TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO THỰC TẬP TỐT NGHIỆP

(tại Doanh nghiệp)

Sinh viên thực hiện: NGUYỄN TUẤN TRƯỜNG

Lớp: KTPM-K17B

Người hướng dẫn tại doanh nghiệp: Nguyễn Anh Tú

Giáo viên quản lý: Nguyễn Toàn Thắng

Thái Nguyên – Năm 2022

LỜI CẢM ƠN

Trong suốt quá trình học tập, rèn luyện kỹ năng trên ghế nhà trường và sau thời gian 2 tháng thực tập thực tế tại Công ty cổ phần VCCORP . Mặc dù thời gian không dài nhưng đã giúp em học hỏi được nhiều kinh nghiệm thực tế hài hòa giữa kiến thức lý thuyết vào trong quá trình thực hành. Với sự giúp đỡ, quan tâm tận tụy của các thầy, cô giáo, cùng các anh chị nhân viên trong công ty đã giúp em học hỏi được nhiều kinh nghiệm quý báu khi ra trường.

Để có được những kiến thức quý báu đó, em xin gửi lời cảm ơn chân thành đến các thầy cô giáo Trường Đại học Công nghệ thông tin và truyền thông Thái Nguyên nói chung, các thầy cô giáo trong Bộ môn Công nghệ phần mềm nói riêng, giảng viên hướng dẫn – Thầy Nguyễn Toàn Thắng đã hết lòng tận tụy, chỉ bảo, hướng dẫn chúng em trong suốt quá trình học tập, gắn liền kiến thức lý thuyết giảng dạy, cũng như hiểu biết về thực tế trong chuyên ngành để khi chúng em ra trường nắm chắc được những kiến thức quan trọng, chuẩn bị cho tương lai sau này.

Bên cạnh đó, với sự giúp đỡ, chỉ bảo tận tình của Công ty cổ phần VCCORP nói chung, anh Nguyễn Anh Tú và các anh chị trong phòng IT nói riêng, đã chỉ bảo tận tình, tạo mọi điều kiện thuận lợi để em thực hiện tốt đợt thực tập vừa qua. Với lòng biết ơn chân thành, em xin chúc toàn thể anh chị trong công ty luôn có một sức khỏe tốt, một tâm thế chiến binh để có thể hoàn thành tốt mọi nhiệm vụ được giao.

Trong quá trình thực tập và làm báo cáo, do còn nhiều thiếu sót về mặt kinh nghiệm thực tế nên em rất mong nhận được sự đóng góp và chỉ bảo từ các thầy cô giáo để em có thể hoàn thành đợt thực tập một cách tốt hơn.

Em xin chân thành cảm ơn!

Thái Nguyên, ngày 10 tháng 10 năm 2022

Sinh viên

Nguyễn Tuấn Trường

LỊCH LÀM VIỆC TẠI NƠI THỰC TẬP

Tuần	Công việc	Người hướng dẫn	Mức độ hoàn thành	Nhận xét của người hướng dẫn
1	- Tìm hiểu về công ty, cách tổ chức của công ty. - Làm quen với các công cụ làm việc trong công ty. - Học cách trao đổi, báo cáo, làm việc qua telegram, notion.	Nguyễn Anh Tú		
2	- Tìm hiểu về code convention, SOLID, công cụ android studio	Nguyễn Anh Tú		
3	- Tìm hiểu về hệ điều hành Android, ngôn ngữ Java, XML	Nguyễn Anh Tú		
4	- Tìm hiểu về mô các mô hình MVC,MVP,MVVM	Nguyễn Anh Tú		
5	- Tìm hiểu về github và cách quản lý source code	Nguyễn Anh Tú		
6	- Xây dựng ứng dụng	Nguyễn Anh Tú		
7	- Tìm hiểu về cách triển khai ứng dụng trên Google Play	Nguyễn Anh Tú		
8	- Triển khai ứng dụng - Báo cáo cuối đợt thực tập	Nguyễn Anh Tú		

MỤC LỤC

CHƯƠNG I
GIỚI THIỆU VỀ TỔ CHỨC TẠI NƠI THỰC TẬP

1. Giới thiệu về công ty

Công ty cổ phần Vccorp là một doanh nghiệp đi đầu trong lĩnh vực nội dung số, công nghệ thông tin của Việt Nam. Với quãng thời gian trên 10 năm hình thành, phát triển trên thị trường, doanh nghiệp đã cống hiến rất nhiều sản phẩm công nghệ chất lượng cao dành cho người tiêu dùng trên toàn Việt Nam.

