**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------------------------------------**

**BÁO CÁO HỌC PHẦN**

**LẬP TRÌNH ỨNG DỤNG ĐƠN NỀN TẢNG**

**XÂY DỰNG SỔ TAY SINH VIÊN TRƯỜNG ĐHCNHN**

**GVHD: ThS. Vũ Thị Dương**

**Nhóm: 06**

**Sinh viên:**

1. Nguyễn Tuấn Kiệt
2. Nguyễn Văn Mạnh

**Lớp: 20222IT6076001 Khoá: 15**

**Hà Nội, 2023**

Mục lục

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT 1](#_1fob9te)

[PHẦN 1. MỞ ĐẦU 2](#_3znysh7)

[1.1.](#_2et92p0) Lý do chọn đề tài 2

[1.2.](#_tyjcwt) Tổng quan về Android 2

[1.2.1.](#_3dy6vkm) Hệ điều hành Android là gì? 2

[1.2.2.](#_1t3h5sf) Ngôn ngữ dùng để lập trình Android 3

[1.3.](#_4d34og8) Kiến trúc của Android 4

[1.3.1.](#_2s8eyo1) Linux Kernel 4

[1.3.2.](#_17dp8vu) Các thư viện 4

[1.3.3.](#_3rdcrjn) Các thư viện Android 4

[1.3.4.](#_26in1rg) Android Runtime 5

[1.3.5.](#_lnxbz9) Application Framework 5

[1.3.6.](#_35nkun2) Application 6

[1.4.](#_1ksv4uv) Công cụ Android Studio: 6

[1.5.](#_44sinio) Ngôn ngữ java 7

[1.5.1.](#_2jxsxqh) Các tính năng của Java 8

[1.5.2.](#_z337ya) Máy ảo Java (JVM – Java Virtual Machine) 9

[1.6.](#_3j2qqm3) SQLite 9

[1.6.1.](#_1y810tw) Tính năng của SQLite 10

[1.6.2.](#_4i7ojhp) Ứng dụng chủ yêu của SQLite 11

[1.6.3.](#_2xcytpi) Nhược điểm của SQLite 11

[PHẦN 2: KẾT QUẢ NGHIÊN CỨU 12](#_1ci93xb)

[2.1.](#_3whwml4) Tìm hiểu yêu cầu bài toán 12

[2.2.](#_2bn6wsx) Xây dựng bài toán 12

[2.2.1.](#_qsh70q) Phân tích yêu cầu người dùng 12

[2.2.2.](#_3as4poj) Mô tả biểu đồ lớp dữ liệu 14

[2.3.](#_1pxezwc) Thiết kế hệ thống 15

[2.3.1.](#_49x2ik5) Mô tả giao diện 16

[2.3.2.](#_2p2csry) Thiết kế dữ liệu 31

[2.4.](#_147n2zr) Thực hiện bài toán 32

[2.4.1.](#_3o7alnk) Nguyễn Tuấn Kiệt – Xem kết quả môn học 32

[2.4.2.](#_23ckvvd) Nguyễn Văn Mạnh – Xem thời khóa biểu 41

[PHẦN 3: KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM 55](#_ihv636)

[3.1. Nội dung đã thực hiện 55](#_32hioqz)

[3.2. Kết quả đạt được 55](#_1hmsyys)

[3.3. Hạn chế của đề tài 55](#_41mghml)

[3.4. Hướng phát triển 55](#_2grqrue)

[Tài liệu tham khảo 56](#_vx1227)

**DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT**

| STT | KÝ HIỆU TẮT | NỘI DUNG ĐẦY ĐỦ |
| --- | --- | --- |
| 1 | UC | Use case |
| 2 | JVM | Java Virtual Machine |
| 3 | CSDL | Cơ sở dữ liệu |

**PHẦN 1. MỞ ĐẦU**

* 1. **Lý do chọn đề tài**

Xuất phát từ nhu cầu quản lí thời gian và tài liệu học tập của sinh viên trường Đại học Công Nghiệp Hà Nội, nhóm 6 chúng em quyết định xây dựng một ứng dụng Sổ tay sinh viên cho sinh viên. Sổ tay sinh viên có nhiều lợi ích đối với sinh viên như sau:

* Giúp sinh viên nắm bắt các thông tin về chương trình học tập của ngành nghề mà mình theo học. Việc này giúp sinh viên có thể chuẩn bị kỹ càng trước khi tiếp nhận lượng kiến thức này.
* Cho phép sinh viên theo dõi kết quả học tập của mình để có thể đưa ra nhưng chiến lược cho tương lai để cải thiện thành tích học tập.
* Học kết hợp giúp cho sinh viên có thể giúp cho sinh viên nâng cao kiến thức của mình một cách chủ động về mặt thời gian.
* Kiểm soát tài chính hiệu quả với các mục thu rõ ràng.
* Tạo ghi chú để tự nhắc nhở bản thân.
* Là một công cụ đơn giản nhưng hiệu quả để giúp sinh viên tự quản lý bản thân và tự định hướng cho sự nghiệp của mình.

Để thực hiện đề tài, nhóm 6 chúng em đã vận dụng những kiến thức liên quan đến việc sử lí giao diện, xử lý sự kiện, Intent, ListView, SQLite,… đã được học ở trong chương trình đào tạo. Bên cạnh đó là một số kiến thức liên quan đến Fragment, TabLayout,… đã được nhóm 6 tìm hiểu thêm để bài báo cáo được chỉnh chu nhất.

* 1. **Tổng quan về Android**
     1. **Hệ điều hành Android là gì?**

Android là một hệ điều hành có dạng mã nguồn mở, nó hoạt động dựa trên nền tảng Linux và được thiết kế dành riêng cho những thiết bị di động cảm ứng hoặc máy tính bảng. Trước đây, hệ điều hành này được phát triển bởi tổng công ty Android và được tài trợ bởi Google. Cho đến năm 2005 thì Google đã mua lại hệ điều hành này và cho ra mắt người dùng vào năm 2007. Android này sở hữu mã nguồn mở nên lập trình viên có thể dễ dàng điều chỉnh và phân phối nó một cách tự do. Đây chính là một trong những yếu tố đã giúp cho Android trở thành nền tảng xây dựng điện thoại thông minh phát triển nhất trên thế giới.

Hiện tại, Android đã chiếm 65% so với thị phần điện thoại thông minh trên toàn thế giới vào quý 3 năm 2012. Theo điều tra thì đã có khoảng 500 triệu thiết bị được kích hoạt và có đến 1.3 triệu lượt được hoạt mỗi ngày. Vào tháng 10/2020 thì android đã có hơn 700.000 ứng dụng và số lượng tải từ Google Play ước tính lên khoảng 25 tỷ lượt. Mặc dù có sự ra đời của iOS của Apple thì khiến Android có phần nào ảnh hưởng. Tuy nhiên, Android vẫn đứng ở vị trí đầu tiên trong thị phần thế giới.

* + 1. **Ngôn ngữ dùng để lập trình Android**

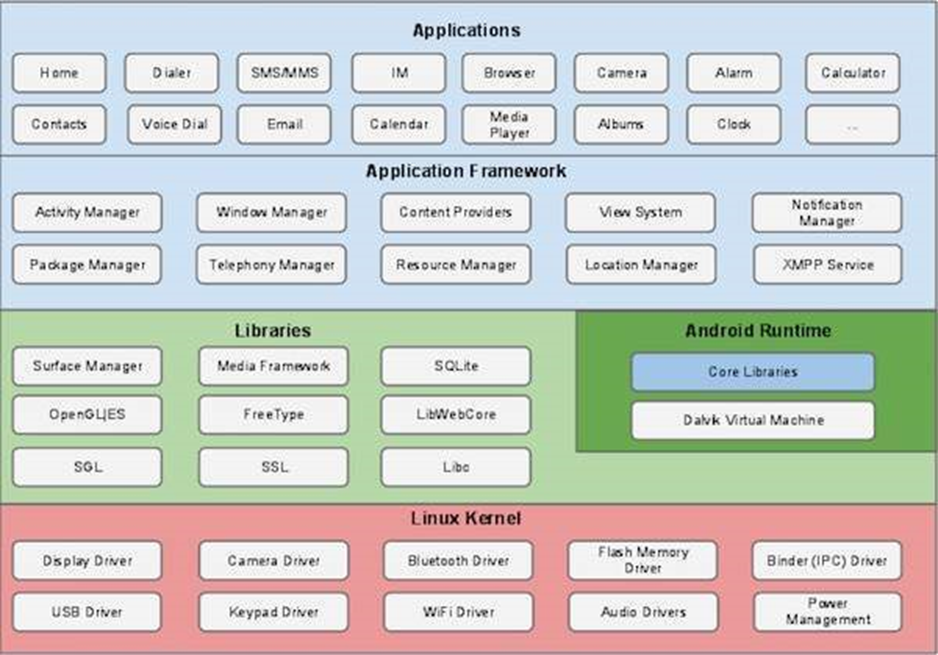
Hiện nay, hệ điều hành Android sử dụng một số loại ngôn ngữ lập trình khác nhau như: Java, Kotlin, Flutter, C, C++, CSS, Python, Lua, XML,... Đây chính là một trong những điểm cộng giúp lập trình viên dễ làm việc hơn với Android. Đặc biệt là các fresher có thể tiếp cận dễ dàng hơn tới môi trường của hệ điều hành Android.

Trong số ngôn ngữ trên thì Java được coi là ngôn ngữ lập trình chính thức của Android và đây là ngôn ngữ mà lập trình viên Android cần phải tiếp xúc thường xuyên nhất.

