**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**THỰC TẬP CƠ SỞ**

*đề tài*:

**TÌM HIỂU NỀN TẢNG LẬP TRÌNH DI ĐỘNG VÀ XÂY DỰNG ỨNG DỤNG QUẢN LÝ THỜI GIAN**

**Giảng viên hướng dẫn:** Nguyễn Thị Tính

**Sinh viên thực hiện:** Nguyễn Tuấn Trường

**Lớp:** KTPM-K17B

**MSSV:** DTC1854801030076

Thái Nguyên, năm 2021

**LỜI CẢM ƠN**

Để hoàn thành tốt bài báo cáo thực tập cơ sở này trước hết em xin gửi đến quý thầy, cô giáo trong khoa Công Nghệ Thông Tin trường Đại Học Công Nghệ Thông Tin và Truyền Thông Thái Nguyên lời cảm ơn chân thành.

Đặc biệt, em xin gửi đến thầy *Nguyễn Thị Tính*, người đã tận tình hướng dẫn, giúp đỡ trong việc nâng cao kiến thức chuyên môn, phát huy cho em tính tự học, tự tìm hiểu, từ đó hình thành cho em phương pháp nghiên cứu, tìm tòi, tiếp cận với khoa học công nghệ. Đồng thời trau dồi cho em kỹ năng học tập năng động và sáng tạo, giúp cho em tiếp cận các ứng dụng thông minh trong thực tế một cách thiết thực hơn.

Vì kiến thức bản thân còn hạn chế, nên trong quá trình xây dựng đề tài này em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ quý thầy cô.

Em xin trân trọng cảm ơn!

Thái Nguyên, Ngày tháng năm 2021

Sinh viên thực hiện

**LỜI NÓI ĐẦU**

Cùng với cách mạng công nghiệp 4.0, ngành công nghệ thông tin nói chung đang phát triển mạnh mẽ và có mặt trong hầu hết mọi lĩnh vực trong đời sống tạo nên bước đột phá mạnh mẽ trong việc thúc đẩy sự phát triển của các ngành khác.

Ngày này những chiếc Smartphone đang dần mạnh mẽ và tiện lợi hơn, có thể thay thế những nhu cầu thiết yếu trước đây của các thiết bị vi tính. Có thể nói, Điện thoại thông minh bây giờ là vật bất ly thân của nhiều người, Nhưng cái gì nhiều quá cũng không tốt, Để quản lý thời gian sinh hoạt, học tập cho bản thân nói riêng và cho sinh viên Thái Nguyên nói chung cần một ứng dụng để quản lý thời gian cho sinh viên hiệu quả.Từ những nhu cầu thực tế trên, người thực hiện đề tài đã chọn đề tài “Xây dựng ứng dụng quản lý thời gian”

Qua thời gian học tập tại trường Đại học Công Nghệ Thông Tin và Truyền Thông Thái Nguyên, dưới sự tận tâm truyền đạt của các thầy cô giáo trong khoa Công nghệ thông tin, người thực hiện đề tài đã trau dồi và tích lũy được những vốn kiến thức nhất định. Từ việc thực hiện đề tài, người thực hiện đề tài mong muốn cải thiện khả năng lập trình cũng như nâng cao khả năng tự nghiên cứu, học hỏi để có thể đáp ứng tốt cho công việc sau này.

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI 8](#_Toc69388510)

[1.1 Tên đề tài 8](#_Toc69388511)

[1.2 Lý do chọn đề tài 8](#_Toc69388512)

[1.3 Mục đích nghiên cứu 8](#_Toc69388513)

[1.4 Tổng quan về lập trình trên hệ điều hành android 9](#_Toc69388514)

[1.4.1 Hệ điều hành Android là gì? 9](#_Toc69388515)

[1.4.2 Tìm hiểu về kiến trúc của Android 10](#_Toc69388516)

[1.4.3 Các phiên bản của hệ điều hành Android 10](#_Toc69388517)

[1.4.4 Ngôn ngữ dùng để lập trình Android 10](#_Toc69388518)

[1.4.5 Một số ngôn ngữ được sử dụng để lập trình Android 11](#_Toc69388519)

[1.4.6 Ngôn ngữ Java trong hệ điều hành Android 11](#_Toc69388520)

[1.5 Tổng quan về ngôn ngữ java 11](#_Toc69388521)

[1.5.1 Giống như C++, hướng đối tượng hoàn toàn 11](#_Toc69388522)

[1.5.2 Độc lập phần cứng và hệ điều hành 11](#_Toc69388523)

[1.5.3 Ngôn ngữ thông dịch 11](#_Toc69388524)

[1.5.4 Cơ chế thu gom rác tự động 12](#_Toc69388525)

[1.5.5 Đa luồng 12](#_Toc69388526)

[1.5.6 Tính an toàn và bảo mật 12](#_Toc69388527)

[1.5.7 Java Virtual Machine 13](#_Toc69388528)

[1.6 Tổng quan về ngôn ngữ đánh dấu XML 13](#_Toc69388529)

[1.7 Công cụ cần thiết cho lập trình android 14](#_Toc69388530)

[1.7.1 Giới thiệu về android studio 14](#_Toc69388531)

[1.7.2 Những tính năng chính của Android Studio: 15](#_Toc69388532)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG. 17](#_Toc69388533)

[2.1 Khảo sát một số ứng dụng hiện có 17](#_Toc69388534)

[2.1.1 myHomework Student Planner 17](#_Toc69388535)

[2.1.2 TimeTable- Weekly Schedule/Planner 18](#_Toc69388536)

[2.2 Yêu cầu từ phỏng vấn người dùng 19](#_Toc69388537)

[2.3 Usecase Hệ thống 20](#_Toc69388538)

[2.4 Kịch bản 23](#_Toc69388539)

[2.4.1 Kịch bản use case xem sự kiện 23](#_Toc69388540)

[2.4.2 Kịch bản use case thêm sự kiện 24](#_Toc69388541)

[2.4.3 Kịch bản use case sửa sự kiện 24](#_Toc69388542)

[2.4.4 Kịch bản use case xóa sự kiện 25](#_Toc69388543)

[2.4.5 Kịch bản use case xem báo thức 26](#_Toc69388544)

[2.4.6 Kịch bản use case sửa báo thức 26](#_Toc69388545)

[2.4.7 Kịch bản use case thêm báo thức 28](#_Toc69388546)

[2.4.8 Kịch bản use case xóa báo thức 28](#_Toc69388547)

[2.4.9 Kịch bản use case sử dụng api của SVO để đăng nhập và lấy lịch học 29](#_Toc69388548)

[2.5 Biểu đồ trạng thái (Activity diagram) 30](#_Toc69388549)

[2.5.1 Biểu đồ trạng thái cho use case thêm sự kiện 30](#_Toc69388550)

[2.5.2 Biểu đồ trạng thái cho use case xem sự kiện 31](#_Toc69388551)

[2.5.3 Biểu đồ trạng thái cho use case sử dụng SVO api để lấy lịch học 31](#_Toc69388552)

[2.5.4 Biểu đồ trạng thái cho use case xóa sự kiện 32](#_Toc69388553)

[2.6 Biểu đồ trình tự (Sequence diagram) 32](#_Toc69388554)

[2.6.1 Biểu đồ trình tự cho usecase xem sự kiện 32](#_Toc69388555)

[2.6.2 Biểu đồ trình tự cho usecase sử dụng SVO api để lấy lịch học 33](#_Toc69388556)

[2.7 Cơ sở dữ liệu 34](#_Toc69388557)

[Mô tả chi tiết bảng trong cơ sở dữ liệu 34](#_Toc69388558)

[CHƯƠNG III. CÀI ĐẶT CHƯƠNG TRÌNH 35](#_Toc69388559)

[3.1 1 Thiết kế giao diện. 35](#_Toc69388560)

[3.1.1 Giao diện cơ bản khi mở ứng dụng 36](#_Toc69388561)

[3.1.2 Một số chức năng 37](#_Toc69388562)

[3.1.3 Giao diện khi chọn chức năng thêm sự kiện mới 38](#_Toc69388563)

[3.1.4 Màn hình chức năng sửa sự kiện có sẵn 38](#_Toc69388564)

[3.1.5 Giao diện khi kéo phải một sự kiện 39](#_Toc69388565)

[3.1.6 Màn hình chính chức năng ngủ khoa học 40](#_Toc69388566)

[3.1.7 Màn hình một số chức năng của chức năng ngủ khoa học 41](#_Toc69388567)

[3.1.8 Màn hình chức năng cá nhân hóa 42](#_Toc69388568)

[3.1.9 Ảnh chụp một số chức năng khác của giao diện cá nhân hóa 44](#_Toc69388569)

[KẾT LUẬN 45](#_Toc69388570)

[TÀI LIỆU THAM KHẢO 46](#_Toc69388571)

**MỤC LỤC ẢNH**

[Hình 1.1 Cấu trúc hệ điều hành android 7](#_Toc69385475)