Hiện nay, 90% số lượng người dùng internet đều đã từng trải nghiệm qua các sản phẩm công nghệ của công ty cổ phần Vccorp. Đây chính là bằng chứng rõ nét nhất về sự đa dạng trong dịch vụ mà công ty đem lại trên thị trường.

Với thâm niên hoạt động trong lĩnh vực công nghệ, nội dung số trong 10 năm. Tập đoàn Vccorp đã đạt được những thành tích rất đáng nói bất chấp những khó khăn chung của cả thị trường. Trong đó phải kể đến một số thành tích như:

33 triệu người sử dụng các sản phẩm, dịch vụ của Vccorp tại thị trường Việt Nam.

2.8 tỷ lượt truy cập các trang tin của công ty Vccorp mỗi tháng.

1 tỷ lượt quảng cáo mỗi ngày được thực hiện thông qua hệ thống quảng cáo trực tuyến admicro vccorp.

16 triệu user sử dụng các thiết bị dịch vụ khác nhau trên điện thoại thông minh.

Ra mắt mạng xã hội của riêng mình với tên gọi **lotus.vn**

2. Lĩnh vực hoạt động

2.1. Hệ thống kênh thông tin

- **CafeF.vn**

  Kênh thông tin Kinh tế- tài chính Việt Nam CafeF.vn là chuyên trang dẫn đầu trong mảng tin tức kinh tế, tài chính, thông tin chứng khoán tại Việt Nam được cập nhật liên tục, đầy đủ và chính xác.

- **Kenh14.vn**

  Nội dung của Kênh 14 đa phần viết về chủ đề giải trí, xã hội, người của công chúng... hướng đến đối tượng chính là các độc giả trẻ như tuổi thanh thiếu niên, học sinh, sinh viên

- **Afamily.vn**

  Afamily là trang báo điện tử uy tín chuyên nghiệp chuyên cung cấp những thông tin thú vị và bổ ích về phụ nữ, làm đẹp, cuộc sống gia đình cùng các tin tức hot

- **Autopro.com.vn**

  Tin tức thị trường ô tô, xe máy trong nước và quốc tế, xe máy độ, video đua xe, kỹ thuật lái xe, đua xe, xe đua.

- **Gamek.vn**

  Kênh Game với thông tin mới nhất, đánh giá các Game hay đang chơi, công nghệ Game trong nước và thế giới cực nhanh, cùng thư viện trò chơi đầy đủ nhất

2.2. Hệ thống thương mại điện tử

- **Rongbay.com**

  Chuyên kênh rao vặt lớn nhất toàn quốc từ dịch vụ tới sản phẩm đa dạng

- **Enbac.com**

  Enbac.com là Chuyên trang mua bán tổng hợp hàng trăm nghìn sản phẩm thời trang, điện tử, ô tô, nội thất Toàn quốc. Sàn giao dịch TMĐT chính thức của Vccorp.

- **Muare.vn**

  Trang thông tin cộng đồng rao vặt nhanh và cập nhật 24h, Thời trang, Điện thoại, Máy tính bảng, Thiết bị văn phòng, Điện tử, Kỹ thuật số, Dịch vụ, Điện lạnh, Du lịch, Ô tô, Xe máy, Nhà đất, Nội thất, Rao vặt, Sim số giá rẻ nhất

2.3. Mạng xã hội và phát triển trò chơi

- **Lotus.vn**

  Là mạng xã hội đa nền tảng, Lotus dưới sự điều hành của tập đoàn VCCorp đã trở thành một cộng đồng thông tin thu hút người Việt Nam sử dụng theo đúng với slogan quen thuộc "Người Việt Nam dùng hàng Việt Nam".

- **Sohagame**

  Là một trong ba đơn vị phát hành trò chơi trên thiết bị di động lớn nhất việt nam có trên 50 tựa game từng lọt top trong bảng xếp hạng game Việt, với những tựa game nổi tiếng như : Làng lá phiêu lưu kí, Tân minh chủ, Tam quốc Ca Ca…

CHƯƠNG II

NỘI DUNG CÔNG VIỆC TÌM HIỂU VÀ THỰC HIỆN

I. Nội dung nhiệm vụ chính được giao

- Tìm hiểu về công ty và các quy định .

