**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**



**BÀI TẬP LỚN**

**HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG … TRÊN NỀN TẢNG ANDROID SỬ DỤNG … ANDROID STUDIO**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện** | **Lớp** | **Khóa** |
| **Nguyễn Văn A** | **DCCNTT12.10.10** | **12** |
| **Nguyễn Văn B** | **DCCNTT12.10.10** | **12** |
| **Bùi Thị C** | **DCCNTT12.10.10** | **12** |
| **Trần Văn D** | **DCCNTT12.10.10** | **12** |
| **…** | **…** | **…** |

**Bắc Ninh, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**BÀI TẬP LỚN**

**HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

**Nhóm: ...**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG … TRÊN NỀN TẢNG ANDROID SỬ DỤNG … ANDROID STUDIO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Sinh viên thực hiện** | **Mã sinh viên** | **Điểm bằng số** | **Điểm bằng chữ** |
| **1** | **Nguyễn Văn A** | **2021xxxx** |  |  |
| **2** | **Nguyễn Văn B** | **2021xxxx** |  |  |
| **3** | **Bùi Thị C** | **2021xxxx** |  |  |
| **4** | **Trần Văn D** | **2021xxxx** |  |  |
| **5** | **…** | **…** |  |  |

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

**Bắc Ninh, năm 2024**

**LỜI NÓI ĐẦU**

…

**MỤC LỤC**

[**CHƯƠNG 1. CƠ SỞ LÝ THUYẾT** 7](#_Toc177843329)

[**1.1 Tổng quan về ứng dụng cho thiết bị di động** 7](#_Toc177843330)

[***1.1.1*** ***Giới thiệu chung về xu hướng phát triển ứng dụng cho thiết bị di động*** 7](#_Toc177843331)

[***1.1.2 Tổng quan về các hệ điều hành thiết bị di động, các công cụ lập trình*** 7](#_Toc177843332)

[***1.1.3 Hệ điều hành Android*** 8](#_Toc177843333)

[***1.1.4 Công cụ lập trình Android Studio*** 8](#_Toc177843334)

[**1.2. Phát triển ứng dụng trên nền tảng Android sử dụng Java** 9](#_Toc177843335)

[***1.2.1. Cấu trúc ứng dụng Android*** 9](#_Toc177843336)

[***1.2.2. Các thành phần phát triển ứng dụng Android*** 9](#_Toc177843337)

[**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 10](#_Toc177843338)

[**2.1 Giới thiệu bài toán** 10](#_Toc177843339)

[**2.2 Phân tích hệ thống** 13](#_Toc177843340)

[***2.2.1. Sơ đồ chức năng của phần mềm*** 13](#_Toc177843341)

[***2.2.2. Danh sách các tác nhân (Actor)*** 14](#_Toc177843342)

[***2.2.3. Xác định các ca sử dụng (use-case)*** 15](#_Toc177843343)

[**2.3. Thiết kế hệ thống** 16](#_Toc177843344)

[***2.3.1. Đặc tả use-case tổng quát*** 16](#_Toc177843345)

[**2.4. Biểu đồ lớp** 22](#_Toc177843346)

[**2.5. Thiết kế cơ sở dữ liệu** 24](#_Toc177843347)

[**CHƯƠNG 3. CÀI ĐẶT CHƯƠNG TRÌNH** 27](#_Toc177843348)

[**3.1 Tổ chức chương trình** 27](#_Toc177843349)

[**3.3. Giao diện chính của chương trình** 27](#_Toc177843350)

[**CHƯƠNG 4. KẾT LUẬN** 28](#_Toc177843351)

[**4.1. Kết quả đạt được** 28](#_Toc177843352)

[***4.1.1. Hoàn thiện hệ thống quản lý sinh viên*** 28](#_Toc177843353)

[***4.1.2. Quản lý cơ sở dữ liệu hiệu quả*** 28](#_Toc177843354)

[***4.1.3. Đảm bảo bảo mật thông tin*** 28](#_Toc177843355)

[***4.1.4. Khả năng mở rộng và tích hợp cao*** 28](#_Toc177843356)

[**4.2. Hướng phát triển** 28](#_Toc177843357)

[***4.2.1. Nâng cấp và bổ sung chức năng mới*** 28](#_Toc177843358)

[***4.2.2. Tăng cường bảo mật hệ thống*** 29](#_Toc177843359)

[***4.2.3. Cải tiến hiệu suất và tối ưu hóa cơ sở dữ liệu*** 29](#_Toc177843360)

[***4.2.4. Phát triển ứng dụng di động*** 29](#_Toc177843361)

[***4.2.5. Phân tích và báo cáo dữ liệu*** 29](#_Toc177843362)

[**TÀI LIỆU THAM KHẢO** 30](#_Toc177843363)

[**PHỤ LỤC** 31](#_Toc177843364)

**DANH MỤC TỪ VIẾT TẮT**

**DANH MỤC HÌNH VẼ**

[Hình 1. Biểu đồ Use case tổng quát 18](#_Toc177843365)

[Hình 2. Biểu đồ Use case quản lý tài khoản người dùng 19](#_Toc177843366)

[Hình 3. Biểu đồ Use case quản lý sinh viên 20](#_Toc177843367)

[Hình 4. Biểu đồ Use case quản lý lớp học 21](#_Toc177843368)

[Hình 5. Biểu đồ Use case quản lý điểm số 22](#_Toc177843369)

[Hình 6. Biểu đồ Use case Đăng nhập/Đăng xuất 23](#_Toc177843370)

[Hình 7. Biểu đồ lớp 24](#_Toc177843371)

[Hình 8. Biểu đồ tổng quát 26](#_Toc177843372)

[Hình 9. Tổ chức chương trình 28](#_Toc177843373)

# **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

## **1.1 Tổng quan về ứng dụng cho thiết bị di động**

### ***Giới thiệu chung về xu hướng phát triển ứng dụng cho thiết bị di động***

Sự phát triển của công nghệ di động đã thúc đẩy một xu hướng mạnh mẽ trong việc xây dựng và sử dụng ứng dụng cho thiết bị di động. Hiện nay, các ứng dụng di động đóng vai trò quan trọng trong mọi lĩnh vực từ giáo dục, y tế, thương mại điện tử đến giải trí và quản lý cá nhân. Với sự gia tăng không ngừng về số lượng thiết bị di động và kết nối Internet, nhu cầu sử dụng các ứng dụng để giải quyết vấn đề hàng ngày càng trở nên phổ biến.

Thống kê cho thấy, tính đến năm 2023, có hơn 6.3 tỷ người sử dụng smartphone trên toàn thế giới và con số này dự kiến sẽ tiếp tục tăng trưởng. Ứng dụng di động không chỉ giúp người dùng thực hiện các tác vụ một cách dễ dàng và nhanh chóng mà còn mang lại cơ hội tiếp cận thị trường cho doanh nghiệp, thúc đẩy doanh thu và xây dựng thương hiệu.

Trong bối cảnh cạnh tranh khốc liệt, các xu hướng phát triển ứng dụng như tối ưu hóa trải nghiệm người dùng (UX), giao diện người dùng (UI) hấp dẫn, tích hợp trí tuệ nhân tạo (AI), và bảo mật thông tin người dùng đang là những yếu tố quyết định thành công của một ứng dụng di động. Việc phát triển ứng dụng đa nền tảng, sử dụng các framework như Flutter hay React Native cũng giúp giảm thời gian và chi phí phát triển.