Java là một trong những ngôn ngữ lập trình chính thức được sử dụng chủ yếu trong hệ điều hành Android. Java đã được thiết kế nhằm tương thức với đa số môi trường phát triển nên nó thường linh hoạt hơn so với các ngôn ngữ lập trình C/C++ khác. Bên cạnh đó thì Java có hiệu suất cao và có trình giải phóng bộ nhớ đến các đối tượng không được sử dụng đến. Hiện nay, Java còn được viết nâng cao để có thể viết ra được những chương trình thực thi từ các vùng tác vụ cùng một lúc nhờ tính năng đa luồng. Ngoài ra, ngôn ngữ lập trình Java còn hỗ trợ bảo mật tốt nhờ các thuật toán dạng mã hóa như: public key hoặc mã one way hashing…

* 1. **Kiến trúc của Android**

Hệ điều hành Android là 1 ngăn xếp các thành phần phần mềm, được chia thành 5 phần và 4 lớp chính như trong hình bên dưới.



* + 1. **Linux Kernel**

Dưới cùng là lớp Linux - Linux 3.6 cùng với khoảng 115 bản vá. Lớp này cung cấp 1 cấp độ trừu tượng giữa phần cứng của thiết bị và các thành trình điều khiển phần cứng thiết yếu như máy ảnh, bàn phím, màn hình hiển thị... Đồng thời, hạt nhân (kernel) còn xử lý tất cả các thứ mà Linux có thể làm tốt như mạng kết nối và 1 chuỗi các trình điều khiển thiết bị, giúp cho giao tiếp với các thiết bị ngoại vi dễ dàng hơn.

* + 1. **Các thư viện**

Ở trên lớp nhân Linux là tập các thư viện bao gồm WebKit - trình duyệt Web mã nguồn mở, được biết đến như thư viện libc, cơ sở dữ liệu SQLite - hữu dụng cho việc lưu trữ và chia sẻ dữ liệu ứng dụng, các thư viênj chơi và ghi âm audio, video, hay các thư viện SSL chiụ trách nhiệm bảo mật Internet...

* + 1. **Các thư viện Android**

Đây là các thư viện dựa trên Java phục vụ cho việc phát triển Android. Ví dụ của các thư viện này bao gồm các thư viện ứng dụng dùng để xây dựng giao diện người dùng, vẽ đồ họa hay truy cập cơ sở dữ liệu. 1 số thư viện chính của Android:

* android.app - Cung cấp quyền truy cập vào ứng dụng và là nền tảng của tất cả ứng dụng Android.
* android.content - Cung cấp quyền truy cập nội dung (content), truyền tải thông điệp giữa các ứng dụng hay các thành phần của ứng dụng.
* android.database - Đựoc sử dụng để truy cập dữ liệu của content provider và cơ sở dữ liệu SQLite
* android.opengl - giao diện các phuơng thức Java để sử dụng OpenGL ES
* android.os - Cung cấp các ứng dụng với quyền truy cập vào các dịch vụ của hệ điều hành bao gồm thông điệp, các dịch vụ hệ thống và các giao tiếp nội tại
* android.text - Đựoc sử dụng để hiển thị và điều chỉnh chữ trên màn hình thiết bị
* android.view - Các thành phần cơ bản trong việc xây dựng giao diện người dùng của ứng dụng.
* android.widget - Tập các thành phần giao diện người dùng đã được xây dựng sẵn như các nút, các nhãn (label), list view,....
* android.webkit - Tập các lớp cho phép xây dựng khả năng duyệt web.
  + 1. **Android Runtime**

Đây là phần thứ 3 của kiến trúc và nằm ở lớp thứ 2 từ dưới lên. Phần này cung cấp 1 bộ phận quan trọng là Dalvik Vỉtual Machine - là 1 loại Java Virtual Machine được thiết kế đặc biệt để tối ưu cho Android.

Dalvik VM sử dụng các đặc trưng của nhân Linux như quản lý bộ nhớ và đa luồng, những thứ mà đã có sẵn trong Java. Dalvik VM giúp mọ ứng dụng Android chạy trong tiến trình riêng của nó, với các thể hiện (instance) riêng của Dalvik virtual Machine.

Android Runtime cũng cung cấp 1 tập các thư viện chính giúp các nhà phát triển ứng dụng Android có thể viết ứng dụng Android bằng Java

* + 1. **Application Framework**

Lớp Android Framework cung cấp các dịch vụ cấp độ cao hơn cho các ứng dụng dưới dạng các lớp Java. Các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong ứng dụng của họ.

Android Framework bao gồm các dịch vụ chính sau:

* Activitty Manager - Kiểm soát tất cả khía cạnh của vòng đời ứng dụng và ngăn xếp các Activity.
* Content Providers - Cho phép các ứng dụng chia sẽ dữ liệu với các ứng dụng khác.
* Resource Manager - Cung cấp quyền truy cập vào các tài nguyên như các chuỗi, màu sắc, các layout giao diện người dùng...
* Notifications Manager - Cho phép các ứng dụng hiển thị cảnh báo và các thông báo cho người dùng.
* View System - Tập các thành phần giao diện (view) được sử dụng để tạo giao diện người dùng.
  + 1. **Application**

Lớp trên cùng của kiến trúc là Application. Các ứng dụng bạn tạo ra sẽ được cài đặt trên lớp này. Ví dụ như: Danh bạ, nhắn tin, trò chơi...

* 1. **Công cụ Android Studio:**

Android Studio là IDE chính thức được sử dụng trong phát triển ứng dụng Android dựa trên IntelliJ IDEA. Chức năng chính của Android Studio là cung cấp các giao diện giúp người dùng có thể tạo các ứng dụng và xử lý các công cụ file phức tạp sau hậu trường. Ngôn ngữ lập trình được sử dụng trong Android Studio là Java và nó sẽ được cài đặt sẵn trên thiết bị của bạn. Khi sử dụng Android Studio thì bạn chỉ cần viết, chỉnh sửa và lưu trữ chúng trên các dự án của mình và các file nằm trong dự án đó. Đồng thời, Android Studio còn cung cấp quyền truy cập vào Android SDK.

Ngoài ra, bạn có thể xem Android Studio là đuôi cho code Java cho phép nó chạy trơn tru trên các thiết bị Android rồi tận dụng được lợi thế của phần cứng gốc. Bạn chỉ cần sử dụng ngôn ngữ lập trình Java để có thể viết chương trình, khi đó Android SDK sẽ có nhiệm vụ kết nối các phần này lại với nhau. Khi đó, Android Studio sẽ kích hoạt để thực hiện chạy code và thông qua trình giả lập hoặc dựa vào bất kỳ phần cứng giúp kết nối với thiết bị. Sau đó, bạn có thể gỡ rối cho chương trình của mình ngay khi nó chạy và nhận phản hồi giúp giải thích các sự cố,...Cho đến hiện nay, Google đã và đang rất nỗ lực để giúp cho Android Studio sẽ trở nên mạnh mẽ và hữu ích hơn. Khi bạn gõ code, nó sẽ giúp bạn cung cấp danh sách gợi ý hoàn thành để giúp người dùng có thể hoàn thiện được dòng code đó. Đây là một trong những chức năng rất hữu ích đề phòng cho trường hợp người dùng không nhớ chính xác cú pháp giúp bạn tiết kiệm thời gian hiệu quả hơn.

Một vài tính năng nổi bật của Android Studio bao gồm:

* Một hệ thống xây dựng linh hoạt dựa trên Gradle
* Một trình mô phỏng nhanh và nhiều tính năng
* Một môi trường hợp nhất nơi bạn có thể phát triển cho mọi thiết bị Android
* Tính năng Live Edit (Chỉnh sửa trực tiếp) để cập nhật các thành phần kết hợp trong trình mô phỏng và thiết bị thực theo thời gian thực
* Mã mẫu và quá trình tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến cũng như nhập mã mẫu
* Đa dạng khung và công cụ thử nghiệm
* Công cụ tìm lỗi mã nguồn (lint) để nắm bắt hiệu suất, khả năng hữu dụng, khả năng tương thích với phiên bản và các vấn đề khác
* Hỗ trợ C++ và NDK
* Tích hợp sẵn tính năng hỗ trợ Google Cloud Platform, giúp dễ dàng tích hợp Google Cloud Messaging và App Engine.
  1. **Ngôn ngữ java**

Java là một một ngôn ngữ lập trình hiện đại, bậc cao, hướng đối tượng, bảo mật và mạnh mẽ. và là một Platform.

Platform: Bất cứ môi trường phần cứng hoặc phần mềm nào mà trong đó có một chương trình chạy, thì được hiểu như là một Platform. Với môi trường runtime riêng cho mình (JRE) và API, Java được gọi là Platform.

Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995. Phiên bản mới nhất của Java Standard Edition là Java SE 8. Với sự tiến bộ của Java và sự phổ biến rộng rãi của nó, nhiều cấu hình đã được xây dựng để phù hợp với nhiều loại nền tảng khác nhau. Ví dụ: J2EE cho các ứng dụng doanh nghiệp, J2ME cho các ứng dụng di động.

Các phiên bản J2 mới đã được đổi tên thành Java SE, Java EE và Java ME. Phương châm của java là "Write Once, Run Anywhere" - viết một lần chạy nhiều nơi, nghĩa là bạn chỉ cần viết một lần trên window chẳng hạn, sau đó vẫn chương trình đó bạn có thể chạy trên Linux, Android, các thiết bị J2ME...