- Tìm hiểu về code convention, các nguyên tắc thiết kế trong lập trình hướng đối tượng (SOLID).

- Tìm hiểu các kiến thức về lập trình android.

- Học java, các thư viện android liên quan, Google firebase.

- Thực hiện dự án nhỏ liên quan đến kiến thức đã học.

II. Nội dung chi tiết

1. Tìm hiểu về Công ty

- Tìm hiểu về công ty, cách tổ chức của công ty.

- Làm quen với các công cụ làm việc trong công ty.

- Học cách trao đổi, báo cáo, làm việc qua notion, Telegram, Email.

2. Các kiến thức về lập trình Mobile , thiết kế giao diện phần mềm, nền tảng Android, GitHub và Firebase

1.1. SOLID:

1.1.1. SOLID là gì?

SOLID là viết tắt của 5 chữ cái đầu trong 5 nguyên tắc thiết kế hướng đối tượng. Giúp cho lập trình viên viết ra những đoạn code dễ đọc, dễ hiểu, dễ maintain. Nó được đưa ra bởi Robert C. Martin và Michael Feathers. 5 nguyên tắc đó bao gồm:

- Single responsibility principle (SRP)
- Open/Closed principle (OCP)
- Liskov substitution principle (LSP)
- Interface segregation principle (ISP)
- Dependency inversion principle (DIP)

Single responsibility principle

"Mỗi lớp chỉ nên chịu trách nhiệm về một nhiệm vụ cụ thể nào đó mà thôi"

Nguyên lý đầu tiên ứng với chữ S trong SOLID, có ý nghĩa là một class chỉ nên giữ một trách nhiệm duy nhất. Một class có quá nhiều chức năng sẽ trở nên cồng kềnh và trở nên khó đọc, khó maintain. Mà đối với ngành IT việc requirement thay đổi, cần thêm sửa chức năng là rất bình thường, nên việc code trong sáng, dễ đọc dễ hiểu là rất cần thiết.

Open/Closed principle

"Không được sửa đổi một Class có sẵn, nhưng có thể mở rộng bằng kế thừa."

Nguyên lý thứ 2 ứng với chữ O trong SOLID.

Theo nguyên lý này, mỗi khi ta muốn thêm chức năng cho chương trình, chúng ta nên viết class mới mở rộng class cũ (bằng cách kế thừa hoặc sở hữu class cũ) chứ không nên sửa đổi class cũ. Việc này dẫn đến tình trạng phát sinh nhiều class, nhưng chúng ta sẽ không cần phải test lại các class cũ nữa, mà chỉ tập trung vào test các class mới, nơi chứa các chức năng mới.

Thông thường việc mở rộng thêm chức năng thì phải viết thêm code, vậy để thiết kế ra một module có thể dễ dàng mở rộng nhưng lại hạn chế sửa đổi code ta cần làm gì. Cách giải quyết là tách những phần dễ thay đổi ra khỏi phần khó thay đổi mà vẫn đảm bảo không ảnh hưởng đến phần còn lại.

Liskov substitution principle

"Các đối tượng (instance) kiểu class con có thể thay thế các đối tượng kiểu class cha mà không gây ra lỗi."

Interface segregation principle

"Thay vì dùng 1 interface lớn, ta nên tách thành nhiều interface nhỏ, với nhiều mục đích cụ thể."

Nguyên lý này rất dễ hiểu. Hãy tưởng tượng chúng ta có 1 interface lớn, khoảng 100 methods. Việc implements sẽ rất vất vả vì các class implement interface này sẽ bắt buộc phải phải thực thi toàn bộ các method của interface. Ngoài ra còn có thể dư thừa vì 1 class không cần dùng hết 100 method. Khi tách interface ra thành nhiều interface nhỏ, gồm các method liên quan tới nhau, việc implement và quản lý sẽ dễ hơn.

Dependency inversion principle

"Các module cấp cao không nên phụ thuộc vào các modules cấp thấp. Cả 2 nên phụ thuộc vào abstraction.

Interface (abstraction) không nên phụ thuộc vào chi tiết, mà ngược lại (Các class giao tiếp với nhau thông qua interface (abstraction), không phải thông qua implementation.)"