[Hình 1.2 Mô hình thông dịch android 9](#_Toc69385476)

[Hình 1.3 Giao diện người dùng của Android Studio 13](#_Toc69385477)

[Hình 2.1 ứng dụng myHomework Student Planner 14](#_Toc69385478)

[Hình 2.2 Ứng dụng TimeTable- Weekly Schedule/Planner 15](#_Toc69385479)

[Hình 2.3 Usecase tổng quát hệ thống 17](#_Toc69385480)

[Hình 2.4 Usecase quản lý sự kiện 18](#_Toc69385481)

[Hình 2.5 Usecase Cài đặt hệ thống 18](#_Toc69385482)

[Hình 2.6 Usecase cài đặt tài khoản 19](#_Toc69385483)

[Hình 2.7 Biểu đồ trạng thái cho use case thêm sự kiện 25](#_Toc69385484)

[Hình 2.8 Biểu đồ trạng thái cho use case xem sự kiện 26](#_Toc69385485)

[Hình 2.9 Biểu đồ trạng thái cho use case sử dụng SVO api để lấy lịch học 26](#_Toc69385486)

[Hình 2.10 Biểu đồ trạng thái cho use case xóa sự kiện 27](#_Toc69385487)

[Hình 2.11 Biểu đồ trình tự cho usecase xem sự kiện 27](#_Toc69385488)

[Hình 2.12 Biểu đồ trình tự cho usecase sử dụng SVO api để lấy lịch học 28](#_Toc69385489)

[Hình 3.1 Giao diện cơ bản khi mở ứng dụng 30](#_Toc69385490)

[Hình 3.2 Một số chức năng 31](#_Toc69385491)

[Hình 3.3 Giao diện khi chọn chức năng thêm sự kiện mới 32](#_Toc69385492)

[Hình 3.4 Giao diện chức năng sửa sự kiện có sẵn 32](#_Toc69385493)

[Hình 3.5 Màn hình chính chức năng ngủ khoa học 34](#_Toc69385494)

[Hình 3.6 Màn hình một số chức năng của chức năng ngủ khoa học 35](#_Toc69385495)

[Hình 3.7 Màn hình chức năng cá nhân hóa 36](#_Toc69385496)

[Hình 3.8 Ảnh chụp một số chức năng khác của giao diện cá nhân hóa 38](#_Toc69385497)

**MỤC LỤC BẢNG**

[Bảng 2.1 Yêu cầu từ phỏng vấn người dùng 18](#_Toc69388102)

[Bảng 2.2 Các chức năng dự tính 18](#_Toc69388103)

[Bảng 2.3 Kịch bản usecase xem sự kiện 22](#_Toc69388104)

[Bảng 2.4 Kịch bản use case thêm sự kiện 23](#_Toc69388105)

[Bảng 2.5 Kịch bản use case sửa sự kiện 24](#_Toc69388106)

[Bảng 2.6 Kịch bản use case xóa sự kiện 24](#_Toc69388107)

[Bảng 2.7 Kịch bản use case xem báo thức 25](#_Toc69388108)

[Bảng 2.8 Kịch bản use case sửa báo thức 25](#_Toc69388109)

[Bảng 2.9 Kịch bản use case thêm báo thức 26](#_Toc69388110)

[Bảng 2.10 Kịch bản use case xóa báo thức 27](#_Toc69388111)

[Bảng 2.11 Kịch bản use case sử dụng api của SVO để đăng nhập và lấy lịch học 27](#_Toc69388112)

[Bảng 2.12 Mô tả chi tiết bảng trong cơ sở dữ liệu 32](#_Toc69388113)

###### CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

# Giới thiệu tổng quan về đề tài

## Tên đề tài

“Xây dựng ứng dụng Quản lý thời gian”

## Lý do chọn đề tài

Là một sinh viên, quản lý thời gian một cách khoa học và hợp lý sẽ giúp cân bằng được giữa việc học cùng những sinh hoạt cá nhân, hoạt động ngoại khóa, giải trí v,v..

Cộng với điện thoại di động hiện tại có thể gọi là vật bất-ly-thân của mỗi người, đặc biệt là người trẻ và còn càng đúng với sinh viên công nghệ.

Dựa vào vấn đề bản thân nhận thấy lối sống hiện tại chưa tích cực, chưa tận dụng được thời gian để học tập và làm việc hiệu quả, mặc dù trên chợ ứng dụng Google Play và Appstore đã có rất nhiều ứng dụng quản lý thời gian nhưng khá chung chung, chưa phù hợp với người trẻ, đặc biệt với sinh viên.

Nhờ được sự giúp đỡ của cô …. trong bước đầu em đã nghiên cứu và vận dụng kiến thức đã học để xây dựng đề tài **“Xây dựng phần mềm quản lý thời gian cho sinh viên trên hệ điều hành android”**  đã đáp ứng tương đối một số vấn đề đã nêu ở trên, Tuy nhiên do kiến thức còn hạn chế nên đề tài chắc chắn không tránh khỏi các thiếu sót, Vì vậy em rất mong các đóng góp từ Thầy Cô và các bạn để đề tài ngày càng hoàn thiện và hiệu quả hơn.

Em xin chân thành cảm ơn các Thầy Cô trong khoa CNTT đã tạo điều kiện cho em hoàn thành đề tài thực tập này.

## Mục đích nghiên cứu

Tìm hiểu về Lập trình Android

Tìm hiểu về ngôn ngữ Java,XML

Tìm hiểu về Java,XML liên quan thế nào tới ứng dụng android

Thực hành vận dụng phân tích hệ thống hướng đối tượng UML

Xây dựng phần mềm quản lý thời gian (Time Wizard)

## Tổng quan về lập trình trên hệ điều hành android

### Hệ điều hành Android là gì?

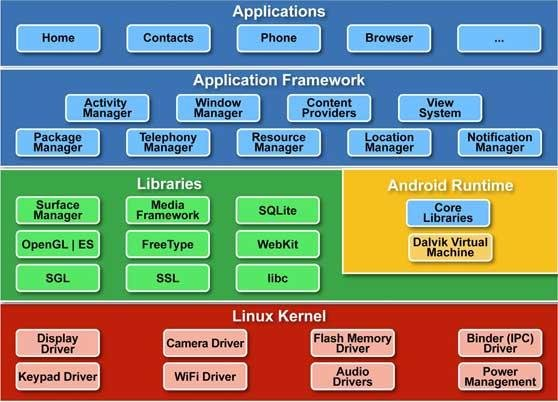
**Android** là một hệ điều hành có dạng mã nguồn mở, nó hoạt động dựa trên nền tảng Linux và được thiết kế dành riêng cho những thiết bị di động cảm ứng hoặc máy tính bảng. Trước đây, hệ điều hành này được phát triển bởi tổng công ty Android và được tài trợ bởi Google. Cho đến năm 2005 thì Google đã 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.

### Tìm hiểu về kiến trúc của Android



##### Hình 1.1 Cấu trúc hệ điều hành android

* + Linux Kernel: Đây là một loại nhân xử lý, nó có khả năng cung cấp độ trừu tượng cho các phần ứng.
  + Thư viện nguồn và các thư viện Android: Hầu hết, các thư viện nằm trên lớp nhân Linux và các thư viện này đều dựa vào Java để có thể phục vụ cho Android.
  + Android Runtime: Có khả năng cung cấp cho 1 bộ phận quan trọng nhất là Dalvik Virtual Machine (nó là một loại Java Virtual Machine) được các chuyên gia thiết kế đặc biệt với mục đích tối ưu cho Android.
  + Application Framework: Nó cung cấp các dịch vụ cao hơn cho những ứng dụng dưới dạng lớp Java. Từ đó, các Developer sẽ có quyền can thiệp vào từ lớp Android Framework này.
  + Application: Đây là nơi các lập trình viên thường xuyên làm việc cùng để có thể triển khai cho ứng dụng.

### Các phiên bản của hệ điều hành Android

Trải qua khoảng thời gian dài phát triển thì hệ điều hành Android đã trải qua rất nhiều phiên bản khác nhau. Nổi bật nhất là cập nhật kể từ version chính thức 1.5 và được gọi là “Cupcake” ra đời năm 2009. Tiếp đó, hệ điều hành Android đã được cập nhật thường xuyên hơn với version 10 vào đầu năm 2019.

### Ngôn ngữ dùng để lập trình Android

* Java
* Kotlin

### Một số ngôn ngữ được sử dụng để lập trình Android

Hiện nay, hệ điều hành Android sử dụng một số loại ngôn ngữ lập trình khác nhau như: Java, 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.

### Ngôn ngữ Java trong hệ điều hành Android

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…

## Tổng quan về ngôn ngữ java

### Giống như C++, hướng đối tượng hoàn toàn

Trong quá trình tạo ra 1 ngôn ngữ mới phục vụ cho mục đích chạy được trên nhiều nền tảng, các kỹ sư của Sun MicroSystem muốn tạo ra 1 ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Vì vậy họ đã sử dụng lại các cú pháp của C và C++.