* + 1. **Các tính năng của Java**

Ngôn ngữ lập trình java có các tính năng sau:

* **Hướng đối tượng** - Trong Java, mọi thứ đều là một Object. Java có thể dễ dàng mở rộng và bảo trì vì nó được xây dựng dựa trên mô hình Object.
* **Nền tảng độc lập** - Không giống nhiều ngôn ngữ lập trình khác bao gồm cả C và C ++, khi Java được biên dịch, nó không được biên dịch thành ngôn ngữ máy nền tảng cụ thể, thay vào mã byte - nền tảng độc lập. Mã byte này được thông dịch bởi máy ảo (JVM) trên nền tảng nào đó mà nó đang chạy.
* **Đơn giản** - Java được thiết kế để dễ học. Nếu bạn hiểu khái niệm cơ bản về OOP Java, sẽ rất dễ để trở thành master về java.
* **Bảo mật** - Với tính năng an toàn của Java, nó cho phép phát triển các hệ thống không có virut, giả mạo. Các kỹ thuật xác thực dựa trên mã hoá khóa công khai.
* **Kiến trúc - trung lập** - Trình biên dịch Java tạo ra định dạng tệp đối tượng kiến trúc trung lập, làm cho mã biên dịch được thực thi trên nhiều bộ vi xử lý, với sự hiện diện của hệ điều hành Java.
* **Portable** - Là kiến trúc tập trung và không có khía cạnh thực hiện phụ thuộc của đặc tả này làm cho Java khả chuyển. Trình biên dịch trong Java được viết bằng ANSI C, đó là một tập con POSIX.
* **Mạnh mẽ** - Java làm nỗ lực để loại trừ các tình huống dễ bị lỗi bằng cách kiểm tra lỗi tại thời gian biên dịch và kiểm tra lỗi tại runtime.
* **Đa luồng** - Với tính năng đa luồng của Java có thể viết các chương trình có thể thực hiện nhiều tác vụ đồng thời. Tính năng thiết kế này cho phép các nhà phát triển xây dựng các ứng dụng tương tác có thể chạy trơn tru hơn.
* **Thông dịch** - Mã byte Java được dịch trực tiếp tới các máy tính gốc và không được lưu trữ ở bất cứ đâu.
* **Hiệu năng cao** - Với việc sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện hiệu năng cao.
* **Phân tán** - Java được thiết kế cho môi trường phân tán của Internet.
* **Năng động** - Java là năng động hơn C hoặc C++ vì nó được thiết kế để thích nghi với môi trường đang phát triển. Các chương trình Java có thể mang một lượng lớn thông tin tại runtime mà có thể được sử dụng để xác minh và giải quyết các truy cập vào các đối tượng tại runtime.
  + 1. **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.

* 1. **SQLite**

SQLite là hệ quả trị cơ sở dữ liệu (DBMS) quan hệ tương tự như Mysql, ... Đặc điểm nổi bật của SQLite so với các DBMS khác là gọn, nhẹ, đơn giản, đặt biệt không cần mô hình server-client, không cần cài đặt, cấu hình hay khởi động nên không có khái niệm user, password hay quyền hạn trong SQLite Database. Dữ liệu cũng được lưu ở một file duy nhất.

SQLite thường không được sử dụng với các hệ thống lớn nhưng với những hệ thống ở quy mô vùa và nhỏ thì SQLite không thua các DBMS khác về chức năng hay tốc độ. Vì không cần cài đặt hay cấu hình nên SQLite được sử dụng nhiều trong việc phát triển, thử nghiệm … vì tránh được những rắc rối trong quá trình cài đặt.

Về mặt cú pháp SQLite hỗ trợ gần như đầy đủ các cú pháp trong chuẩn SQL92.

* + 1. **Tính năng của SQLite**

1. Giao dịch trong SQLite tuân thủ theo nguyên tắc (ACID) ngay cả sau hi hệ thống treo và mất điện.
2. Không cấu hình:

* Không cần thiết lập hoặc quản trị

1. SQLite hỗ trợ với đầy đủ tính năng với các khả năng nâng cao như các chỉ mục 1 phần, các chỉ mục về các biểu thức, JSON và các biểu thức bảng chung.
2. Một sở dữ liệu hoàn chỉnh được lưu trữ trong một tệp đa nền tảng duy nhất. Phù hợp với sử dụng dưới dạng định dạng tệp ứng dụng
3. Hỗ trợ các cơ sở dữ liệu có kích thước terabyte và các chuỗi có kích thước gigabyte.
4. API

* Đơn giản dễ sử dụng
* Nhanh: Trong một số trường hợp, SQLite nhanh hơn hệ thống tệp tin trực tiếp I/O.

1. Được viết bằng ANSI-C.

* Bindings cho hàng chục ngôn ngữ khác có sẵn 1 cách riêng biệt.

1. Mã nguồn đầy, nguồn mở đủ có thể kiểm tra nhánh 100%.
2. Nền tảng đa nền tảng:

* SQLite là có sẵn trên Android, \*BSD, iOS, Linux, Mac, Solaris, Windows,.. Dễ dàng dịch chuyển sang các hệ thống khác.
  + 1. **Ứng dụng chủ yêu của SQLite**

1. Cơ sở dữ liệu cho Internet Of Things.

SQLite là lựa chọn phổ biến cho các công cụ cơ sở dữ liệu trong điện thoại di động, PDA, máy nghe nhạc mp3, hộp set-top, và các tiện ích điện tử khác.

1. Định dạng tệp ứng dụng.

Thay vì sử dụng fopen() để viết XML, JSON, CSV hoặc một số định dạng động quyền vào các tệp đĩa được ứng dụng của bạn sử dụng, hãy sử dụng SQLite.

1. Cơ sở dữ liệu cho web.

Bởi vì SQLite không yêu cầu cấu hình và lưu trữ thông tin trong các tệp đĩa thông thường nên SQLite là lựa chọn phổ biến làm cơ sở dữ liệu để quay lại các trang web vừa và nhỏ.

1. Stand-in cho một doanh nghiệp RDBMS:

SQLite được sử dụng như một RDBMS doanh nghiệp cho mục đích trình diễn hoặc để thử nghiệm vì SQLite nhanh và không yêu cầu thiết lập.

* + 1. **Nhược điểm của SQLite**

Một số tính năng của SQL92 không được hỗ trợ trong SQLite như ALTER DROP COLUMN, ADD CONSTRAINT không được hỗ trợ; RIGHT JOIN; TRIGGER; phân quyền GRANT và REVOKE.

Vì SQLite không cần cấu hình, cài đặt, không hỗ trợ GRANT và REVOKE nên việc phân quyền truy cập cơ sở dữ liệu chỉ có thể là quyền truy cập file của hệ thống.

SQLite sử dụng cơ chế coarse-gained locking nên trong cùng một thời điểm có thể hỗ trợ nhiều người đọc dữ liệu, nhưng chỉ có 1 người có thể ghi dữ liệu.

SQLite không phù hợp với các hệ thống có nhu cầu xử lý trên một khối lượng dữ liệu lớn, phát sinh liên tục.

**PHẦN 2: KẾT QUẢ NGHIÊN CỨU**

* 1. **Tìm hiểu yêu cầu bài toán**

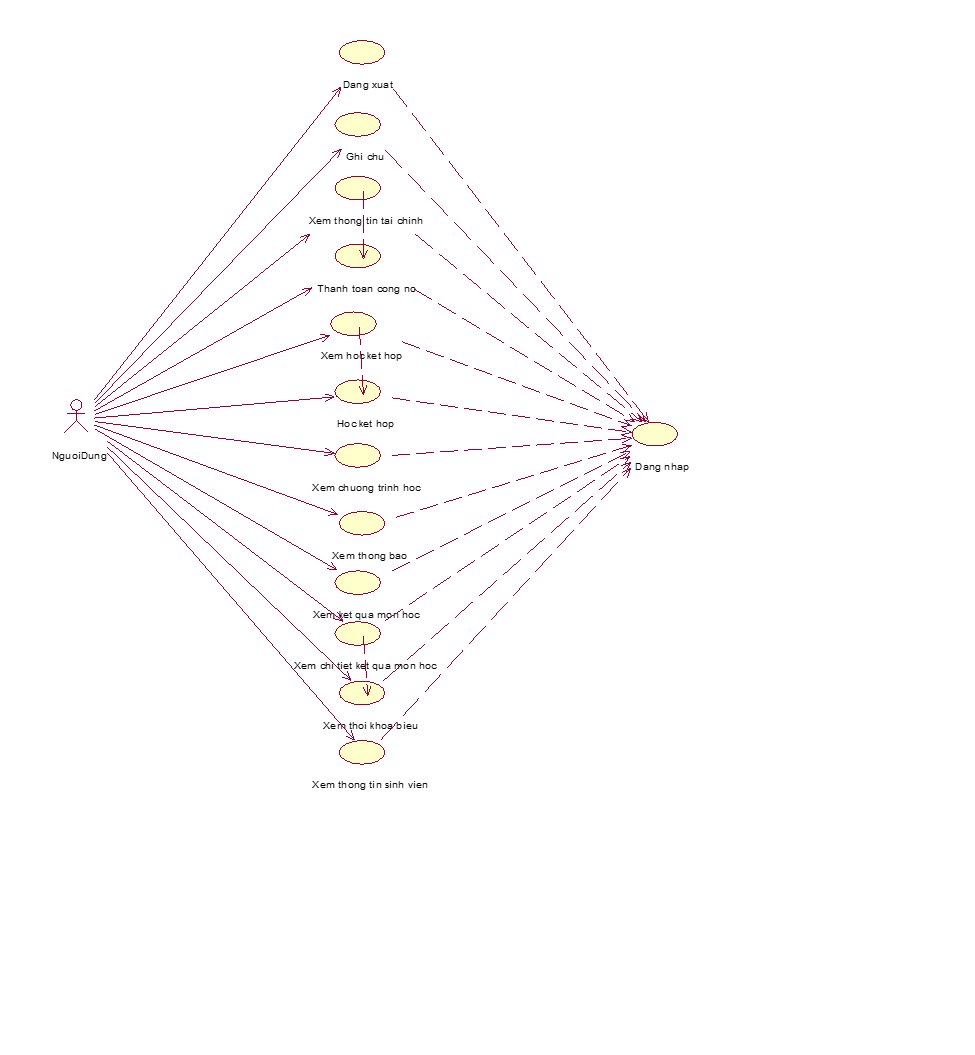
Tên bài toán: **Xây dựng sổ tay sinh viên trường ĐHCNHN**

Trong bối cảnh công nghệ phát triển nhanh chóng, việc ứng dụng công nghệ thông tin vào lĩnh vực giáo dục trở nên ngày càng phổ biến. Đặc biệt, với sự phổ cập của các thiết bị di động, việc xây dựng một ứng dụng sổ tay cho sinh viên trở thành một nhu cầu thiết yếu để tối ưu hóa quá trình học tập và quản lý thông tin cá nhân.