### ***1.1.2 Tổng quan về các hệ điều hành thiết bị di động, các công cụ lập trình***

Các hệ điều hành di động chính hiện nay bao gồm Android, iOS, và HarmonyOS. Trong đó, Android và iOS chiếm ưu thế trên thị trường với hơn 90% thị phần toàn cầu.

* **Android**: Được phát triển bởi Google, là hệ điều hành mã nguồn mở dựa trên nền tảng Linux. Android cho phép các nhà phát triển tiếp cận một lượng lớn người dùng với khả năng tùy biến cao và hỗ trợ từ cộng đồng mạnh mẽ.
* **iOS**: Hệ điều hành của Apple được biết đến với sự bảo mật, trải nghiệm người dùng mượt mà và ổn định. iOS có tính năng độc quyền và thường được phát triển song song với hệ sinh thái phần cứng của Apple.
* **HarmonyOS**: Hệ điều hành được phát triển bởi Huawei, đang dần trở nên phổ biến tại Trung Quốc và một số quốc gia khác. HarmonyOS nổi bật với khả năng liên kết giữa nhiều loại thiết bị, từ điện thoại, máy tính bảng đến thiết bị IoT.

Các công cụ lập trình để phát triển ứng dụng di động phổ biến bao gồm Android Studio, Xcode (dành cho iOS), và các framework đa nền tảng như React Native, Flutter, và Xamarin. Android Studio và Xcode là công cụ chính thức được Google và Apple cung cấp, tích hợp nhiều tính năng mạnh mẽ hỗ trợ phát triển, kiểm thử và triển khai ứng dụng.

### ***1.1.3 Hệ điều hành Android***

Android là hệ điều hành di động được Google phát triển và phát hành đầu tiên vào năm 2008. Với hơn 70% thị phần, Android đã trở thành hệ điều hành phổ biến nhất cho thiết bị di động trên toàn thế giới nhờ vào tính linh hoạt, mã nguồn mở và khả năng tương thích với nhiều loại phần cứng khác nhau.

Android được xây dựng dựa trên nhân Linux, hỗ trợ nhiều ngôn ngữ lập trình như Java, Kotlin, C++, và gần đây là Flutter (Dart). Sự đa dạng về ngôn ngữ lập trình cho phép các nhà phát triển lựa chọn công cụ phù hợp nhất với kỹ năng và yêu cầu dự án của họ.

Một trong những ưu điểm nổi bật của Android là khả năng tùy biến giao diện và tính năng, cho phép các nhà sản xuất thiết bị và lập trình viên điều chỉnh theo nhu cầu cụ thể. Android còn nổi tiếng với Google Play Store, nơi cung cấp hàng triệu ứng dụng từ các nhà phát triển toàn cầu, tạo điều kiện thuận lợi để người dùng dễ dàng tiếp cận với các ứng dụng và dịch vụ mới.

### ***1.1.4 Công cụ lập trình Android Studio***

Android Studio là môi trường phát triển tích hợp (IDE) chính thức cho lập trình Android, được Google giới thiệu vào năm 2013. Android Studio hỗ trợ nhiều ngôn ngữ lập trình, nhưng chủ yếu là Java và Kotlin. IDE này được trang bị nhiều tính năng mạnh mẽ như trình biên dịch nhanh (Instant Run), trình gỡ lỗi (Debugger), trình giả lập (Emulator) và công cụ hỗ trợ thiết kế giao diện trực quan.

Các tính năng nổi bật của Android Studio bao gồm:

* **Trình thiết kế giao diện (Layout Editor): Cho phép lập trình viên thiết kế giao diện người dùng một cách trực quan bằng cách kéo thả các thành phần UI.**
* **Hỗ trợ Gradle: Công cụ tự động hóa và quản lý dependencies, giúp quản lý các thư viện và gói tin trở nên dễ dàng và hiệu quả hơn.**
* **Emulator: Giả lập nhiều loại thiết bị với các phiên bản Android khác nhau, giúp kiểm tra ứng dụng trên nhiều cấu hình phần cứng và phần mềm mà không cần thiết bị thật.**
* **Code Analysis và Refactoring: Hỗ trợ kiểm tra chất lượng mã nguồn và cung cấp các gợi ý để cải thiện hiệu suất, bảo mật và khả năng đọc của mã.**

Android Studio giúp lập trình viên tối ưu hóa quy trình phát triển ứng dụng Android từ khâu thiết kế đến đóng gói sản phẩm, cung cấp một môi trường làm việc toàn diện, linh hoạt và dễ sử dụng, đáp ứng tốt nhu cầu phát triển ứng dụng di động hiện đại.

## **1.2. Phát triển ứng dụng trên nền tảng Android sử dụng Java**

### ***1.2.1. Cấu trúc ứng dụng Android***

Ứng dụng Android được xây dựng dựa trên một cấu trúc module rõ ràng, gồm nhiều thành phần khác nhau để tạo ra một ứng dụng hoàn chỉnh. Mỗi ứng dụng Android đều có ít nhất một Activity (màn hình), một file Manifest (định nghĩa cấu trúc ứng dụng), và các tài nguyên liên quan như layout XML, hình ảnh, và file cấu hình.

* **AndroidManifest.xml: File này định nghĩa cấu trúc cơ bản của ứng dụng, khai báo các Activity, quyền truy cập hệ thống, và các thành phần khác. Manifest cũng xác định phiên bản ứng dụng, biểu tượng, và tên ứng dụng được hiển thị cho người dùng.**
* **Java/Kotlin Code: Chứa mã nguồn chính của ứng dụng, bao gồm các Activity, Service, và các logic xử lý khác. Đây là phần cốt lõi thực hiện các chức năng mà ứng dụng cung cấp.**
* **Resource Files: Bao gồm các file XML dùng để thiết kế giao diện (layout), chuỗi văn bản (strings), màu sắc (colors), và các tài nguyên đa phương tiện khác như hình ảnh và âm thanh.**
* **Gradle Files: Sử dụng để quản lý các dependencies, xây dựng và đóng gói ứng dụng. Gradle giúp tự động hóa quy trình xây dựng và cung cấp môi trường linh hoạt để cấu hình dự án.**

Cấu trúc này giúp ứng dụng dễ dàng mở rộng và bảo trì, cho phép các nhà phát triển làm việc hiệu quả và tối ưu hóa quy trình phát triển.

### ***1.2.2. Các thành phần phát triển ứng dụng Android***

Các thành phần cơ bản trong một ứng dụng Android bao gồm Activity, Fragment, Service, Broadcast Receiver, và Content Provider. Mỗi thành phần đóng một vai trò quan trọng trong việc xử lý dữ liệu và tương tác với người dùng.



Việc phát triển ứng dụng trên nền tảng Android bằng Java cung cấp một môi trường linh hoạt và mạnh mẽ để xây dựng các ứng dụng chất lượng cao, đáp ứng nhu cầu của người dùng và tối ưu hóa trải nghiệm. Các công cụ và thư viện hỗ trợ phong phú giúp lập trình viên dễ dàng triển khai và nâng cấp ứng dụng một cách nhanh chóng và hiệu quả.

# **CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

## **2.1 Giới thiệu bài toán**

***2.1.1 Giới thiệu về đề tài***

Đề tài "Xây dựng ứng dụng quản lý sinh viên" nhằm phát triển một ứng dụng di động giúp tối ưu hóa và hiện đại hóa quy trình quản lý thông tin sinh viên. Ứng dụng này được thiết kế dành cho các trường học, giảng viên, và cán bộ quản lý nhằm giảm bớt khối lượng công việc thủ công và cải thiện khả năng tiếp cận thông tin một cách nhanh chóng và chính xác.