Tuy nhiên, trong Java thao tác với con trỏ bị lược bỏ nhằm đảm bảo tính an toàn và dễ sử dụng hơn. Các thao tác overload, goto hay các cấu trúc như struct và union cũng được loại bỏ khỏi Java. Các vấn đề này xin được làm rõ ở những bài viết sau.

### Độc lập phần cứng và hệ điều hành

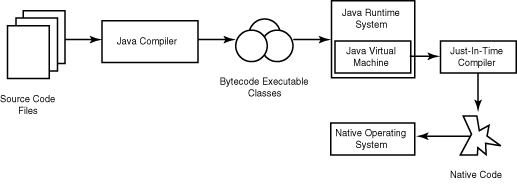
1 chương trình viết bằng ngôn ngữ Java có thể chạy tốt ở nhiều môi trường khác nhau. Gọi là khả năng cross-platform. Khả năng độc lập phần cứng và hệ điều hành được thể hiện ở 2 cấp độ là cấp độ mã nguồn và cấp độ nhị phân.

* Ở cấp độ mã nguồn: kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng 1 bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.
* Ở cấp độ nhị phân: 1 mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có Java Virtual Machine để thông dịch đoạn mã này.

### Ngôn ngữ thông dịch

Ngôn ngữ lập trình thường được chia ra làm 2 loại, tùy theo cách hiện thực hóa ngôn ngữ đó, là ngôn ngữ thông dịch và ngôn ngữ biên dịch. Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch, cụ thể như sau:

1. Khi viết mã trên 1 file .java, khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra Bytecode.
2. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã bytecode này thành machine code (hay native code) khi nhận được yêu cầu thực thi chương trình.



##### Hình 1.2 Mô hình thông dịch android

.java -> Bytecode -> Machine code

#### **Ưu điểm**

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

#### **Nhược điểm**

Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong 1 mức chấp nhận được).

### Cơ chế thu gom rác tự động

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

Với ngôn ngữ như C/C++, phải yêu cầu hủy vùng nhớ mà đã cấp phát để tránh việc thất thoát vùng nhớ. Tuy nhiên không phải lúc nào cũng có thể làm được điều đó như sơ sót hoặc kiến trúc đang code không cho phép dẫn đến việc thất thoát và làm giảm hiệu năng chương trình.

Java hỗ trợ điều đó, không phải hủy các vùng nhớ thủ công. Bộ thu dọn rác của Java sẽ theo vết các tài nguyên đã được cấp. Khi không có tham chiếu nào đến vùng nhớ, bộ thu dọn rác sẽ tiến hành thu hồi vùng nhớ đã được cấp phát theo định kỳ.

### Đa luồng

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

### Tính an toàn và bảo mật

#### Tính an toàn

Ngôn ngữ lập trình Java yêu cầu chặt chẽ về kiểu dữ liệu:

* Dữ liệu phải được khai báo tường minh.
* Không sử dụng con trỏ và các phép toán với con trỏ.
* Java kiểm soát chặt chẽ việc truy xuất mảng, chuỗi, không cho phép sử dụng các kỹ thuật tràn, do đó các truy xuất sẽ không vượt quá kích thước của mảng hoặc chuỗi.
* Quá trình cấp phát và giải phóng bộ nhớ được thực hiện tự động.
* Cơ chế xử lý lỗi giúp việc xử lý và phục hồi lỗi dễ dàng hơn.

#### Tính bảo mật

Java cung cấp 1 môi trường quản lý chương trình với nhiều mức khác nhau:

* Mức 1: chỉ có thể truy xuất dữ liệu cũng như phương phức thông qua giao diện mà lớp cung cấp.
* Mức 2: trình biên dịch kiểm soát các đoạn mã sao cho tuân thủ các quy tắc của ngôn ngữ lập trình Java trước khi thông dịch.
* Mức 3: trình thông dịch sẽ kiểm tra mã byte code xem các đoạn mã này có đảm bảo được các quy định, quy tắc trước khi thực thi.
* Mức 4: Java kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

### Java Virtual Machine

JVM (Java Virtual Machine) có một số đặc điểm cơ bản như sau

* Máy ảo Java là phần mềm giả lập máy tính, nó tập hợp các lệnh logic để xác định hoạt động của máy.
* Có thể xem nó như là 1 hệ điều hành thu nhỏ.
* JVM chuyển mã bytecode thành machine code tùy theo môi trường tương ứng (gọi là khả năng khả chuyển).
* JVM cung cấp môi trường thực thi cho chương trình Java (gọi đó là khả năng độc lập với nền).
* Sun MicroSystem chịu trách nhiệm thiết kế, phát triển các máy ảo Java chạy trên các hệ điều hành cũng như kiến trúc phần cứng khác nhau, điều này cho thấy có khá nhiều loại máy ảo Java.

## Tổng quan về ngôn ngữ đánh dấu XML

XML (viết tắt từ tiếng Anh eXtensible Markup Language, "Ngôn ngữ Đánh dấu khả mở") là ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị, để tạo ra các ngôn ngữ đánh dấu khác. Đây là một tập con đơn giản của SGML, có khả năng mô tả nhiều loại dữ liệu khác nhau. Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet. Các ngôn ngữ dựa trên XML (thí dụ: RDF, RSS, MathML, XHTML, SVG, GML và cXML) được định nghĩa theo cách thông thường, cho phép các chương trình sửa đổi và kiểm tra hợp lệ bằng các ngôn ngữ này mà không cần có hiểu biết trước về hình thức của chúng.

XML là ngôn ngữ xây dựng cấu trúc tài liệu văn bản, dựa theo chuẩn SGML (Standard Generalized Markup Language: siêu ngôn ngữ có khả năng sinh ngôn ngữ khác). SGML được phát triển cho việc định cấu trúc và nội dung tài liệu điện tử, do tổ chức ISO (International Organization for Standards) chuẩn hoá năm 1986.

SGML là do IBM đưa ra, song không thể không kể đến những đóng góp của các công ty khác. XML được W3C (World Wide Web Consortium: tổ chức độc lập định ra tiêu chuẩn cho trình duyệt Web, máy chủ và ngôn ngữ) phát triển, nhưng đặc tả XML lại do Netscape, Microsoft và các thành viên của dự án Text Encoding Initiative (TEI) xây dựng. Tổ chức W3C XML Special Interest Group có đại diện từ hơn 100 công ty cùng nhiều chuyên gia được mời khác.

## Công cụ cần thiết cho lập trình android

### Giới thiệu về android studio

Android Studio (ide android) là môi trường phát triển tích hợp chuyên nghiệp hỗ trợ việc thiết kế và xây dựng các ứng dụng di động trên nền tảng Android được Google và JetBrains hợp tác xây dựng nhằm thay thế các phiên bản plugin android dành cho Eclipse ngày xưa.

Trong nhiều năm trước, trước khi plugin phát triển android của Eclipse không còn được hỗ trợ của google, luôn có 2 luồng ý kiến trái chiều về việc lựa chọn IDE dành cho Android giữa Android Studio và Eclipse Android Plugin. Trong khi Android Studio được dựa trên nền tảng của IDE chuyên nghiệp IntelliJ IDEA dành cho JAVA của công ty JetBrains và được Google tùy chỉnh lại cho phù hợp với nền tảng Android thì Eclipse chỉ là một plugin tổng hợp các công cụ phát triển riêng trong bộ SDK mà Google cung cấp. Là một IDE mạnh mẽ và đa năng lại mã nguồn mở, trong những ngày đầu tiên của giai đoạn bùng nổ ứng dụng Android, Google cần dùng một IDE nào đó để cạnh tranh trực tiếp với XCode của IOS, vào lúc đó Eclipse là ứng cử viên sáng giá nhất. Vào những ngày đó cả hai ứng dụng Android Studio và Eclipse Plugin đều mạnh mẽ và đều có thể phát triển các ứng dụng Android từ cơ bản đến phức tạp, từ những ứng dụng giải trí đến các ứng dụng doanh nghiệp Enterprise.

Tuy nhiên, vào ngày 26 tháng 6 năm 2015, Google đã lên một lộ trình chuyển giao và thông báo rằng Android Studio sẽ chính thức bỏ hỗ trợ cho Plugin Android (ADT) cho IDE Eclipse chức năng mà nó cần để tạo các ứng dụng Android. Android Studio sẽ là IDE chính thức cho phát triển Android, cho phép Google tập trung phát triển trên một nền tảng thống nhất giữa các cấu trúc android project.