* 1. **Xây dựng bài toán**
     1. **Phân tích yêu cầu người dùng**

Yêu cầu chức năng:

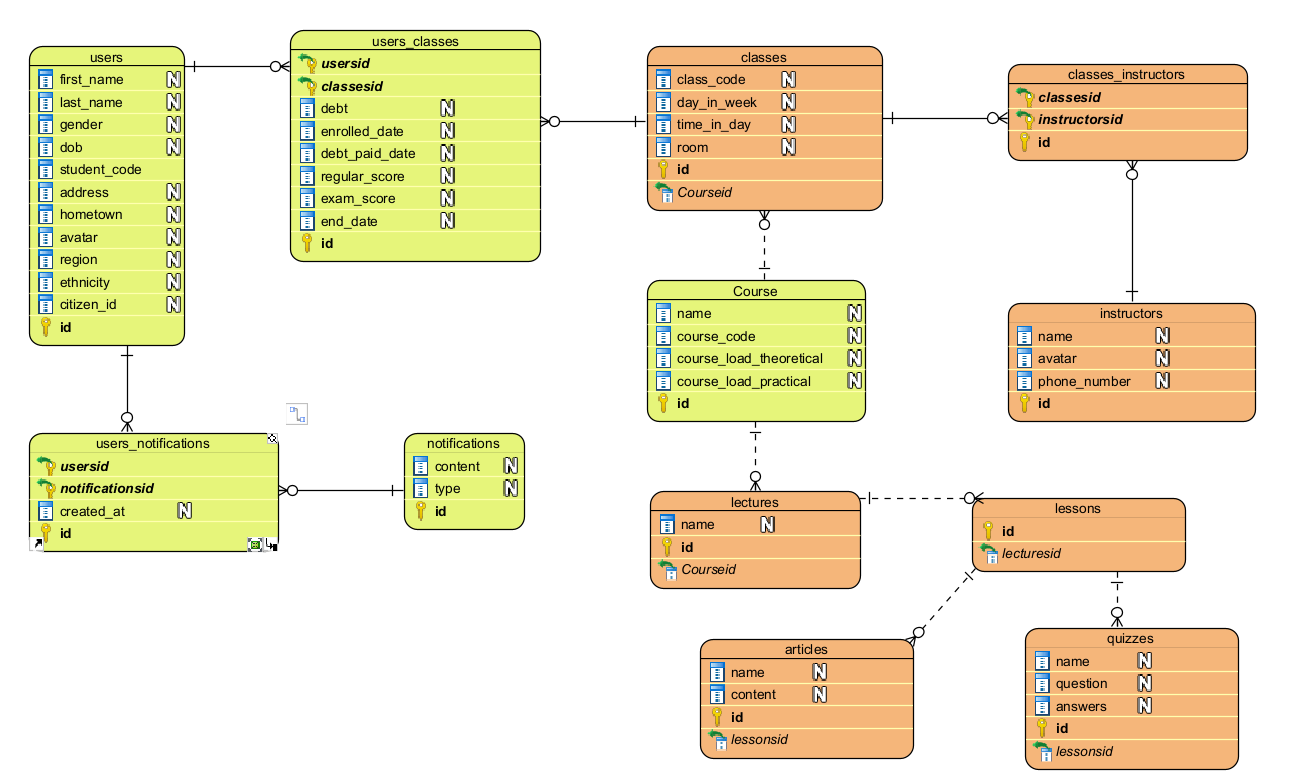
* Đăng nhập bằng mã sinh viên
* Đăng xuất
* Xem chương trình học
* Xem thông tin công nợ
* Xem thông tin hồ sơ sinh viên
* Xem thời khoá biểu
* Thông tin quá trình học tập
* Ghi chú
* Học kết hợp
* Xem thông báo



*Hình 1: Biểu đồ Use case*

Tóm tắt các nhiệm vụ trong chức năng (UC):

* Use case đăng nhập: Use case này cho phép người dùng có thể đăng nhập vào ứng dụng sổ tay sinh viên thông qua mã sinh viên.
* Use case Đăng xuất: Use case này cho phép người dùng có thể đăng xuất mã sinh viên của mình ra khỏi ứng dụng.
* Use case Ghi chú: Use case này cho phép người dùng lưu trữ một vài ghi chú và có thể xem lại trong ứng dụng.
* Use case Xem thông tin tài chính: Use case cho phép người dùng có thể nắm bắt được tình hình tài chính của bản thân bao gồm tổng số tiền còn phải thanh toán và các khoản thanh toán trong đó.
* Use case Thanh toán công nợ: Use case cho phép người dùng có thể thanh toán các công nợ chưa được trả.
* Use case Xem chương trình học: Use case cho phép người dùng xem thông tin chương trình học của mình bao gồm các môn bắt buộc và các môn tự chọn.
* Use case Xem thông báo: Use case cho phép người dùng có thể nhận và xem các thông báo từ phía ứng dụng.
* Use case Xem kết quả môn học: Use case cho phép người dùng có thể xem kết quả môn học mà mình đã học.
* Use case Xem chi tiết kết quả môn học: Use case cho phép người dùng xem chi tiết thông tin kết quả của một môn học gồm số tín chỉ, mã môn học, điểm quá trình, điểm thường xuyên và điểm thi.
* Use case Xem thời khóa biểu: Use case cho phép người dùng xem thông tin lịch học của mình theo từng ngày trong tuần.
* Use case Xem thông tin sinh viên: Use case cho phép người dùng xem lại thông tin cá nhân
* Use case Xem học kết hợp: Use case cho phép người dùng xem các môn học sẽ học kết hợp.
* Use case Học kết hợp: Use case cho phép người dùng học kết hợp.
  + 1. **Mô tả biểu đồ lớp dữ liệu**

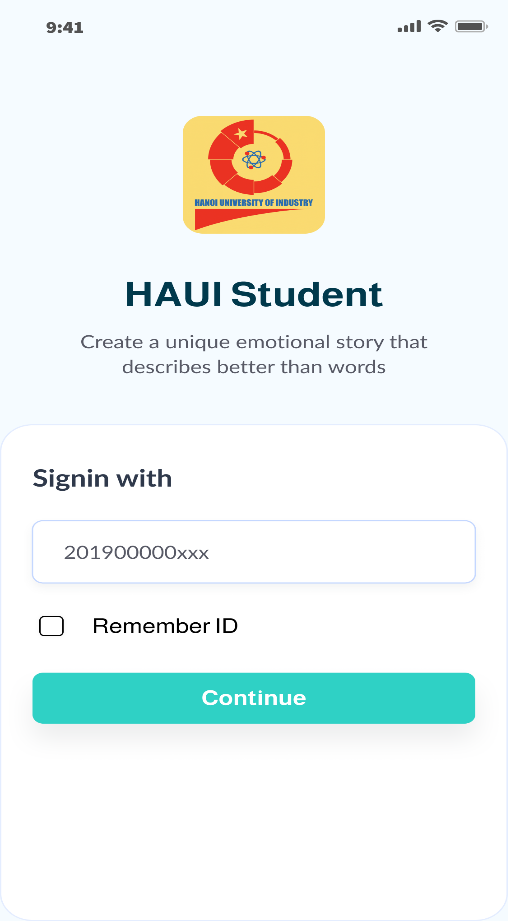


* Bảng user: lưu trữ các thông tin của một sinh viên.
* Bảng enrollment: lưu trữ các thông tin về tài chính và điểm của sinh viên.
* Bảng classes: chứa thông tin của một lớp học
* Bảng course: lưu trữ các thông tin của chương trình học.
* Bảng instructors: lưu trữ thông tin của các giáo viên.
* Bảng notifications: lưu trữ thông tin về các thông báo
* Bảng user\_notifications: lưu trữ các thông tin về ngày tạo thông báo.
* Bảng lectures: lưu trữ thông tin của các bài học.
* Bảng articles: lưu trữ thông tin về các bài giảng trong môn học.
* Bảng quizzes: lưu trữ thông tin về các bài tập trong môn học.
* Bảng lesson: lưu trữ thông tin chung của 2 bảng articles và quizzes.
  1. **Thiết kế hệ thống**

- Trước khi bắt đầu xây dựng ứng dụng, nhóm đã thống nhất một số quy tắc:

* Màu sắc:
  + Sử dụng màu sắc nhấn để làm nổi bật các phần quan trọng hoặc các hành động chính, ví dụ như sử dụng màu đỏ cho nút "Xóa" hoặc màu xanh lục cho nút "Thêm".
* Kích cỡ:
  + Định rõ kích cỡ và khoảng cách giữa các phần tử trên giao diện, đảm bảo tính cân đối và dễ nhìn.
  + Chú trọng đến kích thước và kiểu chữ, đảm bảo cho đọc dễ dàng
* Quy tắc đặt tên: Cần đặt tên dễ hiểu và phản ánh được nội dung của file đó, mỗi loại file, thư mục sẽ có các quy tắc đặt tên khác nhau
  + Package: Đặt tên chức thường, không có dấu cách
  + Class: Đặt tên theo quy tắc PascalCase, nếu là Adapter đặt tên nameAdapter, nếu là đối tượng đặt tên nameModel, nếu là Activity đặt tên Name
  + Layout: Đặt tên theo quy tắc snake\_case, nếu là Activity đặt tên activity\_name, nếu là listView đặt tên lv\_name.
    1. **Mô tả giao diện**

#### Màn hình đăng nhập:

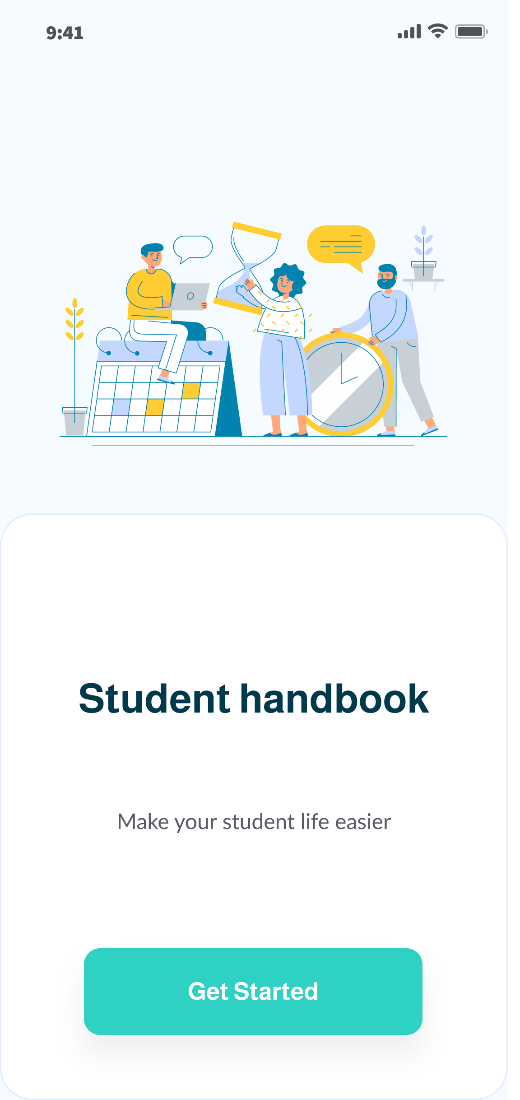


Màn hình đăng nhập bao gồm 1 editText để người dùng có thể nhập thông tin mã sinh viên.

Một checkBox giúp người dùng có thể lưu thông tin đăng nhập cho lần sử dụng sau.

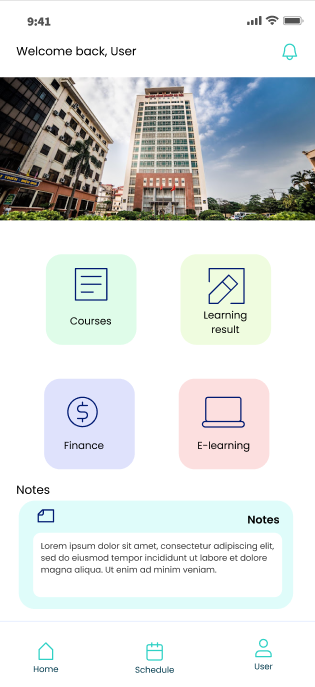
Một button giúp người dùng có thể đăng nhập và truy cập vào màn hình Onboarding.

#### Màn hình Onboarding:



Màn hình chào mừng có 1 button giúp người dùng vào màn hình chính.

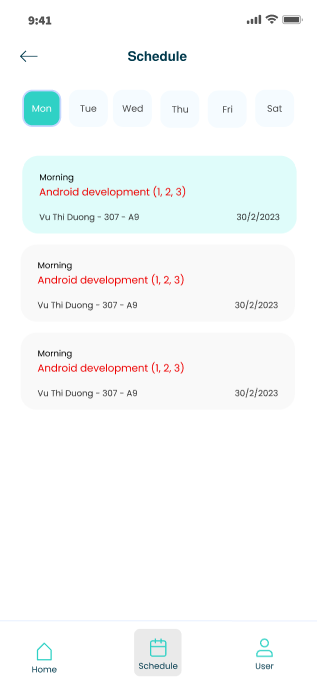
#### Màn hình chính:



Màn hình chính bao gồm 8 button, 1 editText:

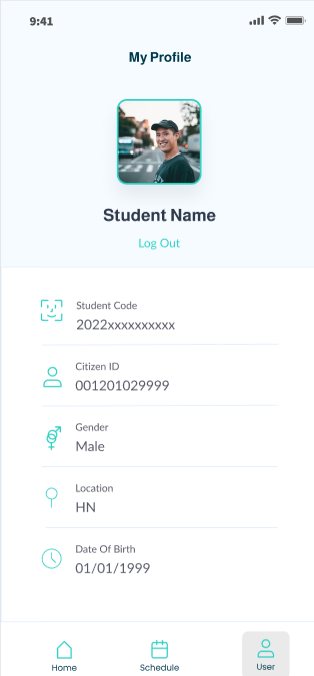
* Button thông báo có biểu tượng cái chuông cho phép người dùng vào màn hình Notifications.
* Button Course cho phép người dùng truy cập vào màn hình Course.
* Button Learning Course cho phép người dùng truy cập vào màn hình Result.
* Button Finance cho phép người dùng truy cập vào màn hình Finance.
* Button E-Learning cho phép người dùng truy cập vào màn hình Learning.
* Button Schedule cho phép người dùng truy cập vào màn hình Schedule.
* Button User cho phép người dùng truy cập vào màn hình User.
* Button Home cho phép người dùng truy cập vào màn hình chính.
* EditText cho phép người dùng lưu lại một số lưu ý và có thể xem lại vào lần truy cập sau.

#### Màn hình Schedule:



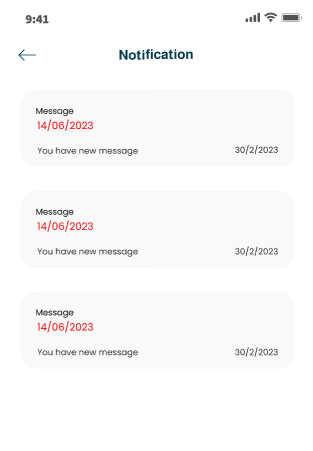
Màn hình bao gồm các button là các ngày trong tuần, 1 listView để hiển thị lịch học trong ngày.

#### Màn hình User:



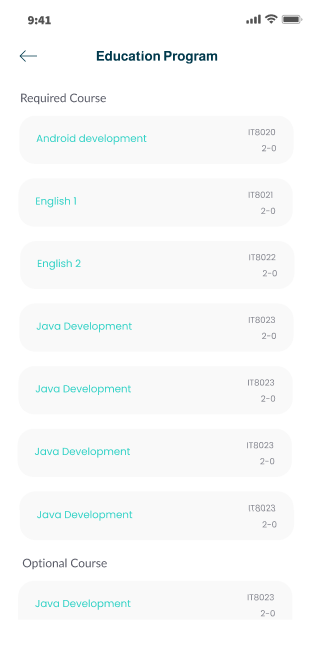
Màn hình bao gồm 1 textView Log out giúp người dùng có thể đăng nhập và quay trở lại màn hình đăng nhập, các textView để hiện thị thông tin sinh viên, 1 imageView hiển thị ảnh của sinh viên.

#### Màn hình Notifications:



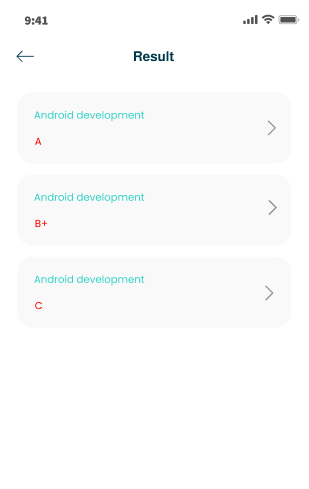
Màn hình gồm 1 button Back giúp người dùng quay lại màn hình trước đó, 1 listView để hiện thị các thông báo.

#### Màn hình Course:



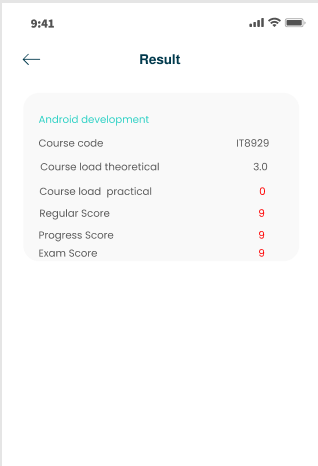
Màn hình bao gồm 1 button Back giúp người dùng quay lại màn hình trước đó, 2 listView hiển thị danh sách các môn học bắt buộc và môn học tự chọn.