**Mục tiêu:**

* **Tăng cường trải nghiệm người dùng:** Ứng dụng cung cấp giao diện thân thiện, dễ sử dụng, giúp người dùng dễ dàng quản lý và theo dõi thông tin sinh viên. Các tính năng như tìm kiếm nhanh, thống kê, và thông báo sẽ giúp người dùng tiết kiệm thời gian và công sức.
* **Tối ưu hóa quy trình quản lý:** Hệ thống tự động hóa các công việc như quản lý học phần, điểm số, và thông tin sinh viên, giúp giảm thiểu sai sót và tăng hiệu quả công việc. Thông tin được lưu trữ và truy xuất một cách có hệ thống, giúp cải thiện khả năng quản lý và báo cáo.
* **Tăng cường bảo mật:** Ứng dụng đảm bảo an toàn cho dữ liệu cá nhân của sinh viên thông qua các biện pháp bảo mật như mã hóa dữ liệu, xác thực người dùng và phân quyền truy cập.
* **Hỗ trợ chăm sóc khách hàng:** Cung cấp kênh thông tin nhanh chóng và tiện lợi để nhà trường có thể giao tiếp và hỗ trợ sinh viên cũng như phụ huynh một cách hiệu quả, giúp nâng cao chất lượng dịch vụ.

***2.1.2 Yêu cầu bài toán***

Để xây dựng ứng dụng quản lý sinh viên, các yêu cầu chính bao gồm việc thiết kế giao diện người dùng trực quan, tích hợp đầy đủ các chức năng quản lý, đảm bảo hiệu suất và tính bảo mật cao.

***2.1.2.1 Giao diện người dùng:***

Giao diện người dùng đóng vai trò quan trọng trong việc giúp người dùng tương tác dễ dàng với ứng dụng. Các chức năng chính trong giao diện bao gồm:

* **Đăng nhập:** Cho phép người dùng đăng nhập vào hệ thống bằng tài khoản và mật khẩu để truy cập các tính năng quản lý. Hỗ trợ xác thực hai yếu tố để tăng cường bảo mật.
* **Đăng ký:** Người dùng có thể đăng ký tài khoản mới với các thông tin cơ bản như tên, email, và mật khẩu. Hệ thống kiểm tra tính hợp lệ và an toàn của thông tin đầu vào trước khi lưu trữ.
* **Quản lý học phần:** Hỗ trợ tạo, chỉnh sửa, và xóa các học phần. Người dùng có thể thêm thông tin chi tiết về học phần như tên môn học, mã môn, số tín chỉ, và giáo viên phụ trách.
* **Quản lý sinh viên:** Cho phép thêm mới, chỉnh sửa, và xóa thông tin sinh viên. Ứng dụng lưu trữ các thông tin cá nhân như họ tên, mã số sinh viên, ngày sinh, lớp học, và địa chỉ liên hệ.
* **Quản lý sinh viên trong lớp học:** Hỗ trợ phân chia sinh viên vào các lớp học cụ thể, giúp giảng viên dễ dàng theo dõi và quản lý số lượng sinh viên trong từng lớp. Chức năng này bao gồm tìm kiếm và cập nhật danh sách sinh viên trong lớp.
* **Quản lý điểm của sinh viên:** Cho phép nhập và theo dõi điểm số của sinh viên cho từng học phần. Hệ thống có thể tự động tính toán điểm trung bình, xếp loại và xuất báo cáo theo yêu cầu. Người dùng có thể tìm kiếm điểm số theo từng sinh viên hoặc môn học cụ thể.

Các yêu cầu trên nhằm đảm bảo rằng ứng dụng quản lý sinh viên không chỉ thân thiện, dễ sử dụng mà còn đáp ứng đầy đủ các chức năng cần thiết trong việc quản lý và theo dõi thông tin học tập, giúp nâng cao hiệu quả quản lý và cải thiện trải nghiệm người dùng.

**2.1.2.2 Chức năng quản lý:**

Các chức năng quản lý là phần cốt lõi của ứng dụng, giúp người dùng thực hiện các tác vụ liên quan đến quản lý sinh viên và học phần một cách hiệu quả và chính xác. Các chức năng chính bao gồm:

* **Quản lý tài khoản người dùng:**
  + Cho phép quản lý tài khoản của người dùng, bao gồm tạo mới, sửa đổi thông tin, và xóa tài khoản.
  + Tính năng phân quyền người dùng theo vai trò như quản trị viên, giảng viên, và sinh viên để đảm bảo các quyền truy cập phù hợp với từng đối tượng.
* **Quản lý sinh viên:**
  + Hỗ trợ thêm mới, chỉnh sửa, xóa, và tìm kiếm sinh viên theo nhiều tiêu chí như tên, mã sinh viên, lớp học, hoặc khóa học.
  + Cho phép cập nhật thông tin chi tiết của sinh viên như liên hệ, địa chỉ, điểm số và thông tin học tập.
* **Quản lý lớp học:**
  + Tạo và quản lý các lớp học, bao gồm thông tin về tên lớp, mã lớp, học phần được giảng dạy, và danh sách sinh viên tham gia.
  + Giúp giảng viên dễ dàng phân bổ sinh viên vào các lớp học và theo dõi danh sách sinh viên của từng lớp.
* **Quản lý học phần:**
  + Chức năng tạo, sửa đổi, và xóa các học phần, cung cấp thông tin chi tiết như mã học phần, tên môn học, số tín chỉ, thời gian học và giảng viên phụ trách.
  + Hỗ trợ xem danh sách các sinh viên đã đăng ký học phần và cập nhật trạng thái đăng ký.
* **Quản lý điểm:**
  + Cho phép nhập điểm cho các sinh viên theo từng học phần, tự động tính điểm trung bình, xếp loại, và xuất báo cáo điểm chi tiết.
  + Hỗ trợ giảng viên theo dõi kết quả học tập của sinh viên và cung cấp tính năng thông báo cho sinh viên về kết quả học tập của họ.

***2.1.2.3 Các bước triển khai***

Để phát triển ứng dụng quản lý sinh viên một cách hiệu quả, quá trình triển khai cần tuân thủ các bước cụ thể nhằm đảm bảo rằng sản phẩm cuối cùng đáp ứng đúng yêu cầu và hoạt động ổn định.

**1. Phân tích yêu cầu**

* **Thu thập yêu cầu:** Giai đoạn đầu tiên là làm việc với khách hàng và các bên liên quan như giảng viên, cán bộ quản lý, và sinh viên để hiểu rõ mong đợi về chức năng của ứng dụng. Quá trình này bao gồm các cuộc họp, khảo sát, và phỏng vấn để thu thập thông tin chi tiết.
* **Phân tích và tài liệu hóa yêu cầu:** Sau khi thu thập thông tin, cần phân tích và xác định các yêu cầu chi tiết về chức năng (quản lý tài khoản, quản lý sinh viên, quản lý học phần…) và phi chức năng (bảo mật, hiệu suất, khả năng mở rộng). Các yêu cầu này sẽ được ghi chép trong tài liệu yêu cầu, giúp định hướng cho các bước thiết kế và phát triển tiếp theo.