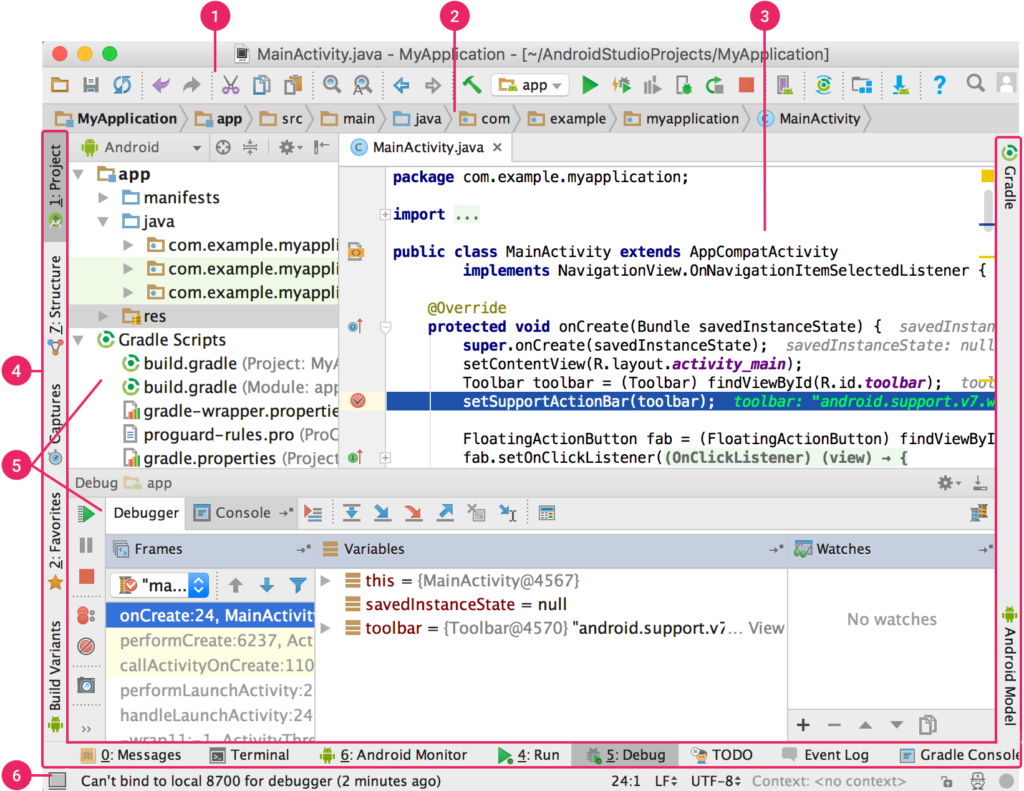
### Những tính năng chính của Android Studio:

Android Studio là Môi trường phát triển tích hợp chính thức (IDE) dành cho phát triển ứng dụng Android, dựa trên IntelliJ IDEA . Trên trình soạn thảo mã và công cụ phát triển mạnh mẽ của IntelliJ, Android Studio cung cấp nhiều tính năng nâng cao hiệu suất của bạn khi xây dựng ứng dụng Android, chẳng hạn như:

* Một hệ thống xây dựng Gradle linh hoạt
* Trình mô phỏng nhanh và tính năng phong phú
* Một môi trường hợp nhất nơi bạn có thể phát triển cho tất cả các thiết bị Android
* Instant Run để đẩy các thay đổi vào ứng dụng đang chạy của bạn mà không cần xây dựng một APK mới
* Mẫu mã và tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến và nhập mã mẫu
* Các công cụ và khuôn khổ thử nghiệm mở rộng
* Lint công cụ để bắt hiệu suất, khả năng sử dụng, tương thích phiên bản, và các vấn đề khác
* Hỗ trợ C ++ và NDK
* Tích hợp hỗ trợ Google Cloud Platform , giúp dễ dàng tích hợp Google Cloud Messaging và App Engine
* Hỗ trợ tích hợp sâu Firebase vào trong các ứng dụng chỉ sau một click chuột.

Giao diện thiết kế được bố trí hợp lý

Hỗ trợ kéo thả khi thiết kế giao diện UI cho app, thời gian phản hồi nhanh, đáp ứng nhanh hơn so Eclipse. Android Studio luôn luôn được hỗ trợ và cải tiến từ Google và JetBrains.



##### Hình 1.3 Giao diện người dùng của Android Studio

* Toolbar: Tổng hợp các công cụ cho phép bạn thực hiện một loạt các hành động, bao gồm cả chạy ứng dụng của bạn và liên kết hiển thị các tool trong Android SDK.
* Breadcrumb Bar: Thanh điều hướng giúp bạn điều hướng qua dự án của bạn và mở các tệp để chỉnh sửa. Nó cung cấp một cái nhìn nhỏ gọn hơn về cấu trúc có thể nhìn thấy được trong cửa sổ Project .
* Editor Window: Cửa sổ soạn thảo là nơi bạn tạo và sửa đổi mã. Tùy thuộc vào loại tệp hiện tại, trình soạn thảo có thể thay đổi. Ví dụ: khi xem tệp bố trí, trình soạn thảo sẽ hiển thị giao diện thiết kế ứng dụng Android trực quan.
* Tool Window bar: các liên kết đến các công cụ tích hợp trong môi trương phát triển như: các biến môi trường, cấu hình build project, thông tin profiles.
* Tool windows: Tập hợp các công cụ phân tích, điều hướng, gỡ rối và báo cáo khi phân tích, debug các ứng dụng đang phát triển.
* Status bar: Hiển thị thông tín báo cáo trạng thái các phân tích về tài nguyên của IDE Android Studio khi chạy.

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

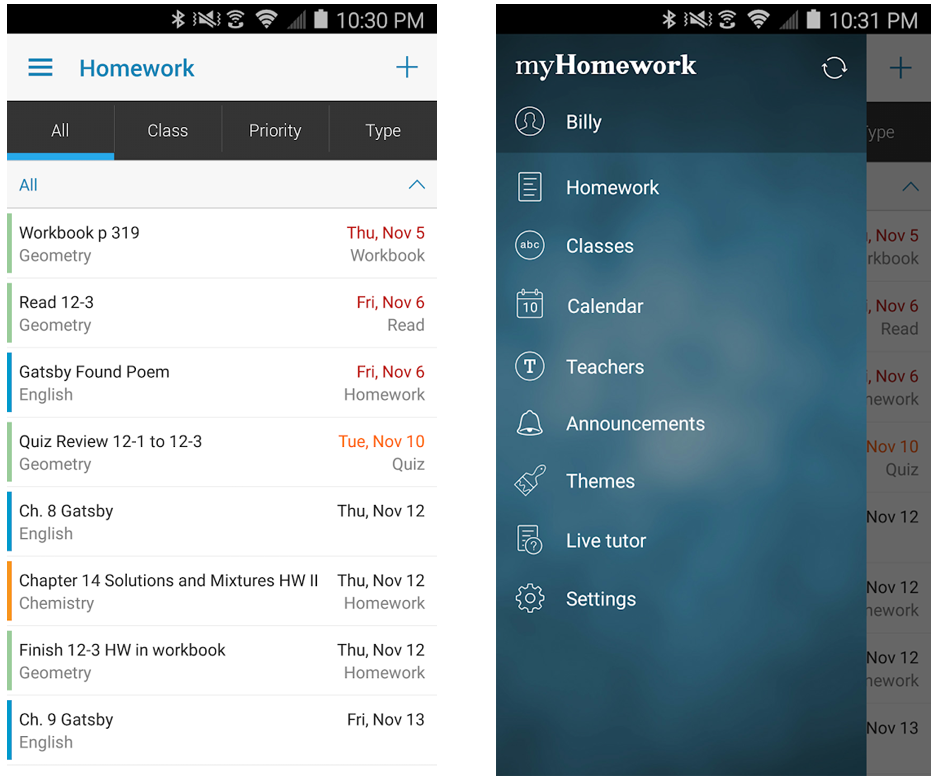
# Phân tích thiết kế hệ thống

## Khảo sát một số ứng dụng hiện có

### myHomework Student Planner

Cái tên có thể khiến người ta liên tưởng đến homework (bài tập về nhà) và có suy nghĩ rằng ứng dụng này sẽ làm sinh viên trở nên bận bịu hơn, nhưng trên thực tế thì đây là ứng dụng rất tuyệt vời chop sinh viên ở mọi bậc học, từ phổ thông đến đại học. myHomework Student Planner cung cấp chop sinh viên lịch biểu để cá nhân có thể dễ dàng theo dõi deadline bài tập, kỳ thi, các dự án và các sự kiện quan trọng. Đặc biệt ở chỗ là ứng dụng này chop phép sinh viên đồng bộ lịch biểu cá nhân để được nhắc nhở mỗi khi có sự kiện hoặc deadline sắp tới, tránh việc quên và để lỡ mất các công việc quan trọng.

Số tiền để sở hữu ứng dụng này là hoàn toàn miễn phí, tuy nhiên trong trường hợp người dùng muốn loại bỏ quảng cáo trong ứng dụng thì sẽ cần chi một số tiền là 4.99 USD (tương đương 114,000 VND) cho một năm sử dụng.



##### Hình 2.1 ứng dụng myHomework Student Planner

Ưu điểm:

-Phù hợp cho học sinh, sinh viên có thể đồng bộ lịch học

+Nhược điểm:

-Không có tiêng việt.