#### Màn hình Result:



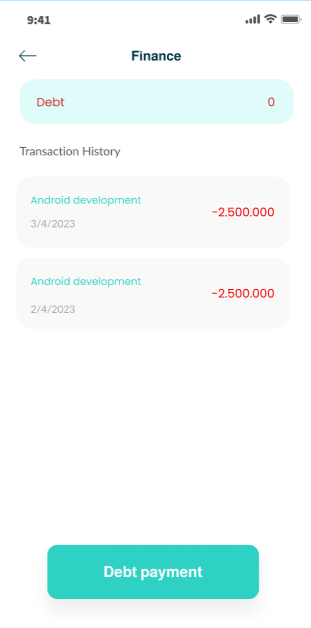
Màn hình bao gồm 1 button Back giúp người dùng quay lại màn hình trước đó, 1 listView hiển thị kết quả môn học. Khi kích vào 1 item trong listView chuyển sang màn hình Result Detail.

#### Màn hình Result Detail:



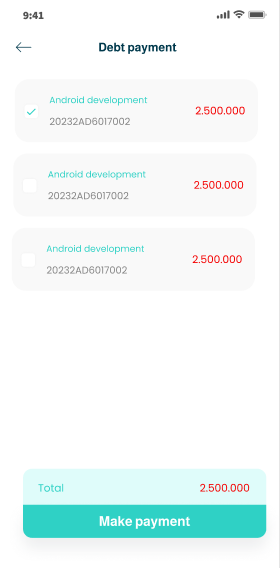
Màn hình bao gồm các textView hiển thị thông tin chi tiết của kết quả môn học, 1 button Back giúp người dùng quay lại màn hình trước đó.

#### Màn hình Finance:



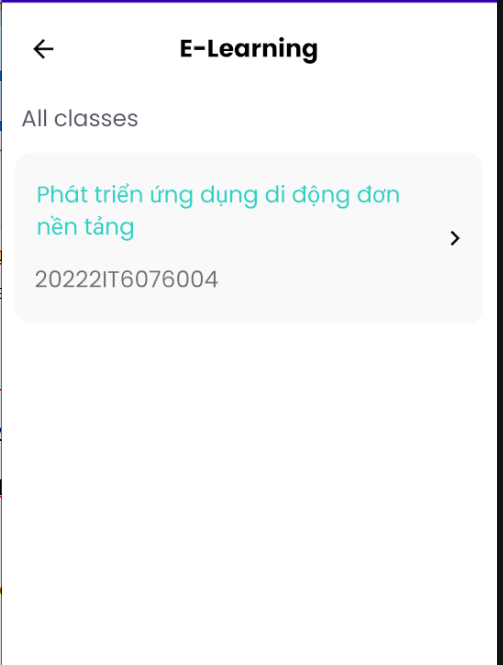
Màn hình này gồm 1 button Back giúp người dùng quay lại màn hình trước đó, 1 button Debt Payment giúp người dùng chuyển sang màn hình Payment, 1 listView hiển thị danh sách các môn học đã nộp tiền và 1 textView hiện thị tổng tiền chưa nộp.

#### Màn hình Payment:



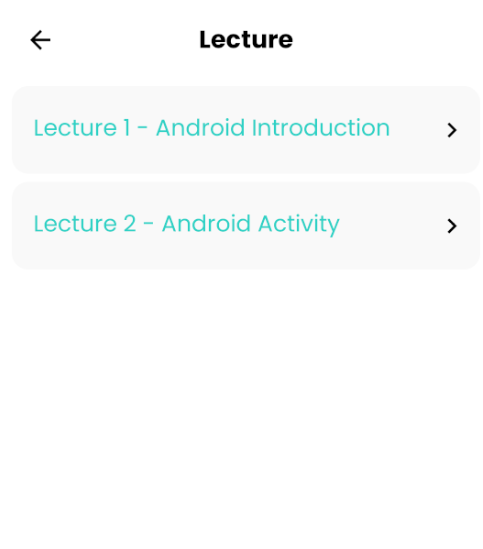
Màn hình này bao gồm 1 listView hiển thị các môn học còn chưa được thanh toán, 1 button ‘Make payment’ thực hiện thanh toán, 1 button Back để quay lại Activity trước đó.

#### Màn hình E-Learning:



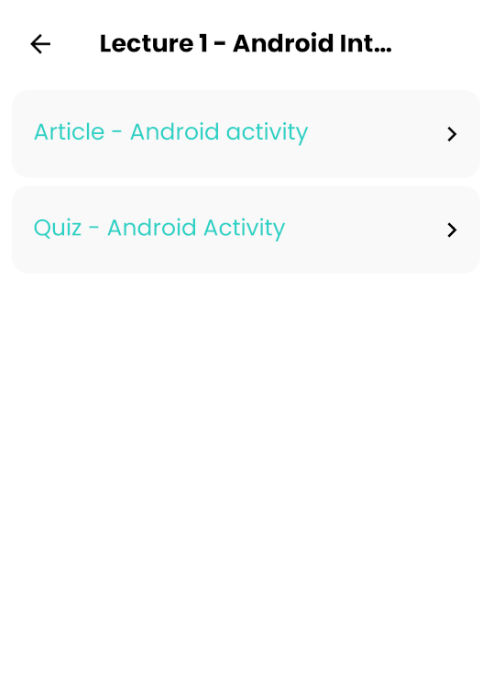
Màn hình bao gồm 1 listView hiển thị danh sách các môn học kết hợp, 1 buton Back để quay lại màn hình trước đó. Khi click vào item trong listView sẽ hiển thị ra màn hỉnh Lecture.

#### Màn hình Lecture:



Màn hình gồm 1 listView chứa có bài học, 1 button Back để quay lại màn hình trước đó, khi click vào item của listView sẽ hiển thị ra màn hình Lecture Detail.

#### Màn hình Lecture Detail



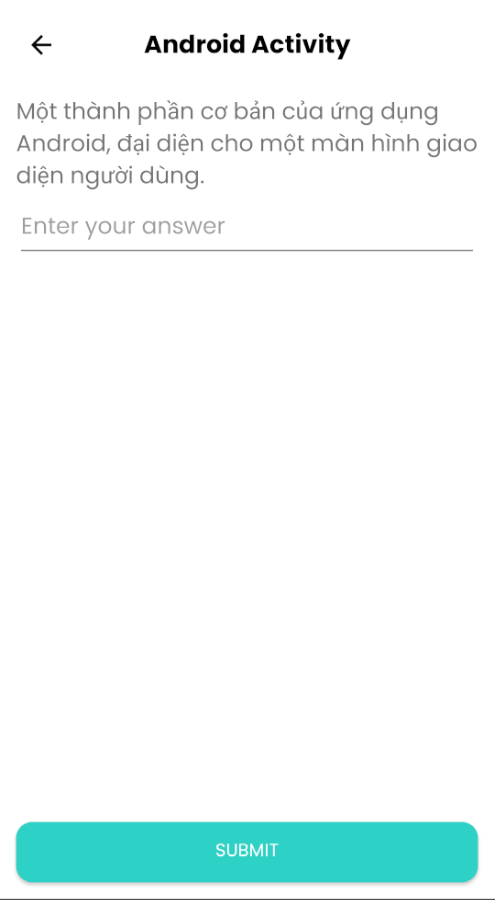
Màn hình gồm 1 listView chứa phần bài học và phần câu hỏi, 1 button Back để quay lại màn hình trước đó, khi click vào item bài học của listView sẽ hiển thị ra màn hình Article, khi click vào item câu hỏi sẽ hiển thị ra màn hình Quizzes.

#### Màn hình Article



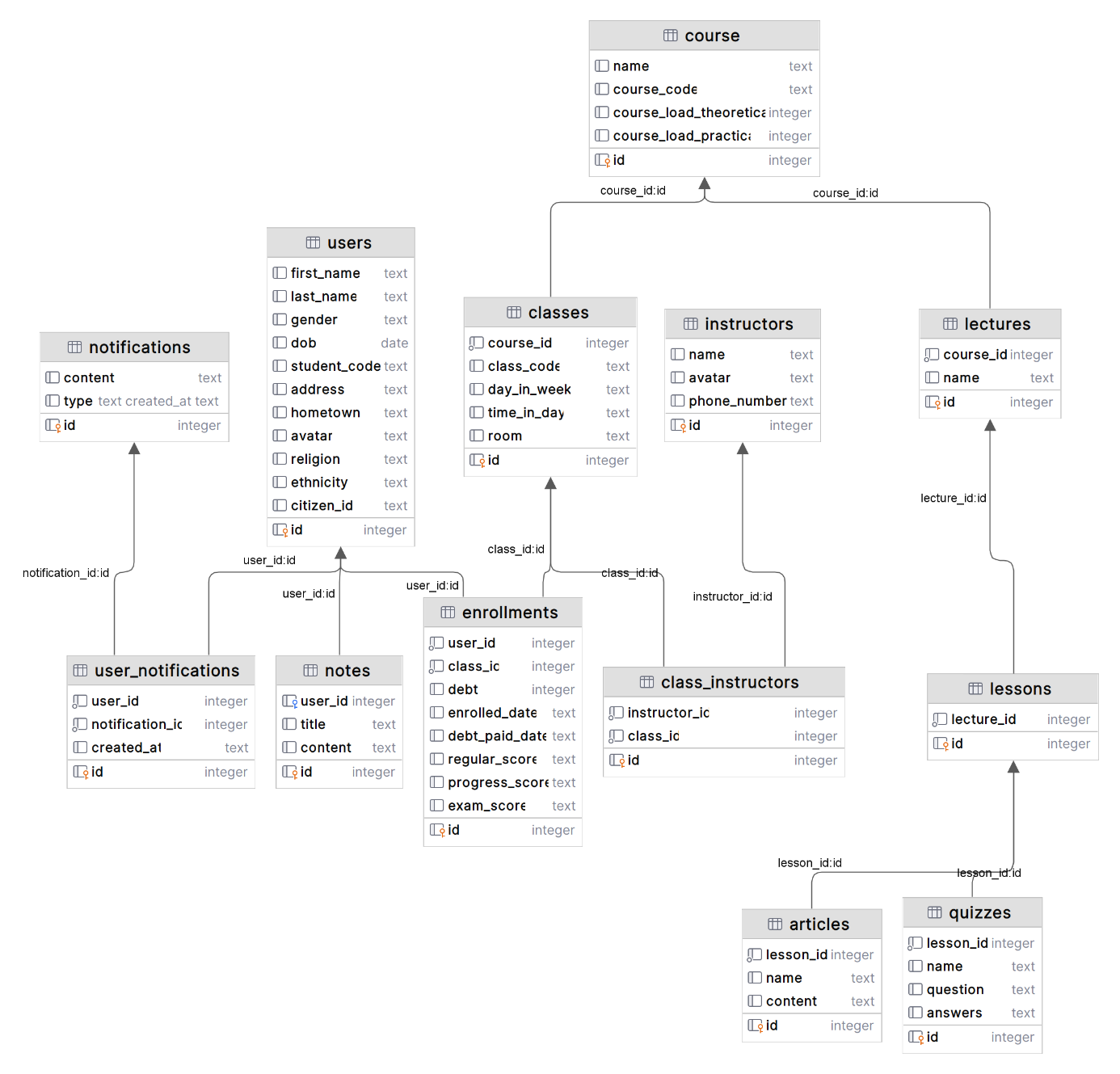
Màn hình gồm 1 webView chứa nội dung bài học, 1 button Back để quay lại màn hình trước đó.