**2. Thiết kế hệ thống**

* **Thiết kế kiến trúc tổng thể:** Lựa chọn kiến trúc hệ thống phù hợp, như mô hình MVC (Model-View-Controller) để đảm bảo tính linh hoạt và dễ bảo trì. Kiến trúc này giúp phân tách rõ ràng giữa giao diện người dùng, logic xử lý và quản lý dữ liệu, từ đó tối ưu hóa hiệu suất và khả năng phát triển mở rộng sau này.
* **Thiết kế cơ sở dữ liệu:** Thiết kế cơ sở dữ liệu nhằm đảm bảo việc lưu trữ và truy xuất dữ liệu hiệu quả. Cấu trúc cơ sở dữ liệu cần bao gồm các bảng chính như bảng sinh viên, học phần, lớp học, điểm số, và tài khoản người dùng. Thiết kế cần chú trọng đến việc tối ưu hóa các truy vấn và đảm bảo tính toàn vẹn dữ liệu.
* **Thiết kế giao diện người dùng:** Thiết kế giao diện đảm bảo tính thân thiện, dễ sử dụng và phù hợp với nhu cầu người dùng. Giao diện cần rõ ràng, dễ truy cập và có tính thẩm mỹ cao, với các thành phần được bố trí hợp lý để hỗ trợ người dùng thực hiện các thao tác một cách nhanh chóng.

**3. Phát triển hệ thống**

* **Xây dựng các module chức năng:** Phát triển từng module chức năng theo thiết kế đã được phê duyệt, bao gồm các module như quản lý sinh viên, quản lý lớp học, quản lý học phần, quản lý điểm, và quản lý tài khoản người dùng. Mỗi module sẽ được phát triển độc lập để dễ dàng kiểm thử và tích hợp.
* **Tích hợp hệ thống và kiểm thử từng phần:** Sau khi phát triển xong từng module, tiến hành tích hợp các module lại với nhau và thực hiện kiểm thử toàn diện. Quá trình kiểm thử giúp đảm bảo rằng các chức năng hoạt động đúng đắn, không có lỗi và đáp ứng các yêu cầu ban đầu. Các loại kiểm thử có thể bao gồm kiểm thử chức năng, kiểm thử giao diện, và kiểm thử hiệu suất.

***2.1.2.4 Các yêu cầu cài đặt***

**a) IDE (Môi trường phát triển tích hợp)**

* **Android Studio:** Đây là IDE chính thức cho phát triển ứng dụng Android, hỗ trợ ngôn ngữ Java và Kotlin. Android Studio cung cấp đầy đủ các công cụ cần thiết như trình giả lập, trình thiết kế giao diện, trình biên dịch, và các công cụ kiểm thử giúp phát triển ứng dụng một cách nhanh chóng và hiệu quả.

**b) Hệ quản trị cơ sở dữ liệu**

* **SQLite:** SQLite là hệ quản trị cơ sở dữ liệu nhẹ, không cần cài đặt phức tạp, và được tích hợp sẵn trong Android. Nó phù hợp với các ứng dụng di động có nhu cầu lưu trữ dữ liệu cục bộ. SQLite hỗ trợ đầy đủ các thao tác CRUD (Create, Read, Update, Delete), cho phép ứng dụng quản lý dữ liệu hiệu quả với hiệu suất cao.

## **2.2 Phân tích hệ thống**

### ***2.2.1. Sơ đồ chức năng của phần mềm***

Sơ đồ chức năng của phần mềm quản lý sinh viên là một biểu đồ mô tả các chức năng chính của hệ thống và cách chúng tương tác với nhau. Sơ đồ này giúp hình dung rõ ràng về cấu trúc tổng thể của ứng dụng, các chức năng cốt lõi và mối liên kết giữa chúng. Dưới đây là mô tả chi tiết về các chức năng chính trong sơ đồ.

**1. Quản lý tài khoản người dùng**

* **Đăng nhập/Đăng xuất:** Cho phép người dùng truy cập vào hệ thống hoặc thoát khỏi hệ thống một cách an toàn.
* **Đăng ký tài khoản:** Người dùng mới có thể tạo tài khoản với thông tin cá nhân cơ bản.
* **Phân quyền người dùng:** Phân quyền cho từng vai trò như quản trị viên, giảng viên, và sinh viên để kiểm soát quyền truy cập vào các chức năng khác nhau của ứng dụng.
* **Quản lý thông tin tài khoản:** Thay đổi mật khẩu, cập nhật thông tin cá nhân, và quản lý bảo mật tài khoản.

**2. Quản lý sinh viên**

* **Thêm sinh viên:** Nhập thông tin sinh viên mới vào hệ thống.
* **Chỉnh sửa thông tin sinh viên:** Cập nhật thông tin cá nhân, liên hệ, và học tập của sinh viên.
* **Xóa sinh viên:** Xóa bỏ thông tin sinh viên khỏi hệ thống.
* **Tìm kiếm sinh viên:** Tìm kiếm nhanh thông tin sinh viên theo tên, mã số, lớp học hoặc khóa học.

**3. Quản lý lớp học**

* **Tạo lớp học mới:** Khởi tạo lớp học mới với các thông tin cần thiết như tên lớp, mã lớp, và giảng viên phụ trách.
* **Cập nhật thông tin lớp học:** Sửa đổi thông tin về lớp học và danh sách sinh viên tham gia.
* **Xóa lớp học:** Xóa lớp học khi không còn sử dụng hoặc cần thay đổi cấu trúc.

**4. Quản lý học phần**

* **Thêm học phần:** Tạo mới các học phần với thông tin chi tiết như mã học phần, tên môn học, và số tín chỉ.
* **Chỉnh sửa học phần:** Cập nhật thông tin và trạng thái học phần khi cần thiết.
* **Xóa học phần:** Loại bỏ các học phần đã hoàn thành hoặc không còn cần thiết.
* **Quản lý đăng ký học phần:** Theo dõi và quản lý việc sinh viên đăng ký các học phần.

**5. Quản lý điểm số**

* **Nhập điểm số:** Nhập điểm cho các sinh viên theo từng học phần cụ thể.
* **Cập nhật điểm số:** Sửa đổi điểm số nếu có sai sót hoặc cần điều chỉnh.
* **Xem báo cáo điểm:** Xuất báo cáo điểm chi tiết cho từng sinh viên và học phần.
* **Thông báo kết quả:** Gửi thông báo kết quả học tập đến sinh viên.

### ***2.2.2. Danh sách các tác nhân (Actor)***

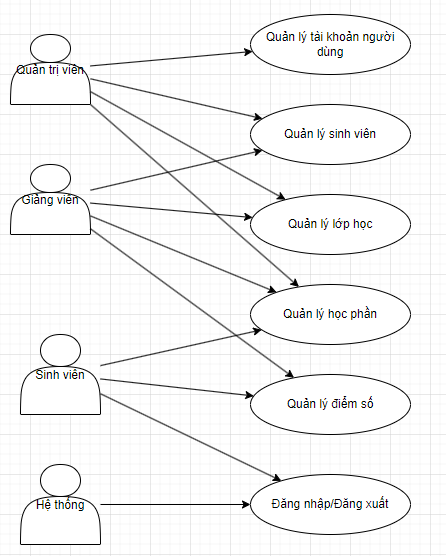
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **KÝ HIỆU** | **TÊN TÁC NHÂN** | **MÔ TẢ** | **PHÂN LOẠI** |
| 1 | Admin | Quản trị viên | Người quản lý hệ thống, có quyền cao nhất, thực hiện các chức năng quản lý tài khoản, sinh viên, lớp học, và học phần. | Cao |
| 2 | Lecturer | Giảng viên | Thực hiện các chức năng quản lý lớp học, quản lý học phần, nhập và cập nhật điểm cho sinh viên. | Trung bình |
| 3 | Student | Sinh viên | Sinh viên đăng nhập vào hệ thống để xem thông tin cá nhân, đăng ký học phần và theo dõi điểm số. | Thấp |
| 4 | System | Hệ thống | Thành phần xử lý tự động các tác vụ như kiểm tra dữ liệu, lưu trữ, và cung cấp phản hồi cho người dùng. | Cao |
| 5 | Guest | Khách (Guest) | Người dùng chưa đăng nhập, có quyền hạn chế, chỉ có thể truy cập thông tin công khai và đăng ký tài khoản mới. | Thấp |