-Không thể đồng bộ lịch học của DHTN

-Phải trả phí nếu không muốn có quảng cáo

### TimeTable- Weekly Schedule/Planner

Là ứng dụng được viết bởi người Việt Nam, Ứng dụng này quản lý thời gian của bạn bằng cách tạo thời gian biểu, lịch học, thời khóa biểu, một cách dễ dàng.

Ưu điểm:

- Giao diện người dùng đơn giản, dễ nắm bắt, sử dụng

- Ứng dụng cho phép tùy biến giao diện, chức năng linh hoạt

- Quản lý lịch học bằng thời khóa biểu một cách hiệu quả

Nhược điểm:

-Không thể thêm lịch học tự động

-Đơn giản đẫn đến thiếu một số chức năng

|  |  |  |
| --- | --- | --- |
|  |  |  |

##### Hình 2.2 Ứng dụng TimeTable- Weekly Schedule/Planner

## Yêu cầu từ phỏng vấn người dùng

|  |  |
| --- | --- |
|  | Hệ thống cho phép người dùng tạo thời gian biểu |
|  | Hệ thống cho phép người dùng quản lý (xem sửa xóa) thời gian biểu đã tạo |
|  | Hệ thống có thể tự động tạo báo thức, nhắc nhở khi đến thời gian có sự kiện |
|  | Hệ thống có thể tự cập nhật lịch học của người dùng nếu người dùng đã đăng nhập tài khoản sinh viên |
|  | Hệ thống có thể tính toán giấc ngủ khoa học theo yêu cầu của người dùng |
|  | Hệ thống có thể tự động tạo báo thức và nhắc nhở người sử dụng đi ngủ để đảm bảo giấc ngủ đủ giấc và thức dậy đúng thời gian chuẩn bị đi học |
|  | Hệ thống có thể hiển thị thông báo khi cần thiết |
|  | Hệ thống cho phép thông báo đến giờ học trước thời gian được cài đặt |
|  | Hệ thống cho phép bỏ qua sự kiện trước nếu sự kiện không diễn ra  (ví dụ như hôm đó không phải học vì GV bận) |

##### Bảng 2.1 Yêu cầu từ phỏng vấn người dùng

|  |  |  |
| --- | --- | --- |
| **STT** | **NHÓM CHỨC NĂNG** | **MỤC ĐÍCH** |
| 1 | Cấu hình hệ thống | cho phép người sử dụng cấu hình thông số hệ thống |
| 2 | Quản lý sự kiện | Cho phép người dùng  quản lý các thông tin về sự kiện đã ,sẽ và đang diễn ra |
| 3 | Quản lý giấc ngủ thông minh | cho phép người dùng tinh chỉnh các thông số của giấc ngủ thông minh |
| 4 | Quản lý tài khoản sinh viên | Cho phép người dùng chỉnh sửa tài khoản sinh viên để nhận được lịch học |

##### Bảng 2.2 Các chức năng dự tính

**Các tính năng dự tính**

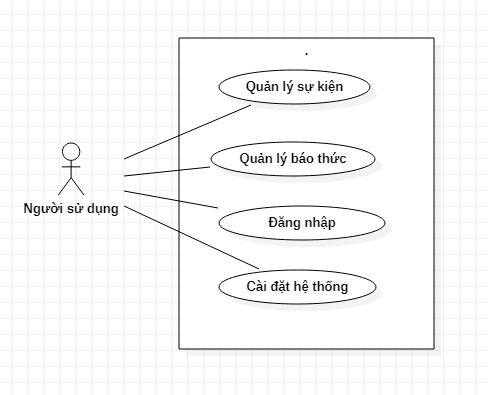
+Tạo thời gian biểu

+Xem sửa xóa thời gian biểu

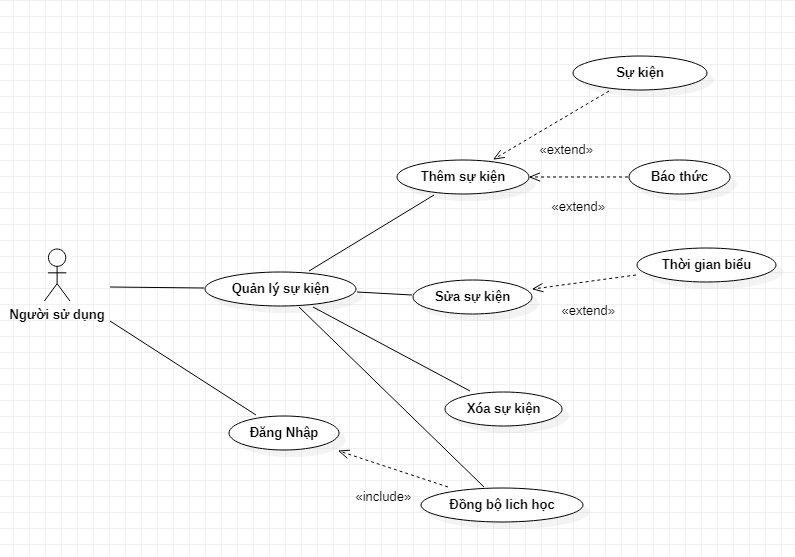
+Tải và đồng bộ lịch học vào thời gian biểu

+Tính toán giấc ngủ và đặt báo thức

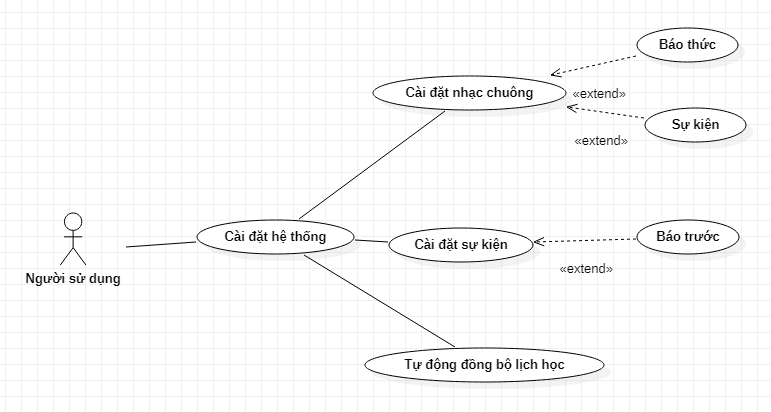
## Usecase Hệ thống



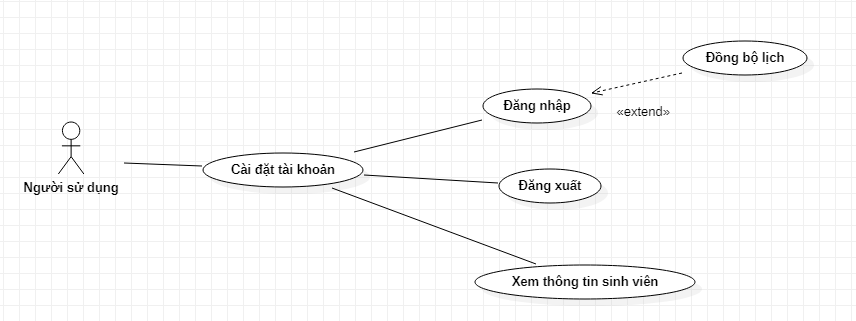
##### Hình 2.3 Usecase tổng quát hệ thống



##### Hình 2.4 Usecase quản lý sự kiện



##### Hình 2.5 Usecase Cài đặt hệ thống



##### Hình 2.6 Usecase cài đặt tài khoản

## Kịch bản

### Kịch bản use case xem sự kiện

|  |  |
| --- | --- |
| Tên usecase | Xem sự kiện |
| Tác nhân | Người sử dụng |
| Tiền điều kiện |  |
| Kết quả | Hệ thống hiển thị sự kiện của thời gian biểu |
| Kích hoạt | Người dùng chọn vào tab “thời gian biểu” ở dưới cùng màn hình |
| Chuỗi sự kiện chính:   1. Người dùng chọn chức năng xem thời gian biểu ở dưới cùng màn hình 2. Hệ thống Kiểm tra xem cơ sở dữ liệu có sự kiện nào không 3. Hệ thống xử lý sự kiện rồi hiển thị ra màn hình theo ngày từ cũ đến mới | |
| Ngoại lệ:  2.1 Hệ thống kiểm tra không có bản ghi sự kiện nào ở trong cơ sở dữ liệu  2.2 Hệ thống thông báo với người dùng là hiện tại không có bản ghi sự kiện nào trong cơ sở dữ liệu và gợi ý thêm mới | |