#### Màn hình Quizzes

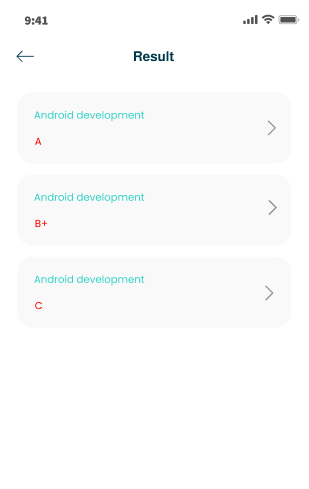


Màn hình bao gồm 1 button Back để quay lại màn hình trước đó, 1 textView hiển thị câu hỏi, 1 editText để nhập nội dung câu hỏi, 1 button ‘Submit’ hiển thị hộp thoại thông báo.

* + 1. **Thiết kế dữ liệu**



* 1. **Thực hiện bài toán**
     1. **Nguyễn Tuấn Kiệt – Xem kết quả môn học**
* **Màn hình thực hiện:**



* **Nhiệm vụ của Activity:**

Activity Result được dùng để hiển thị kết quả học tập của người dùng dưới dạng 1 listView. Khi người dùng click chọn vào một item sẽ mở ra màn hình Activity ResultDetail để xem chi tiết kết quả môn học đó và gửi đi đối tượng chứa thông tin kết quả qua cho màn hình đó.fjkjcvjaitjkcjvkjitfjkcáit

Tên môn học và điểm thành phần được lấy từ trong cơ sở dữ liệu SQLite.

Điểm dạng chữ được tính theo trung bình cộng tất cả các đầu điểm.

Nếu nhỏ hơn 4 trả về điểm ‘F’

Nếu từ 4 đến 5.4 trả về điểm ‘D’

Nếu từ 5.5 đến 6.9 trả về điểm ‘C’

Nếu từ 7 đến 8.4 trả về điểm ‘B’

Lớn hơn bằng 8.5 trả về điểm ‘A’

* **Mã nguồn giao diện màn hình Result**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".Result">

<androidx.constraintlayout.widget.ConstraintLayout

android:id="@+id/constraint"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="MissingConstraints">

<ImageView

android:id="@+id/btnBack"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_8sdp"

android:padding="@dimen/\_8sdp"

android:src="@drawable/ic\_arrow\_back\_24"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/textView5"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_8sdp"

android:padding="@dimen/\_8sdp"

android:text="Result"

android:textAlignment="center"

android:textColor="@color/black"

android:textSize="@dimen/\_15sdp"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

<ListView

android:id="@+id/lvResult"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="@dimen/\_8sdp"

android:padding="@dimen/\_8sdp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/constraint" />

</androidx.constraintlayout.widget.ConstraintLayout>

* **Thuật toán chính trong màn hình:**

**Đối tượng Result:**

Chứa các thông tin, constructor, getter and setter và phương thức getResult trả về điểm tổng kết dạng chữ:

private String nameCourse;

private String course\_code;

private String regularScore;

private String progressScore;

private String examScore;

private int course\_load\_theoretical;

private int course\_load\_practical;

public String getResult(){

int a = Integer.parseInt(getExamScore());

int b = Integer.parseInt(getProgressScore());

int c = Integer.parseInt(getRegularScore());

double sum = (a + b + c) \* 1.0 / 3;

if(sum < 4)

return "F";

else if(sum >= 4 && sum < 5.5)

return "D";

else if(sum >= 5.5 && sum < 7)

return "C";

else if(sum >= 7 && sum < 8.5)

return "B";

else return "A";

}

**Đối tượng ResultRepository lấy dữ liệu từ SQLite :**

Chứa phương thức getResultByStudentID lấy ra danh sách đối tượng Result từ id:

public ArrayList<ResultModel> getResultByStudentCode(String studentCode) {

ArrayList<ResultModel> list = new ArrayList<>();

//Câu lệnh lấy dữ liệu

Cursor cursor = rawQuery("SELECT course.name, course\_code, regular\_score, progress\_score, exam\_score, course\_load\_theoretical, course\_load\_practical \n" +

"FROM enrollments \n" +

"INNER JOIN classes ON enrollments.class\_id = classes.id\n" +

"INNER JOIN course on classes.course\_id = course.id\n" +

"WHERE user\_id = " + studentCode);

//Đưa dữ liệu vào danh sách

while (cursor.moveToNext()) {

ResultModel resultModel = new ResultModel();

resultModel.setNameCourse(cursor.getString(0));

resultModel.setCourse\_code(cursor.getString(1));

resultModel.setRegularScore(cursor.getString(2));

resultModel.setProgressScore(cursor.getString(3));

resultModel.setExamScore(cursor.getString(4));

resultModel.setCourse\_load\_theoretical(cursor.getInt(5));

resultModel.setCourse\_load\_practical(cursor.getInt(6));

list.add(resultModel);

}

return list;

}

* **Tạo ra một custom adapter:**

public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

View view = inflater.inflate(R.layout.lv\_result, parent, false);

//Lấy đối tượng Result theo vị trí position

ResultModel resultModel = getItem(position);

//Ánh xạ

TextView name = view.findViewById(R.id.txtResultName);

TextView result = view.findViewById(R.id.txtResult);

//Set dữ liệu

name.setText(resultModel.getNameCourse());

result.setText(resultModel.getResult());

return view;

}

* **Trong Activity Result:**

//Ánh xạ

private void getWidget(){

btnBack = findViewById(R.id.btnBack);

lvResult = findViewById(R.id.lvResult);

databaseHelper = new DatabaseHelper(this);

//Đối tượng chứa phương thức getResultByStudentCode trả về một mảng danh sách

resultRepository = new ResultRepository(databaseHelper);

//Danh sách đối tượng Result;

resultList = new ArrayList<>();

resultList.clear();

//Lấy dữ liêu từ db ra danh sách qua phương thức getResultByStudentCode

resultList.addAll(resultRepository.getResultByStudentCode((((UserID) getApplication()).userId) + ""));

resultAdapter = new ResultAdapter(this, resultList);

lvResult.setAdapter(resultAdapter);

}

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_result);

getWidget();

//Sự kiện click vào item trong listView

lvResult.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

//Lấy dữ liệu từ danh sách sau đó truyền qua Activity Result\_Detail

ResultModel resultModel = resultList.get(i);

Intent intent = new Intent(Result.this, Result\_Detail.class);

intent.putExtra("resultModel", resultModel);

startActivity(intent);

}

});

//Bắt sự kiện quay lại

btnBack.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

onBackPressed();

}

});

}

* + 1. **Nguyễn Văn Mạnh – Xem thời khóa biểu**
* **Màn hình thực hiện:**

A screenshot of a phone

Description automatically generated with medium confidence

* **Nhiệm vụ của Activity:**

Hiển thị thời khoá biểu cho sinh viên đang đăng nhập theo ngày trong tuần

Dữ liệu được lấy từ database SQL Lite dựa trên thời gian đăng kí môn học

Có thể chọn thứ trong tuần để xem theo ngày

* **Mã nguồn giao diện màn hình Schedule**

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="fragment.ScheduleFragment">