### ***2.2.3. Xác định các ca sử dụng (use-case)***

|  |  |
| --- | --- |
| **Tác nhân** | **Ca sử dụng** |
| Quản trị viên | - Quản lý tài khoản người dùng: tạo, chỉnh sửa, xóa tài khoản và phân quyền người dùng. - Quản lý sinh viên: thêm mới, chỉnh sửa, xóa, và tìm kiếm thông tin sinh viên.  - Quản lý lớp học: tạo, cập nhật, xóa lớp học và quản lý danh sách sinh viên trong lớp.  - Quản lý học phần: tạo, sửa đổi, xóa học phần và quản lý đăng ký học phần của sinh viên. |
| Giảng viên | - Quản lý lớp học: xem và chỉnh sửa thông tin lớp học và danh sách sinh viên.  - Quản lý điểm: nhập điểm, cập nhật điểm và thông báo kết quả học tập cho sinh viên. |
| Sinh viên | - Đăng nhập/Đăng xuất: truy cập vào hệ thống hoặc thoát khỏi hệ thống.  - Đăng ký học phần: tìm kiếm và đăng ký các học phần có sẵn.  - Xem điểm: xem kết quả học tập cá nhân và các thông tin liên quan đến điểm số. |
| Hệ thống | - Xử lý đăng nhập/đăng ký: xác thực thông tin người dùng và quản lý phiên làm việc.  - Lưu trữ và truy xuất dữ liệu: quản lý lưu trữ dữ liệu người dùng, sinh viên, lớp học, và điểm. |

## **2.3. Thiết kế hệ thống**

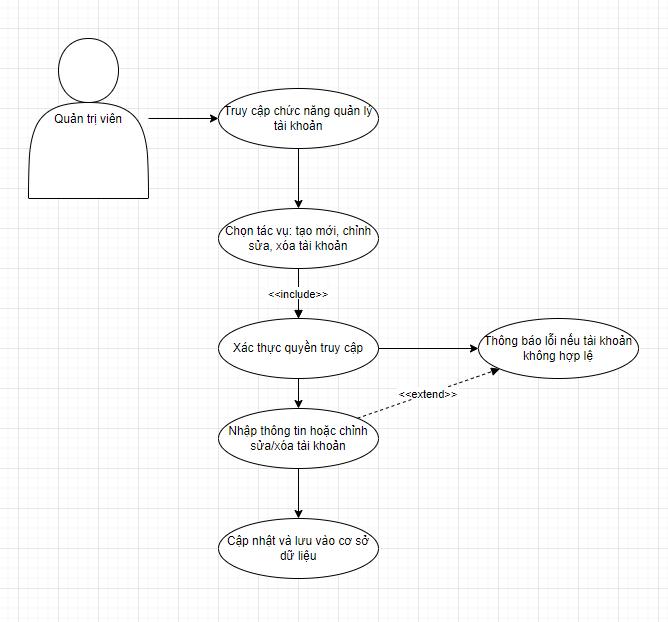
### ***2.3.1. Đặc tả use-case tổng quát***



Hình 1. Biểu đồ Use case tổng quát

##### ***2.3.2. Use-Case 1: Quản lý tài khoản người dùng***

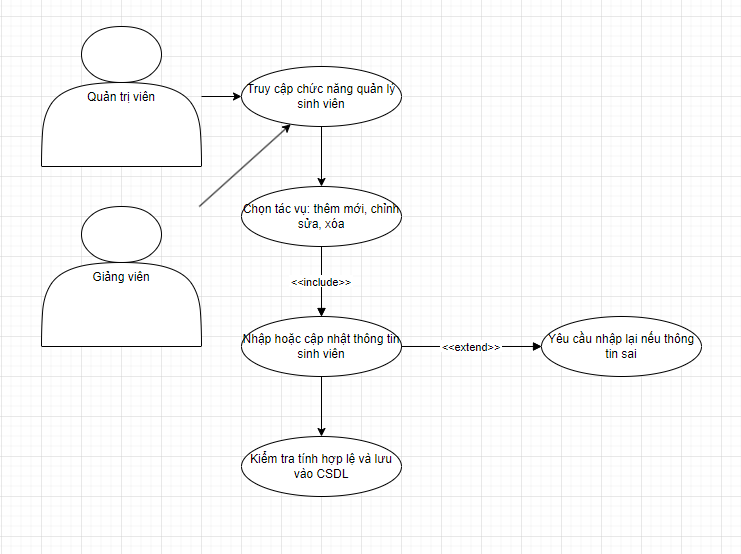
* **Tác nhân:** Quản trị viên
* **Mô tả:** Cho phép quản trị viên tạo, chỉnh sửa, xóa tài khoản người dùng và phân quyền cho các vai trò khác nhau trong hệ thống.
* **Luồng sự kiện chính:**
  + Quản trị viên truy cập vào chức năng quản lý tài khoản.
  + Quản trị viên chọn tác vụ: tạo mới, chỉnh sửa, hoặc xóa tài khoản.
  + Hệ thống xác thực quyền truy cập của quản trị viên.
  + Quản trị viên nhập thông tin tài khoản hoặc thực hiện chỉnh sửa/xóa.
  + Hệ thống cập nhật thông tin và lưu vào cơ sở dữ liệu.
* **Luồng phụ:**
  + Nếu tài khoản không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu nhập lại.
* **Điều kiện tiền đề:** Quản trị viên đã đăng nhập thành công.
* **Kết quả:** Tài khoản người dùng được quản lý thành công.



Hình 2. Biểu đồ Use case quản lý tài khoản người dùng

##### ***2.3.3. Use-Case 2: Quản lý sinh viên***

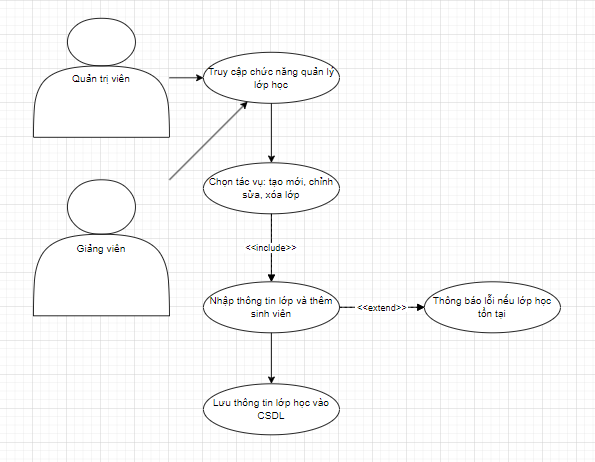
* **Tác nhân: Quản trị viên, Giảng viên**
* **Mô tả: Hỗ trợ thêm mới, chỉnh sửa, xóa, và tìm kiếm sinh viên theo nhiều tiêu chí.**
* **Luồng sự kiện chính:**
  + Tác nhân truy cập vào chức năng quản lý sinh viên.
  + Chọn tác vụ: thêm mới, chỉnh sửa, hoặc xóa sinh viên.
  + Nhập hoặc cập nhật thông tin sinh viên.
  + Hệ thống kiểm tra tính hợp lệ của thông tin và lưu trữ vào cơ sở dữ liệu.
* **Luồng phụ:**
  + Nếu dữ liệu nhập sai định dạng, hệ thống yêu cầu nhập lại thông tin.
  + Điều kiện tiền đề: Tác nhân đã đăng nhập với quyền phù hợp.
  + Kết quả: Thông tin sinh viên được cập nhật trong hệ thống.