##### Bảng 2.3 Kịch bản usecase xem sự kiện

### Kịch bản use case thêm sự kiện

|  |  |
| --- | --- |
| Tên usecase | Thêm sự kiện |
| Tác nhân | Người sử dụng |
| Tiền điều kiện | Người dùng đang ở tab  “Thời gian biểu” |
| Kết quả | Hệ thống thêm thành công bản ghi sự kiện vào cơ sở dữ liệu |
| Kích hoạt | Người dùng chọn vào nút “+” ở trên cùng bên phải màn hình |
| Chuỗi sự kiện chính:   1. Người dùng chọn nút “+” để thêm bản ghi sự kiện mới 2. Hệ thống hiển thị giờ để người dùng chọn cho sự kiện 3. Hệ thống hiển thị lịch để người dùng chọn 4. Hệ thống yêu cầu nhập tiêu đề và mô tả cho sự kiện 5. Người dùng ấn nút hoàn tất 6. Hệ thống kiểm tra thông tin người dùng nhập vào rồi thêm vào cơ sở dữ liệu 7. Hệ thống cập nhật sự kiện người dùng vừa nhập lên | |
| Ngoại lệ:  2.1 Người dùng nhấn nút “<<” để hủy sự kiện  3.1 Người dùng nhấn nút “<<” để quay về chọn giờ cho sự kiện  4.1 Người dùng nhấn nút “<<” để quay lại chọn ngày cho sự kiện  6.1 Trong trường hợp người dùng để trống trường “tiêu đề” hoặc “mô tả” hệ thống sẽ ghi vào cơ sở dữ liệu là “không có tiêu đề” hoặc “không có mô tả” | |

##### Bảng 2.4 Kịch bản use case thêm sự kiện

### Kịch bản use case sửa sự kiện

|  |  |
| --- | --- |
| Tên usecase | Sửa sự kiện |
| Tác nhân | Người sử dụng |
| Tiền điều kiện | Người dùng đang ở tab  “Thời gian biểu” |
| Kết quả | Người dùng sửa thành công sự kiện đã có sẵn |
| Kích hoạt | Người dùng chọn sự kiện rồi ấn vào sự kiện cần sửa |
| Chuỗi sự kiện chính:   1. Người dùng ấn vào sự kiện cần sửa 2. Hệ thống hiển thị giờ cũ để người dùng sửa cho sự kiện 3. Hệ thống hiển thị lịch cũ để người dùng sửa 4. Hệ thống hiển thị tiêu đề và mô tả lịch cũ cho người dùng sửa 5. Người dùng ấn nút hoàn tất 6. Hệ thống kiểm tra thông tin người dùng nhập vào rồi cập nhật vào cơ sở dữ liệu 7. Hệ thống cập nhật sự kiện người dùng vừa sửa lên | |
| Ngoại lệ:  2.1 Người dùng nhấn nút “<<” để hủy sửa sự kiện  3.1 Người dùng nhấn nút “<<” để quay về chọn giờ cho sự kiện  4.1 Người dùng nhấn nút “<<” để quay lại chọn ngày cho sự kiện  6.1 Trong trường hợp người dùng không nhập thêm gì hệ thống sẽ tự hiểu không có thay đổi trong 2 trường này. | |

##### Bảng 2.5 Kịch bản use case sửa sự kiện

### Kịch bản use case xóa sự kiện

|  |  |
| --- | --- |
| Tên usecase | Xóa sự kiện |
| Tác nhân | Người sử dụng |
| Tiền điều kiện | Người dùng đang ở tab  “Thời gian biểu” |
| Kết quả | Hệ thống xóa thành công bản ghi sự kiện ở cơ sở dữ liệu |
| Kích hoạt | Người dùng chọn vào sự kiện rồi nhấn giữ sự kiện đó |
| Chuỗi sự kiện chính:   1. Người dùng nhấn giữ sự kiện có sẵn 2. Hệ thống hiển thị yêu cầu xác nhận “có” hoặc “không“ 3. Người dùng chọn “có” 4. Hệ thống xóa sự kiện trong cơ sở dữ liệu và cập nhật lại danh sách sự kiện | |
| Ngoại lệ:  2.1 Người dùng nhấn nút “không” để hủy bỏ xóa sự kiện | |

##### Bảng 2.6 Kịch bản use case xóa sự kiện

### Kịch bản use case xem báo thức

|  |  |
| --- | --- |
| Tên usecase | Xem báo thức |
| Tác nhân | Người sử dụng |
| Tiền điều kiện |  |
| Kết quả | Hệ thống hiển thị báo thức đã được thêm |
| Kích hoạt | Người dùng chọn vào tab “ngủ khoa học” ở dưới cùng màn hình |
| Chuỗi sự kiện chính:   1. Người dùng chọn chức năng ngủ khoa học ở dưới cùng màn hình 2. Hệ thống Kiểm tra xem cơ sở dữ liệu có báo thức nào không 3. Hệ thống xử lý báo thức rồi hiển thị ra màn hình | |
| Ngoại lệ:  2.1 Hệ thống kiểm tra không có bản ghi báo thức nào ở trong cơ sở dữ liệu  2.2 Hệ thống thông báo với người dùng là hiện tại không có bản ghi báo thức nào trong cơ sở dữ liệu và gợi ý thêm mới | |

##### Bảng 2.7 Kịch bản use case xem báo thức

### Kịch bản use case sửa báo thức

|  |  |
| --- | --- |
| Tên usecase | Sửa báo thức |
| Tác nhân | Người sử dụng |
| Tiền điều kiện | Người dùng đã nhấn vào tab “ngủ khoa học” |
| Kết quả | Hệ thống cập nhật lại thông tin của báo thức đã chọn |
| Kích hoạt | Người dùng chọn vào báo thức cần sửa |
| Chuỗi sự kiện chính:   1. Người dùng chọn báo thức cần sửa bằng cách chạm vào báo thức 2. Hệ thống hiển thị giao diện yêu cầu người dùng chọn lại giờ cho báo thức 3. Hệ thống hiển thị mô tả của báo thức cũ và yêu cầu người dùng nhập 4. Người dùng ấn nút kết thúc 5. Hệ thống cập nhật báo thức đã sửa | |
| Ngoại lệ:  2.1 Người dùng ấn nút “<<” để hủy sửa báo thức | |

##### Bảng 2.8 Kịch bản use case sửa báo thức

### Kịch bản use case thêm báo thức

|  |  |
| --- | --- |
| Tên usecase | Thêm báo thức |
| Tác nhân | Người sử dụng |
| Tiền điều kiện | Người dùng đang ở tab  “ngủ khoa học” |
| Kết quả | Hệ thống thêm thành công bản ghi báo thức vào cơ sở dữ liệu |
| Kích hoạt | Người dùng chọn vào nút “+” ở trên cùng bên phải màn hình |
| Chuỗi sự kiện chính:   1. Người dùng chọn nút “+” để thêm bản ghi báo thức mới 2. Hệ thống hiển thị giờ để người dùng chọn cho báo thức 3. Hệ thống yêu cầu nhập tiêu đề và mô tả cho báo thức 4. Người dùng ấn nút hoàn tất 5. Hệ thống kiểm tra thông tin người dùng nhập vào rồi thêm vào cơ sở dữ liệu 6. Hệ thống cập nhật báo thức người dùng vừa nhập lên | |
| Ngoại lệ:  2.1 Người dùng nhấn nút “<<” để hủy tạo báo thức  3.1 Người dùng nhấn nút “<<” để quay về chọn giờ cho báo thức  6.1 Trong trường hợp người dùng để trống trường “tiêu đề” hoặc “mô tả” hệ thống sẽ ghi vào cơ sở dữ liệu là “không có tiêu đề” hoặc “không có mô tả” | |

##### Bảng 2.9 Kịch bản use case thêm báo thức

### Kịch bản use case xóa báo thức

|  |  |
| --- | --- |
| Tên usecase | Xóa báo thức |
| Tác nhân | Người sử dụng |
| Tiền điều kiện | Người dùng đang ở tab  “Ngủ khoa học” |
| Kết quả | Hệ thống xóa thành công bản ghi báo thức ở cơ sở dữ liệu |
| Kích hoạt | Người dùng chọn vào báo thức rồi nhấn giữ báo thức đó |
| Chuỗi sự kiện chính:   1. Người dùng nhấn giữ báo thức có sẵn 2. Hệ thống hiển thị yêu cầu xác nhận “có” hoặc “không“ 3. Người dùng chọn “có” 4. Hệ thống xóa sự kiện trong cơ sở dữ liệu và cập nhật lại danh sách báo thức | |
| Ngoại lệ:  2.1 Người dùng nhấn nút “không” để hủy bỏ xóa báo thức | |

