HashMap<Integer, Student>.
2. **Các thao tác chính:**
   * Thêm, sửa, xóa, tìm kiếm sinh viên trong danh sách.
   * Duyệt qua danh sách bằng vòng lặp for-each hoặc Iterator.
   * Sắp xếp danh sách sinh viên bằng Collections.sort() hoặc sử dụng Comparator.

**2. Công cụ thực hiện**

Để phát triển phần mềm *“Phần mềm Quản lý sinh viên”*, nhóm đã sử dụng các công cụ hỗ trợ phát triển phần mềm và quản lý cơ sở dữ liệu. Dưới đây là chi tiết các công cụ được áp dụng:

**2.1 JDK 17**

**JDK (Java Development Kit)** là nền tảng cốt lõi cho việc lập trình và phát triển các ứng dụng Java.

1. **Tổng quan về JDK 17:**
   * JDK 17 là phiên bản LTS (Long-Term Support), được hỗ trợ bảo mật và cập nhật lâu dài, đảm bảo sự ổn định trong quá trình phát triển phần mềm.
   * Đây là công cụ cần thiết để biên dịch, chạy và gỡ lỗi các chương trình Java.
2. **Các thành phần chính của JDK 17:**
   * **Compiler (javac):** Dùng để biên dịch mã nguồn Java (.java) thành bytecode (.class), giúp mã nguồn có thể chạy trên mọi nền tảng có JVM.
   * **Java Runtime Environment (JRE):** Cung cấp môi trường thực thi ứng dụng Java, bao gồm các thư viện lõi và máy ảo Java (JVM).
   * **Thư viện API:** Bao gồm các gói và lớp tiêu chuẩn, như java.util, java.sql, hỗ trợ lập trình viên thao tác với dữ liệu, kết nối cơ sở dữ liệu và xử lý luồng công việc.
3. **Vai trò trong đề tài:**
   * Biên dịch và thực thi chương trình quản lý sinh viên.
   * Cung cấp các thư viện tích hợp để xử lý danh sách sinh viên (Java Collections) và kết nối cơ sở dữ liệu (JDBC).
4. **Ưu điểm của JDK 17:**
   * Cải thiện hiệu năng của JVM, giảm thời gian khởi chạy ứng dụng.
   * Hỗ trợ tính năng "Sealed Classes", giúp kiểm soát kế thừa, đảm bảo an toàn trong thiết kế hệ thống.

**2.2 SQL Server**

**Microsoft SQL Server** là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, được sử dụng để lưu trữ và quản lý thông tin sinh viên trong hệ thống.

1. **Tổng quan về SQL Server:**
   * SQL Server cung cấp các công cụ mạnh mẽ để thiết kế cơ sở dữ liệu, truy vấn dữ liệu và quản lý các quy trình xử lý dữ liệu.
   * Giao diện đồ họa trực quan (SQL Server Management Studio - SSMS) giúp dễ dàng thao tác với cơ sở dữ liệu.
2. **Ứng dụng trong đề tài:**
   * **Thiết kế cơ sở dữ liệu:** Tạo các bảng như Students, Dormitories, Assessments.
   * **Lưu trữ dữ liệu:**
     + Thông tin sinh viên: Mã sinh viên, họ tên, ngày sinh, lớp học.
     + Dữ liệu đánh giá rèn luyện: Điểm rèn luyện theo kỳ, xếp loại.
     + Quản lý ký túc xá: Thông tin phòng ở, chi phí.
   * **Truy vấn và chỉnh sửa dữ liệu:**
     + Truy vấn danh sách sinh viên theo mã, lớp hoặc khoa.
     + Thêm, sửa, xóa thông tin sinh viên khi cần thiết.
3. **Ưu điểm:**
   * Đảm bảo tính bảo mật và toàn vẹn dữ liệu.
   * Hiệu suất cao khi xử lý khối lượng lớn thông tin.
   * Tích hợp tốt với ứng dụng Java thông qua JDBC.

**2.3 Rational Rose**

**Rational Rose** là một công cụ chuyên dụng trong thiết kế hệ thống phần mềm, đặc biệt là tạo các mô hình UML (Unified Modeling Language).

1. **Tổng quan về Rational Rose:**
   * Công cụ này hỗ trợ phân tích và thiết kế hệ thống trước khi triển khai mã nguồn, giúp lập trình viên hiểu rõ cấu trúc và luồng hoạt động của ứng dụng.
   * Cho phép tạo nhiều loại sơ đồ UML như Use Case, Class Diagram, Sequence Diagram.
2. **Ứng dụng trong đề tài:**
   * **Thiết kế sơ đồ lớp (Class Diagram):**
     + Lớp Student: Quản lý thông tin cá nhân của sinh viên.
     + Lớp Dormitory: Quản lý phòng ở và chi phí ký túc xá.
     + Lớp Assessment: Quản lý điểm rèn luyện.
   * **Thiết kế sơ đồ trường hợp sử dụng (Use Case Diagram):**
     + Xác định các chức năng chính như "Quản lý sinh viên", "Quản lý ký túc xá", "Thống kê báo cáo".
   * **Thiết kế sơ đồ trình tự (Sequence Diagram):**
     + Minh họa cách các đối tượng tương tác để thực hiện các chức năng cụ thể.
3. **Ưu điểm:**
   * Giúp hình dung rõ ràng cấu trúc và luồng công việc của hệ thống trước khi viết mã nguồn.
   * Tăng hiệu quả trong việc giao tiếp ý tưởng giữa các thành viên nhóm.

**2.4 Eclipse**

**Eclipse** là một môi trường phát triển tích hợp (IDE) phổ biến dành cho lập trình Java, với nhiều tính năng mạnh mẽ hỗ trợ lập trình viên trong việc viết, kiểm tra và gỡ lỗi mã nguồn.

1. **Tổng quan về Eclipse:**
   * Eclipse cung cấp giao diện trực quan, hỗ trợ lập trình viên thao tác với mã nguồn dễ dàng.
   * Hỗ trợ cài đặt plugin để mở rộng các tính năng như kết nối cơ sở dữ liệu, quản lý dự án.
2. **Ứng dụng trong đề tài:**
   * **Viết mã nguồn:** Là công cụ chính để lập trình và kiểm thử ứng dụng quản lý sinh viên.
   * **Debugging:** Xác định và sửa lỗi logic hoặc cú pháp trong chương trình.
   * **Quản lý dự án:** Cấu trúc mã nguồn theo chuẩn dự án Java, dễ dàng tổ chức và bảo trì.
   * **Tích hợp với công cụ khác:**
     + Kết nối cơ sở dữ liệu SQL Server qua JDBC.
     + Xuất báo cáo từ ứng dụng ra các định dạng như PDF hoặc Excel.
3. **Tính năng nổi bật:**
   * Hỗ trợ tự động hoàn thành mã nguồn (Code Completion).
   * Tích hợp công cụ kiểm tra lỗi cú pháp thời gian thực.
   * Hỗ trợ công cụ quản lý phiên bản như Git để lưu trữ và chia sẻ mã nguồn.
4. **Ưu điểm:**
   * Miễn phí, dễ sử dụng và mạnh mẽ.
   * Hỗ trợ phát triển các dự án Java từ đơn giản đến phức tạp.