Hình 3. Biểu đồ Use case quản lý sinh viên

##### ***2.3.4. Use-Case 3: Quản lý lớp học***

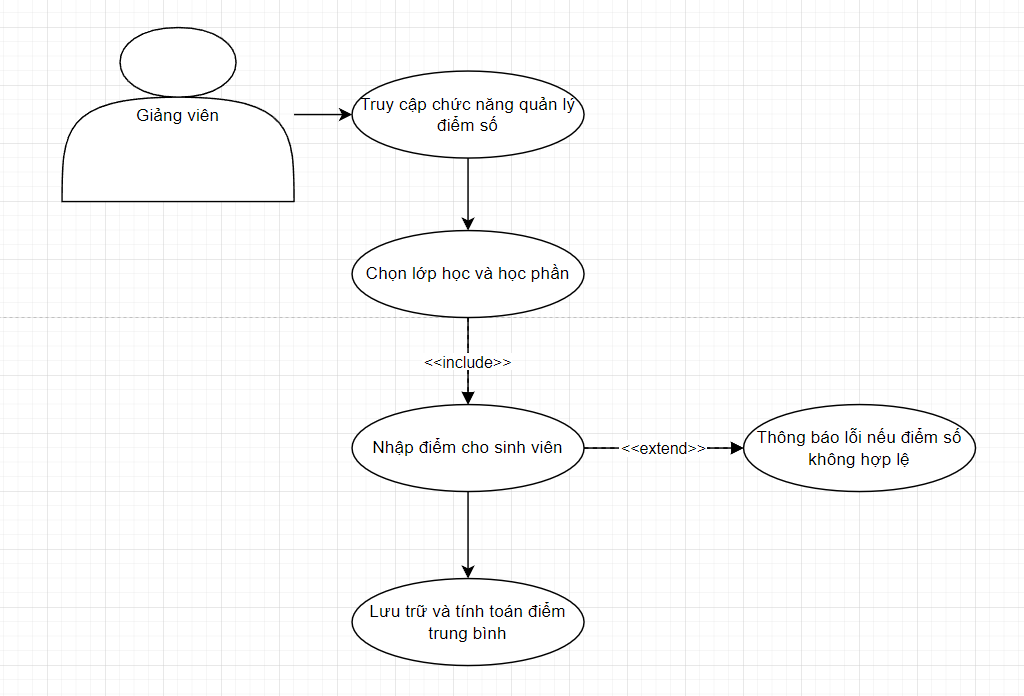
* **Tác nhân: Quản trị viên, Giảng viên**
* **Mô tả: Cho phép tạo mới, chỉnh sửa, và quản lý danh sách sinh viên trong lớp học.**
* **Luồng sự kiện chính:**
  + Tác nhân truy cập vào chức năng quản lý lớp học.
  + Tác nhân chọn tác vụ: tạo mới, chỉnh sửa, hoặc xóa lớp học.
  + Nhập thông tin chi tiết về lớp học, thêm sinh viên vào lớp.
  + Hệ thống lưu lại thông tin lớp học và danh sách sinh viên.
* **Luồng phụ:**
  + Nếu lớp học đã tồn tại hoặc không có sinh viên nào được thêm, hệ thống thông báo lỗi.
  + Điều kiện tiền đề: Tác nhân phải có quyền quản lý lớp học.
  + Kết quả: Lớp học và danh sách sinh viên được quản lý thành công.



Hình 4. Biểu đồ Use case quản lý lớp học

##### ***2.3.4. Use-Case 4: Quản lý điểm số***

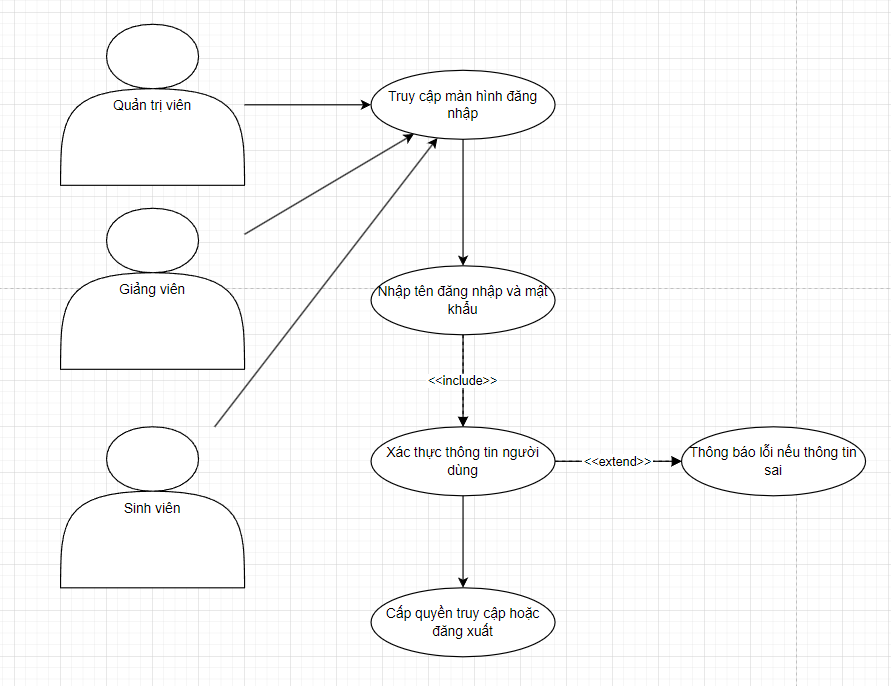
* **Tác nhân: Giảng viên**
* **Mô tả: Nhập, cập nhật và quản lý điểm số của sinh viên cho từng học phần.**
* **Luồng sự kiện chính:**
  + Giảng viên truy cập vào chức năng quản lý điểm số.
  + Chọn lớp học và học phần để nhập hoặc cập nhật điểm số.
  + Nhập điểm cho từng sinh viên.
  + Hệ thống lưu trữ điểm số và tính toán điểm trung bình.
* **Luồng phụ:**
  + Hệ thống thông báo lỗi nếu điểm số nhập không hợp lệ.
  + Điều kiện tiền đề: Giảng viên đã đăng nhập và có quyền quản lý điểm.
  + Kết quả: Điểm số sinh viên được cập nhật chính xác trong hệ thống.