##### Bảng 2.10 Kịch bản use case xóa báo thức

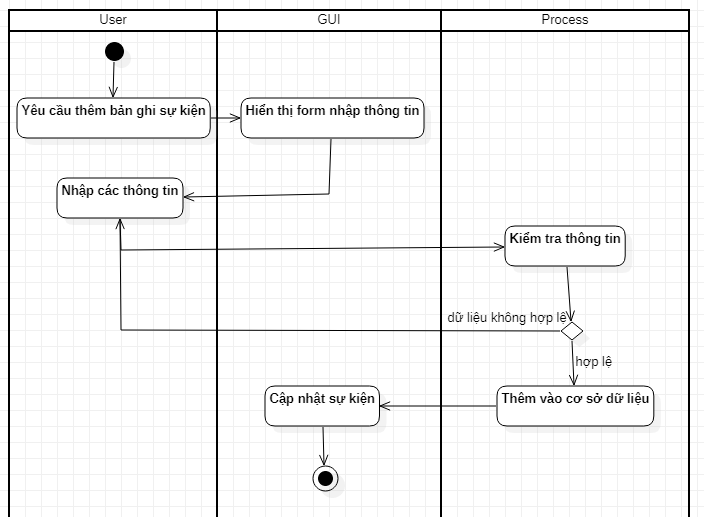
### Kịch bản use case sử dụng api của SVO để đăng nhập và lấy lịch học

|  |  |
| --- | --- |
| Tên usecase | Sử dụng api của SVO để đăng nhập và lấy lịch học |
| Tác nhân | Người sử dụng |
| Tiền điều kiện | Người dùng đang ở tab  “cá nhân hóa” |
| Kết quả | Hệ thống thêm lịch học được lấy từ SVO và thêm vào cơ sở dữ liệu |
| Kích hoạt | Người dùng chọn chức năng lấy lịch học từ SVO |
| Chuỗi sự kiện chính:   1. Người dùng yêu cầu sử dụng chức năng lấy lịch học từ SVO 2. Hệ thống yêu cầu người sử dụng nhập tài khoản(có thể là Mã sinh viên, số điện thoại, email…) và mật khẩu 3. Hệ thống xác thực dữ liệu người dùng 4. Hệ thống tiến hành kiểm tra tên tài khoản bằng SVO api 5. Hệ thống kiểm tra mật khẩu bằng SVO api 6. Hệ thống yêu cầu lịch học từ SVO api bằng token đã lấy được từ bước đăng nhập và thêm vào cơ sở dữ liệu | |
| Ngoại lệ:  3.1 Người dùng nhập tài khoản hoặc mật khẩu không đúng định dạng  3.2 Hệ thống yêu cầu người sử dụng nhập lại thông tin  4.1 SVO api trả về tên người dùng không hợp lệ  4.2 Hệ thống thông báo tên người dùng không chính xác  5.1 SVO api trả về mật khẩu người dùng không hợp lệ  5.2 Hệ thống thông báo mật khẩu người dùng không chính xác | |

##### Bảng 2.11 Kịch bản use case sử dụng api của SVO để đăng nhập và lấy lịch học

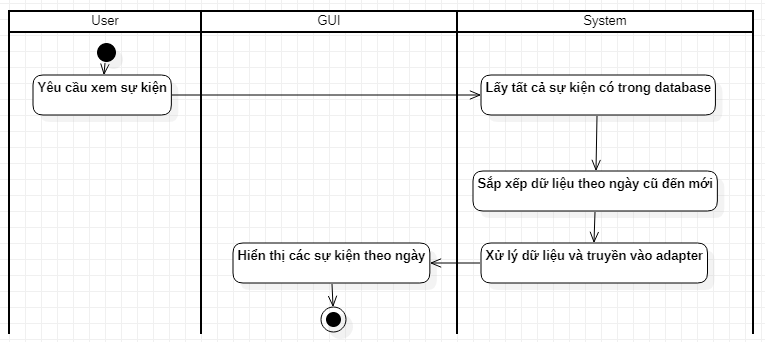
## Biểu đồ trạng thái (Activity diagram)

### Biểu đồ trạng thái cho use case thêm sự kiện



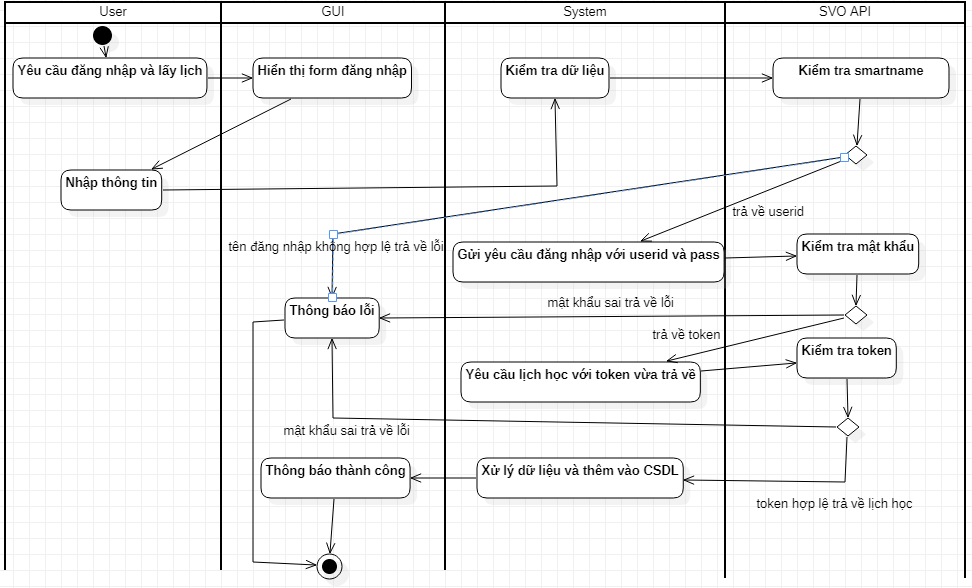
##### Hình 2.7 Biểu đồ trạng thái cho use case thêm sự kiện

### Biểu đồ trạng thái cho use case xem sự kiện



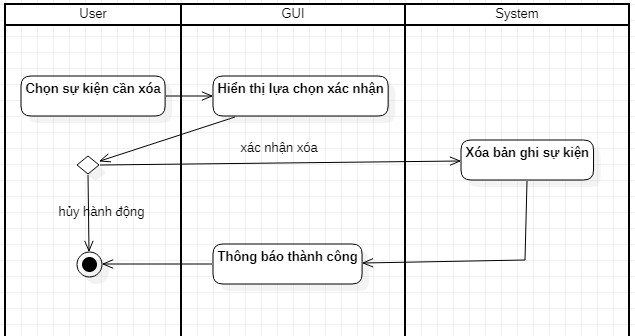
##### Hình 2.8 Biểu đồ trạng thái cho use case xem sự kiện

### Biểu đồ trạng thái cho use case sử dụng SVO api để lấy lịch học



##### Hình 2.9 Biểu đồ trạng thái cho use case sử dụng SVO api để lấy lịch học

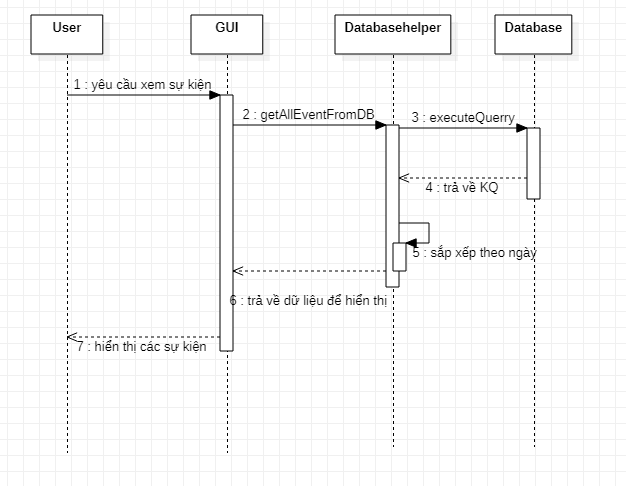
### Biểu đồ trạng thái cho use case xóa sự kiện



##### Hình 2.10 Biểu đồ trạng thái cho use case xóa sự kiện

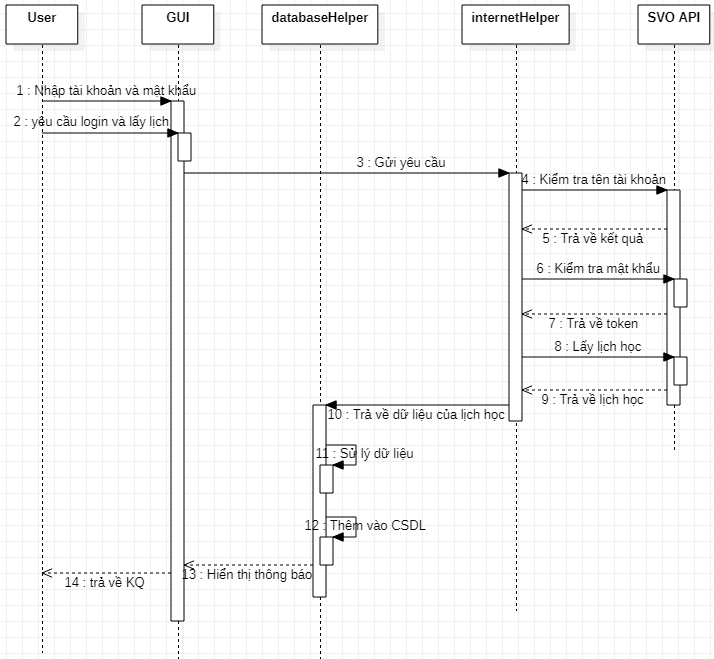
## Biểu đồ trình tự (Sequence diagram)