<androidx.constraintlayout.widget.ConstraintLayout

android:id="@+id/constraint"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

tools:ignore="MissingConstraints">

<LinearLayout

android:id="@+id/changeWeekLayout"

android:layout\_width="match\_parent"

android:paddingLeft="@dimen/\_8sdp"

android:paddingRight="@dimen/\_8sdp"

android:paddingTop="@dimen/\_15sdp"

android:layout\_height="wrap\_content">

<ImageView

android:id="@+id/btnPreviousWeek"

android:layout\_width="@dimen/\_24sdp"

android:layout\_height="@dimen/\_24sdp"

android:src="@drawable/baseline\_chevron\_left\_24" />

<TextView

android:id="@+id/tvCurrentWeek"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:gravity="center"

android:layout\_weight="1"

android:textSize="@dimen/\_14sdp"

android:textColor="@color/black"

tools:text="10/8/2023 - 17/8/2023" />

<ImageView

android:id="@+id/btnNextWeek"

android:layout\_width="@dimen/\_24sdp"

android:layout\_height="@dimen/\_24sdp"

android:src="@drawable/baseline\_chevron\_right\_24"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingStart="@dimen/\_5sdp"

android:paddingEnd="@dimen/\_5sdp"

android:layout\_marginTop="@dimen/\_15sdp"

android:showDividers="middle"

app:layout\_constraintTop\_toBottomOf="@+id/changeWeekLayout">

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/mondayBtn"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_3sdp"

android:layout\_weight="1"

android:background="@drawable/cus\_button"

android:maxLines="1"

android:stateListAnimator="@null"

android:text="Mon"

android:textSize="@dimen/\_7sdp" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/tuesdayBtn"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_3sdp"

android:layout\_weight="1"

android:background="@drawable/cus\_button\_secondary"

android:maxLines="1"

android:stateListAnimator="@null"

android:text="Tue"

android:textSize="@dimen/\_7sdp" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/wednesdayBtn"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_3sdp"

android:layout\_weight="1"

android:background="@drawable/cus\_button\_secondary"

android:maxLines="1"

android:stateListAnimator="@null"

android:text="Wed"

android:textSize="@dimen/\_7sdp" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/thursdayBtn"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_3sdp"

android:layout\_weight="1"

android:background="@drawable/cus\_button\_secondary"

android:maxLines="1"

android:stateListAnimator="@null"

android:text="Thu"

android:textSize="@dimen/\_7sdp" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/fridayBtn"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_3sdp"

android:layout\_weight="1"

android:background="@drawable/cus\_button\_secondary"

android:maxLines="1"

android:stateListAnimator="@null"

android:text="Fri"

android:textSize="@dimen/\_7sdp" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/saturdayBtn"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_3sdp"

android:layout\_weight="1"

android:background="@drawable/cus\_button\_secondary"

android:maxLines="1"

android:stateListAnimator="@null"

android:text="Sat"

android:textSize="@dimen/\_7sdp" />

<androidx.appcompat.widget.AppCompatButton

android:id="@+id/sundayBtn"

android:layout\_width="wrap\_content"

android:layout\_height="@dimen/\_40sdp"

android:layout\_margin="@dimen/\_3sdp"

android:layout\_weight="1"

android:background="@drawable/cus\_button\_secondary"

android:maxLines="1"

android:stateListAnimator="@null"

android:text="Sun"

android:textSize="@dimen/\_7sdp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

<ListView

android:id="@+id/lvSchedule"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_marginTop="@dimen/\_8sdp"

android:divider="#00ffffff"

android:dividerHeight="@dimen/\_5sdp"

android:overScrollMode="never"

android:paddingStart="@dimen/\_8sdp"

android:paddingEnd="@dimen/\_8sdp"

android:scrollbars="none"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/constraint"

app:layout\_constraintVertical\_bias="0.0"

tools:layout\_editor\_absoluteX="0dp"

tools:listitem="@layout/schedule\_class\_item" />

<ImageView

android:id="@+id/noDataPlaceholder"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/no\_data\_placeholder"

app:layout\_constraintTop\_toBottomOf="@id/constraint"

app:layout\_constraintStart\_toStartOf="parent"

android:visibility="gone"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

* **Thuật toán chính trong màn hình:**

**Đối tượng Class:**

Chứa các thông tin, constructor, getter and setter :

private String courseId;

private String classCourse;

private String dayInWeek;

private String timeInDay;

private String room;

private String classCode;

public ClassModel(String courseId, String classCourse, String dayInWeek, String timeInDay, String room, String classCode){

this.courseId = courseId;

this.classCourse = classCourse;

this.dayInWeek = dayInWeek;

this.timeInDay = timeInDay;

this.room = room;

this.classCode = classCode;

}

public ClassModel(){}

public String getCourseId() {

return courseId;

}

public void setCourseId(String courseId) {

this.courseId = courseId;

}

public String getClassCourse() {

return classCourse;

}

public void setClassCourse(String classCourse) {

this.classCourse = classCourse;

}

public String getDayInWeek() {

return dayInWeek;

}

public void setDayInWeek(String dayInWeek) {

this.dayInWeek = dayInWeek;

}

public String getTimeInDay() {

return timeInDay;

}

public void setTimeInDay(String timeInDay) {

this.timeInDay = timeInDay;

}

public String getRoom() {

return room;

}

public void setRoom(String room) {

this.room = room;

}

public String getClassCode() {

return classCode;

}

public void setClassCode(String classCode) {

this.classCode = classCode;

}

**Đối tượng UserRepository lấy dữ liệu schedule từ SQLite :**

Chứa phương thức getClassByUserID lấy ra danh sách đối tượng Class từ id:

ArrayList<ClassModel> list = new ArrayList<>();

Cursor cursor = rawQuery("select day\_in\_week, time\_in\_day, room, class\_code, cl.course\_id from users inner join enrollments er on users.id = er.user\_id\n" +

"inner join classes cl on er.class\_id = cl.id\n" +

"where day\_in\_week like '%" + dayOfWeek + "%'\n" +

"and users.student\_code = 1 and er.enrolled\_date < '" + fromDate + "' and er.end\_date > CURRENT\_DATE");

while (cursor.moveToNext()) {

ClassModel classModel = new ClassModel();

classModel.setDayInWeek(cursor.getString(0));

classModel.setTimeInDay(cursor.getString(1));

classModel.setRoom(cursor.getString(2));

classModel.setRoom(cursor.getString(3));

classModel.setCourseId(cursor.getString(4));

list.add(classModel);

}

cursor.close();

return list;

* **Tạo ra một custom adapter cho schedule:**

public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

View view = inflater.inflate(R.layout.schedule\_class\_item, parent, false);

ClassModel classModel = getItem(position);

CourseModel courseModel = courseRepository.findById(Long.valueOf(classModel.getCourseId()));

TextView name = view.findViewById(R.id.schedule\_name);

TextView time = view.findViewById(R.id.schedule\_time);

TextView date = view.findViewById(R.id.schedule\_date);

TextView classInfo = view.findViewById(R.id.schedule\_class\_info);

name.setText(courseModel.getName());

time.setText("Lesson: " + classModel.getTimeInDay());

date.setText("");

classInfo.setText(classModel.getRoom());

return view;

}

* **Trong Activity Result:**

private void onSelectDay(View v) {

if (btnSelected != null) {

btnSelected.setBackgroundResource(R.drawable.cus\_button\_secondary);

}

btnSelected = (Button) v;

int dayInWeek = 1;

for (int id : btnIds) {

if (btnSelected.getId() == id) {

break;

}

dayInWeek++;

}

v.setBackgroundResource(R.drawable.cus\_button);

setClassList(dayInWeek);

}

private void setClassList(int dayInWeek) {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd 00:00:00", Locale.US);

classList.clear();

classList.addAll(userRepository.getClassByStudentCode(homePage.getStudentID() + "", "" + dayInWeek, sdf.format(currentWeek.getTime())));

if (classList.size() == 0) {

noDataPlaceholder.setVisibility(View.VISIBLE);

} else {

noDataPlaceholder.setVisibility(View.GONE);

}

classAdapter.notifyDataSetChanged();

}

private String getWeekString(Calendar calendar) {

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy", Locale.US);

String startDate = sdf.format(calendar.getTime());

Calendar calendar2 = (Calendar) calendar.clone();

calendar2.add(Calendar.DAY\_OF\_MONTH, 6);

String endDate = sdf.format(calendar2.getTime());

return String.format("%s - %s", startDate, endDate);

}

# **PHẦN 3: KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM**

## 3.1. Nội dung đã thực hiện

Thông qua bài tập lớn nhóm 6 chúng em đã thực hiện được một số điều sau:

- Nắm rõ được kiến thức liên quan đến việc xử lý các sự kiện trên một ứng dụng android, thiết kế màn hình giao diện, làm việc với SQLite xây dựng cơ sở dữ liệu cục bộ cho ứng dụng.

- Có kỹ năng phân tích nghiệp vụ, xây dựng sơ đồ quan hệ, biểu đồ lớp cho một ứng dụng android.

- Có kĩ năng làm việc với nhóm.

- Có kĩ năng tìm kiếm và giải quyết các vấn đề gặp phải.

- Có thêm vốn từ vựng Tiếng Anh liên quan đến ngành học.

## 3.2. Kết quả đạt được

Những chuẩn đầu ra của học phần mà nhóm 6 đã đạt được trong quá trình thực hiện Bài tập lớn:

* Cài đặt được bài toán đơn giản sử dụng các thành phần chính trong lập trình di động trên nền tảng Android.
* Cài đặt được bài toán cụ thể trên thiết bị di động trên nền tảng Android
* Triển khai được ứng dụng thực tế trên nền tảng Android

## 3.3. Hạn chế của đề tài

Do thời gian có hạn nên đề tài của nhóm vẫn chưa được hoàn chỉnh nhất, chưa tìm hiểu được hết những vấn đề và bài toán phát sinh trong đề bài.

## 3.4. Hướng phát triển

* Tích hợp tính năng tương tác giữa sinh viên với sinh viên, giữa sinh viên với giáo viên
* Tich hợp đăng ký học phần
* Tiếp tục nâng cao hiệu suất và tối ưu hóa ứng dụng để đảm bảo hoạt động mượt mà trên các thiết bị di động khác nhau

# **Tài liệu tham khảo**

[1]. Android Developer Docs - <https://developer.android.com/docs>

[2]. Android Repository Pattern - <https://developer.android.com/codelabs/basic-android-kotlin-training-repository-pattern>

[3]. <https://xuanthulab.net/android-java/>