Hình 5. Biểu đồ Use case quản lý điểm số

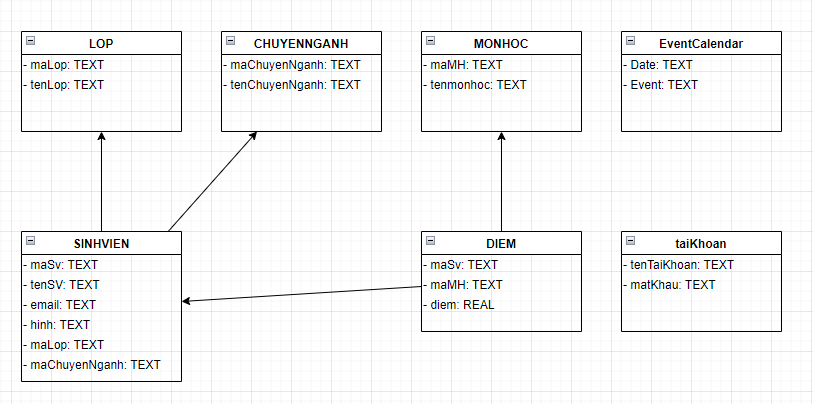
##### ***2.3.5. Use-Case 4: Đăng nhập/Đăng xuất***

* **Tác nhân: Quản trị viên, Giảng viên, Sinh viên**
* **Mô tả: Xác thực người dùng để truy cập hoặc thoát khỏi hệ thống.**
* **Luồng sự kiện chính:**
  + Người dùng truy cập vào màn hình đăng nhập.
  + Nhập tên đăng nhập và mật khẩu.
  + Hệ thống xác thực thông tin và cấp quyền truy cập.
  + Người dùng có thể đăng xuất khi hoàn thành công việc.
* **Luồng phụ:**
  + Hiển thị thông báo lỗi nếu thông tin đăng nhập sai.
  + Điều kiện tiền đề: Người dùng có tài khoản hợp lệ.
  + Kết quả: Người dùng truy cập hoặc thoát khỏi hệ thống thành công.



Hình 6. Biểu đồ Use case Đăng nhập/Đăng xuất

## **2.4. Biểu đồ lớp**

****

Hình 7. Biểu đồ lớp

##### ***2.4.1. LOP:***

* **Thuộc tính**:
  + maLop: Mã lớp (kiểu TEXT, khóa chính).
  + tenLop: Tên lớp.
* **Chức năng**: Lớp này đại diện cho thông tin về các lớp học trong hệ thống.

##### ***2.4.2. CHUYENNGANH:***

* **Thuộc tính:**
  + maChuyenNganh: Mã chuyên ngành (kiểu TEXT, khóa chính).
  + tenChuyenNganh: Tên chuyên ngành.
* **Chức năng:** Quản lý các chuyên ngành mà sinh viên có thể theo học.

##### ***2.4.3. MONHOC:***

* **Thuộc tính:**
  + maMH: Mã môn học (kiểu TEXT, khóa chính).
  + tenmonhoc: Tên môn học.
* **Chức năng:** Chứa thông tin về các môn học mà sinh viên có thể đăng ký.

##### ***2.4.4. EventCalendar:***

* **Thuộc tính:**
  + Date: Ngày sự kiện.
  + Event: Mô tả sự kiện.
* **Chức năng:** Quản lý các sự kiện lịch như lịch học, lịch thi.

##### ***2.4.5. SINHVIEN:***

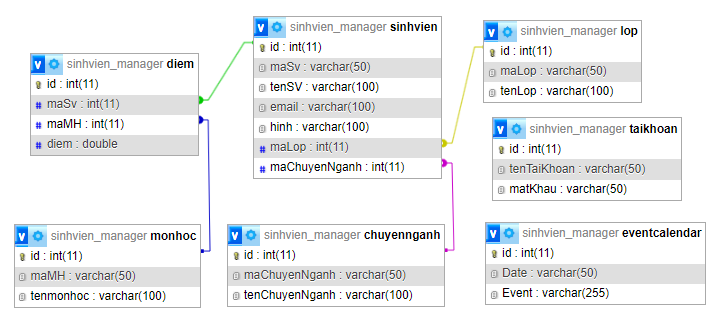
* **Thuộc tính:**
  + maSv: Mã sinh viên (kiểu TEXT, khóa chính).
  + tenSV: Tên sinh viên.
  + email: Email sinh viên.
  + hinh: Hình ảnh sinh viên.
  + maLop: Mã lớp (liên kết đến lớp LOP).
  + maChuyenNganh: Mã chuyên ngành (liên kết đến lớp CHUYENNGANH).
* **Chức năng:** Quản lý thông tin chi tiết của sinh viên bao gồm thông tin cá nhân và lớp học.

##### ***2.4.6. taiKhoan:***

* **Thuộc tính:**
  + tenTaiKhoan: Tên tài khoản (kiểu TEXT, khóa chính).
  + matKhau: Mật khẩu.
* **Chức năng:** Quản lý thông tin đăng nhập của người dùng hệ thống.

## **2.5. Thiết kế cơ sở dữ liệu**

Sơ đồ tổng quát



Hình 8. Biểu đồ tổng quát

Bảng: LOP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Kích thước | Ý nghĩa |
| 1 | id | INT | 11 | Khóa chính, tự động tăng |
| 2 | maLop | VARCHAR | 50 | Mã lớp |
| 3 | tenLop | VARCHAR | 100 | Tên lớp học |

Bảng: CHUYENNGANH

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Kích thước | Ý nghĩa |
| 1 | id | INT | 11 | Khóa chính, tự động tăng |
| 2 | maChuyenNganh | VARCHAR | 50 | Mã chuyên ngành |
| 3 | tenChuyenNganh | VARCHAR | 100 | Tên chuyên ngành |

Bảng: MONHOC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Kích thước | Ý nghĩa |
| 1 | id | INT | 11 | Khóa chính, tự động tăng |
| 2 | maMH | VARCHAR | 50 | Mã môn học |
| 3 | tenmonhoc | VARCHAR | 100 | Tên môn học |

Bảng: EventCalendar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Kích thước | Ý nghĩa |
| 1 | id | INT | 11 | Khóa chính, tự động tăng |
| 2 | Date | VARCHAR | 50 | Ngày của sự kiện |
| 3 | Event | VARCHAR | 255 | Mô tả sự kiện |

Bảng: SINHVIEN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Kích thước | Ý nghĩa |
| 1 | id | INT | 11 | Khóa chính, tự động tăng |
| 2 | maSv | VARCHAR | 50 | Mã sinh viên |
| 3 | tenSV | VARCHAR | 100 | Tên sinh viên |
| 4 | email | VARCHAR | 100 | Email của sinh viên |
| 5 | hinh | VARCHAR | 100 | Hình ảnh của sinh viên |
| 6 | maLop | INT | 11 | Mã lớp, liên kết đến bảng LOP |
| 7 | maChuyenNganh | INT | 11 | Mã chuyên ngành, liên kết đến bảng CHUYENNGANH |

Bảng: taiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Kích thước | Ý nghĩa |
| 1 | id | INT | 11 | Khóa chính, tự động tăng |
| 2 | tenTaiKhoan | VARCHAR | 50 | Tên tài khoản |
| 3 | matKhau | VARCHAR | 50 | Mật khẩu tài khoản |

Bảng: DIEM

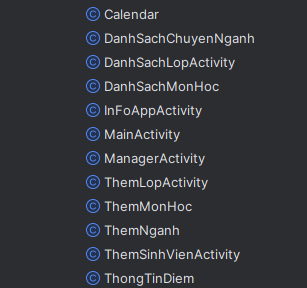
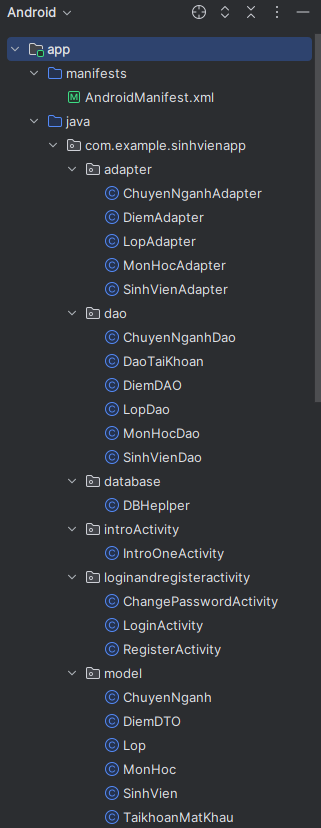
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên Thuộc tính | Kiểu dữ liệu | Kích thước | Ý nghĩa |
| 1 | id | INT | 11 | Khóa chính, tự động tăng |
| 2 | maSv | INT | 11 | Mã sinh viên, liên kết đến bảng SINHVIEN |
| 3 | maMH | INT | 11 | Mã môn học, liên kết đến bảng MONHOC |
| 4 | diem | REAL | - | Điểm của sinh viên trong môn học |