### Biểu đồ trình tự cho usecase xem sự kiện



##### Hình 2.11 Biểu đồ trình tự cho usecase xem sự kiện

### Biểu đồ trình tự cho usecase sử dụng SVO api để lấy lịch học



##### Hình 2.12 Biểu đồ trình tự cho usecase sử dụng SVO api để lấy lịch học

## Cơ sở dữ liệu

### Mô tả chi tiết bảng trong cơ sở dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Null** | **Ghi chú** |
| id | int |  | Khóa chính |
| title | String | x |  |
| description | String | x |  |
| startAt | DateTime |  |  |
| endAt | DateTime |  |  |
| type | String |  |  |
| note | String | x |  |
| alarmAt | DateTime |  |  |
| enable | int |  |  |

##### Bảng 2.12 Mô tả chi tiết bảng trong cơ sở dữ liệu

###### CHƯƠNG III. CÀI ĐẶT CHƯƠNG TRÌNH

# Cài đặt chương trình

## 1 Thiết kế giao diện.

Yêu cầu hệ thống: hệ điều hành android 8.0(API level 26) - Oreo trở lên

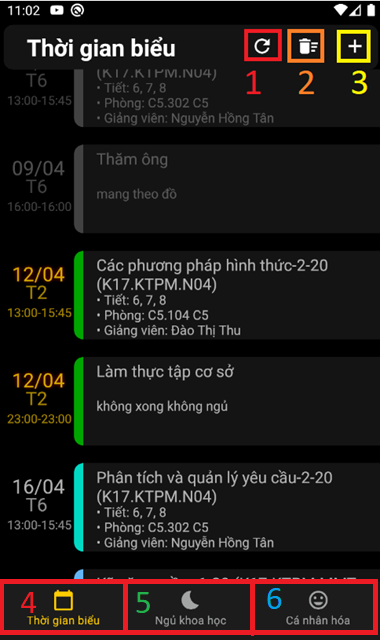
Bộ nhớ tối thiểu : 1.5gb ram

### Giao diện cơ bản khi mở ứng dụng



##### Hình 3.1 Giao diện cơ bản khi mở ứng dụng

### Một số chức năng



##### Hình 3.2 Một số chức năng

**Mô tả chức năng:**

* Kí hiệu số 1(đỏ): Cập nhật lịch học từ SVO.
* Kí hiệu số 2(cam): Xóa lịch học hoặc xóa tất cả lịch (nhấn 1 lần hoặc nhấn giữ).
* Kí hiệu số 3(vàng): Thêm sự kiện mới thủ công.
* Kí hiệu số 4(đỏ): Quản lý lịch biểu
* Kí hiệu số 5(xanh): Quản lý báo thức
* Kí hiệu số 6(xanh lục): Cài đặt & cá nhân hóa

### Giao diện khi chọn chức năng thêm sự kiện mới

|  |  |  |
| --- | --- | --- |
|  |  |  |

##### Hình 3.3 Giao diện khi chọn chức năng thêm sự kiện mới

### Màn hình chức năng sửa sự kiện có sẵn

|  |  |  |
| --- | --- | --- |
|  |  |  |

##### Hình 3.4 Giao diện chức năng sửa sự kiện có sẵn

### Giao diện khi kéo phải một sự kiện



**Mô tả chức năng:**

* Kí hiệu số 1(xanh): Sửa thông tin sự kiện.
* Kí hiệu số 2(đỏ): Xóa sự kiện hiện tại.

### Màn hình chính chức năng ngủ khoa học



##### Hình 3.5 Màn hình chính chức năng ngủ khoa học

### Màn hình một số chức năng của chức năng ngủ khoa học

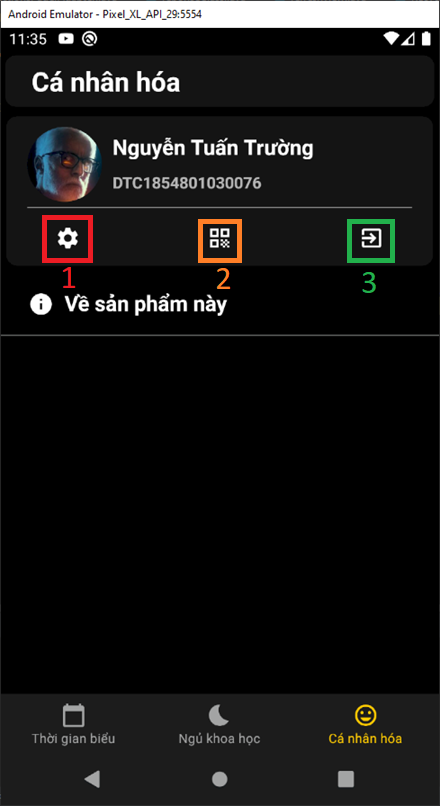
|  |  |
| --- | --- |
|  |  |

##### Hình 3.6 Màn hình một số chức năng của chức năng ngủ khoa học

**Mô tả chức năng:**

* Hình trái: hiển thị thông báo và phát nhạc.
* Hình phải: xóa báo thức.

### Màn hình chức năng cá nhân hóa

****

##### Hình 3.7 Màn hình chức năng cá nhân hóa

**Mô tả chức năng:**

* Kí hiệu số 1(đỏ): Cài đặt ứng dụng.
* Kí hiệu số 2(cam): Hiển thị mã QR để quét.
* Kí hiệu số 3(xanh): Đăng nhập nếu chưa đăng nhập hoặc đăng xuất nếu đã đăng nhập

### Ảnh chụp một số chức năng khác của giao diện cá nhân hóa

|  |  |  |
| --- | --- | --- |
|  |  |  |

##### Hình 3.8 Ảnh chụp một số chức năng khác của giao diện cá nhân hóa

**Mô tả chức năng:**

* Hình trái: Cài đặt hệ thống.
* Hình giữa: Hiển thị mã QR.
* Hình phải: Form đăng nhập.

###### KẾT LUẬN

Qua quá trình nghiên cứu, khảo sát, phân tích và xây dựng đề tài “Xây dựng ứng dụng quản lý thời gian”. Với sự cố gắng của bản thân, sự tận tình của thầy cô hướng dẫn và sự trao đổi của bạn bè, người thực hiện đề tài đã thực hiện được các chức năng cơ bản mà yêu cầu đề ra ban đầu. Qua quá trình thực hiện đề tài, người thực hiện cũng nâng cao được khả năng lập trình cũng như việc tự nghiên cứu, tìm hiểu các công nghệ, công cụ thông qua các tài liệu khác nhau. Người thực hiện đề tài đã thành công trong việc xây dựng một ứng Quản lý thời gian cho sinh viên thái nguyên nói riêng và học sinh, sinh viên cả nước nói chung.

* Kết quả đạt được

Qua thời gian tìm tìm hiểu về các công cụ lập trình và nghiên cứu nghiệp vụ người thực hiện đề tài đã đạt được kết quả sau:

* Lập trình được ứng dụng android quản lý thời gian
* Ứng dụng SQLite để quản lý dữ liệu
* Hoàn thành các chức năng đề ra để giải quyết vấn đề
* Sử dụng SVO API để cập nhật lịch học tự động

Bên cạnh những kết quả đạt được, vì thời gian không nhiều và kiến thức còn hạn chế nên đề tài còn một số nhược điểm

* Nhược điểm của đề tài

Đề tài còn có một số nhược điểm sau:

* Giao diện còn chưa thực sự bắt mắt và tối ưu
* Ứng dụng còn một số lỗi nhỏ tồn đọng.
* Hướng phát triển của đề tài
* Khắc phục các nhược điểm trên
* Tìm hiểu và cải thiện giao diện bắt mắt hơn.
* Ứng dụng firebase để lưu dữ liệu người dung.

###### TÀI LIỆU THAM KHẢO

[1] John Horton (2015), *Android Programming for Beginners.*

[2] Kathy Sierra, Bert Bates (2003), *Head First Java.*

[3] Tao Wang, Ryan Cohen (2014), *GUI Design for Android Apps.*

[4] Lê Văn Phùng (2013), Cơ Sở Dữ Liệu Quan Hệ Và Công Nghệ Phân Tích - Thiết Kế, Nhà xuất bản thông tin và truyền thông.

[5] *Github*

+ SwipeLayout – daimajia

<https://github.com/daimajia/AndroidSwipeLayout>

+ Gson – Google

<https://github.com/google/gson>

+ Okhttp - SquareUp

[https://square.github.io/okhttp/](https://square.github.io/okhttp/%20)

[6] *Youtube*

+ Coding in Flow

<https://www.youtube.com/channel/UC_Fh8kvtkVPkeihBs42jGcA>

+ Khoa phạm - Ứng dụng báo thức

<https://www.youtube.com/channel/UCX1g7Ciyjv6pFeX7EhvX4sQ>

+ Codestormx dev – LoginUI

<https://www.youtube.com/channel/UCMrc7COBma48LbxH8HZ10CA>