# 

# **CHƯƠNG 3. CÀI ĐẶT CHƯƠNG TRÌNH**

## **3.1 Tổ chức chương trình**

Hệ thống quản lý sinh viên được xây dựng theo mô hình MVC (Model-View-Controller), giúp tách biệt rõ ràng giữa phần xử lý dữ liệu (Model), giao diện người dùng (View), và điều khiển luồng xử lý (Controller). Việc áp dụng mô hình MVC trong phát triển ứng dụng mang lại nhiều lợi ích, giúp hệ thống dễ dàng bảo trì, nâng cấp, và mở rộng trong tương lai. Dưới đây là cách tổ chức cụ thể của chương trình theo mô hình MVC.



Hình 9. Tổ chức chương trình

**3.3. Giao diện chính của chương trình**

# **CHƯƠNG 4. KẾT LUẬN**

## **4.1. Kết quả đạt được**

Trong quá trình xây dựng và phát triển hệ thống quản lý sinh viên, nhóm đã hoàn thành các nhiệm vụ chính và đạt được những kết quả đáng kể như sau:

### ***4.1.1. Hoàn thiện hệ thống quản lý sinh viên***

* Hệ thống được xây dựng theo mô hình MVC, đảm bảo tính rõ ràng và phân tách các thành phần Model, View, và Controller, giúp tối ưu hóa quy trình phát triển, bảo trì và mở rộng trong tương lai.
* Các chức năng chính như quản lý sinh viên, lớp học, chuyên ngành, môn học, và điểm số đã được triển khai đầy đủ, đáp ứng yêu cầu đặt ra của đề tài.
* Giao diện người dùng được thiết kế thân thiện, dễ sử dụng, giúp người dùng dễ dàng thao tác và quản lý thông tin.

### ***4.1.2. Quản lý cơ sở dữ liệu hiệu quả***

* Hệ thống cơ sở dữ liệu được thiết kế với các bảng và quan hệ hợp lý, tối ưu hóa các truy vấn và đảm bảo tính toàn vẹn dữ liệu.
* Các thao tác thêm mới, chỉnh sửa, xóa, và tìm kiếm thông tin được thực hiện nhanh chóng và chính xác.

### ***4.1.3. Đảm bảo bảo mật thông tin***

* Hệ thống quản lý tài khoản với các quyền hạn khác nhau (quản trị viên, giảng viên, sinh viên) được triển khai chặt chẽ, giúp bảo vệ dữ liệu người dùng và ngăn chặn truy cập trái phép.
* Chức năng đăng nhập, đăng xuất và xác thực thông tin người dùng hoạt động ổn định, đảm bảo an toàn cho dữ liệu cá nhân.

### ***4.1.4. Khả năng mở rộng và tích hợp cao***

* Cấu trúc chương trình cho phép dễ dàng tích hợp thêm các chức năng mới, nâng cấp các module hiện có mà không ảnh hưởng đến toàn bộ hệ thống.
* Mô hình MVC giúp dễ dàng điều chỉnh và cải thiện giao diện người dùng mà không cần thay đổi logic xử lý.

## **4.2. Hướng phát triển**

Mặc dù đã đạt được nhiều kết quả đáng khích lệ, hệ thống vẫn còn nhiều tiềm năng để phát triển và hoàn thiện hơn nữa. Một số hướng phát triển trong tương lai bao gồm:

### ***4.2.1. Nâng cấp và bổ sung chức năng mới***

* Tích hợp chức năng thông báo và nhắc nhở qua email hoặc ứng dụng di động để giúp sinh viên và giảng viên quản lý lịch học, lịch thi hiệu quả hơn.
* Phát triển module quản lý học phí và tài chính để cung cấp thông tin chi tiết về các khoản thu chi, giúp sinh viên dễ dàng theo dõi tình hình học phí của mình.

### ***4.2.2. Tăng cường bảo mật hệ thống***

* Áp dụng các kỹ thuật mã hóa dữ liệu và xác thực hai yếu tố để bảo vệ thông tin người dùng tốt hơn.
* Cải thiện quy trình xác thực và quản lý phiên làm việc để tăng cường bảo mật khi đăng nhập từ nhiều thiết bị khác nhau.

### ***4.2.3. Cải tiến hiệu suất và tối ưu hóa cơ sở dữ liệu***

* Tối ưu hóa các truy vấn SQL để giảm thời gian xử lý, đặc biệt là khi hệ thống có quy mô dữ liệu lớn hơn.
* Xây dựng các chỉ số hiệu năng (performance indexes) để tăng tốc các thao tác truy vấn và tìm kiếm dữ liệu.

### ***4.2.4. Phát triển ứng dụng di động***

* Xây dựng ứng dụng di động tích hợp với hệ thống quản lý để sinh viên và giảng viên có thể dễ dàng truy cập thông tin từ bất kỳ đâu.
* Ứng dụng sẽ hỗ trợ các chức năng như xem điểm, đăng ký môn học, quản lý lịch học ngay trên thiết bị di động.

### ***4.2.5. Phân tích và báo cáo dữ liệu***

* Phát triển các module phân tích dữ liệu và báo cáo để hỗ trợ công tác quản lý của nhà trường, giảng viên và sinh viên.
* Cung cấp các báo cáo về kết quả học tập, điểm số, tỷ lệ đỗ/trượt để đánh giá hiệu quả đào tạo.

Những hướng phát triển trên sẽ giúp hệ thống quản lý sinh viên ngày càng hoàn thiện hơn, đáp ứng nhu cầu thực tế và góp phần nâng cao chất lượng quản lý giáo dục.

# **TÀI LIỆU THAM KHẢO**

[Website]: Oracle Java Documentation, <https://docs.oracle.com/javase/>.

[Website]: Android Developer Documentation, <https://developer.android.com/>.

[Website]: Stack Overflow, Các giải pháp và thảo luận về lập trình, <https://stackoverflow.com/>.

# **PHỤ LỤC**

*BẢNG PHÂN CÔNG NHIỆM VỤ VÀ ĐÁNH GIÁ KẾT QUẢ CÁC THÀNH VIÊN TRONG NHÓM*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Họ và tên** | **Vai trò** | **Nhiệm vụ** | **Đánh giá**  **(*tổng điểm thành viên 100%)*** |
| 1 | **Nguyễn Văn A** | Nhóm trưởng | - Chịu trách nhiệm phân công, đảm bảo tiến độ, chất lượng  - Thực hiện (code) chức năng….  - … | *40%* |
| 2 | **Nguyễn Văn B** | Thành viên chính | - Thực hiện (code) chức năng…  - | *35%* |
| 3 | **Bùi Thị C** | Thành viên | - Hỗ trợ tổng hợp báo cáo  - Hỗ trợ test | *15%* |
| 4 | **Trần Văn D** | Thành viên | - Hỗ trợ tổng hợp báo cáo  - Hỗ trợ test | *10%* |
| 5 | … |  |  |  |