Những thành phần trong 1 chương trình chỉ nên phụ thuộc vào những cái trừu tượng (abstraction). Những thành phần trừu tượng không nên phụ thuộc vào các thành phần mang tính cụ thể mà nên ngược lại.

Những cái trừu tượng (abstraction) là những cái ít thay đổi và biến động, nó tập hợp những đặc tính chung nhất của những cái cụ thể. Những cái cụ thể dù khác nhau thế nào đi nữa đều tuân theo các quy tắc chung mà cái trừu tượng đã định ra. Việc phụ thuộc vào cái trừu tượng sẽ giúp chương trình linh động và thích ứng tốt với các sự thay đổi diễn ra liên tục.

1.2. Tổng quan về giao diện người dùng:

1.2.1. Giao diện người dùng là gì?

Giao diện người sử dụng –**User Interface** (UI) là một khái niệm không mấy xa lạ với những nhà thiết kế chuyên nghiệp. UI là khái niệm xuất hiện với sự ra đời của máy tính, vì thế nói tới UI có thể hiểu là một giao diện để người dùng và máy móc làm việc với nhau.

User Interface là một khái niệm để nói tới nơi mà con người và máy móc cùng làm việc với nhau. Với sự ra đời của máy tính, UI có thể coi là những gì chúng ta nhìn thấy trên màn hình và tương tác với máy tính thông qua những câu lệnh được mã hóa.

Các thành phần của UI Design :

• **Bố cục:** Bố cục quy định cụ thể thành phần nào bạn sẽ có trên trang, chúng sẽ được đặt ở vị trí nào và như thế nào. Đây là yếu tố quyết định. Bố cục nên đơn giản, dễ dàng cho người dùng tìm được cái họ muốn tìm, và quan trọng hơn cả là thu hút họ làm thứ bạn muốn.

• **Màu sắc:** Những màu sắc bạn sử dụng có ảnh hưởng trực tiếp đến thiết kế giao diện người dùng. Chúng phải thể hiện được thương hiệu của công ty, và cũng gây được tiếng vang với người dùng. Sử dụng màu xám đục cho một trang web của phòng tập thể hình sẽ không thúc đẩy và khuyến khích người dùng tham gia.

• **Kiểu chữ:** Nghệ thuật sắp chữ có thể khiến bạn thành công hoặc thất bại trong việc thiết kế giao diện người dùng. Dù nó có vẻ như là một dòng chữ, nhưng nó cần phải hấp dẫn thị giác. Ví dụ, website của tạp trí Fortune 500 sẽ trông tệ thế nào khi toàn bộ font chữ đều là Comic Sans?

Các nguyên tắc thiết kế giao diện

Sự quen thuộc của người sử dụng: giao diện phải được xây dựng dựa trên các thuật ngữ và các khái niệm mà người sử dụng có thể hiểu được hơn là những khái niệm liên quan đến máy tính. Ví dụ: hệ thống văn phòng nên sử dụng các khái niệm như thư, tài liệu, cặp giấy… mà không nên sử dụng những khái niệm như thư mục, danh mục…

Thống nhất: hệ thống nên hiển thị ở mức thống nhất thích hợp. Ví dụ: các câu lệnh và menu nên có cùng định dạng …

Tối thiểu hoá sự bất ngờ: nếu một yêu cầu được xử lý theo cách đã biết trước thì người sử dụng có thể dự đoán các thao tác của những yêu cầu tương tư.

Khả năng phục hồi: hệ thống nên cung cấp một số khả năng phục hồi từ lỗi của người sử dụng và cho phép người sử dụng khôi phục lại từ chỗ bị lỗi. Khả năng này bao gồm cho phép làm lại, hỏi lại những hành động như xoá, huỷ …

Hướng dẫn người sử dụng: như hệ thống trợ giúp, hướng dẫn trực tuyến …

Tính đa dạng: hỗ trợ nhiều loại tương tác cho nhiều loại người sử dung khác nhau.

1.2.2.Tầm quan trọng của thiết kế giao diện người dùng

Toàn bộ quá trình thu thập yêu cầu người dùng, đặt những yếu tố khác nhau của phần mềm và tạo ra một giao diện người dùng hiệu quả được gọi là thiết kế giao diện người dùng (UI design).

Tầm quan trọng của thiết kế giao diện người dùng :

Tạo ra giao diện dễ nhìn , dễ sử dụng, phù hợp với người dùng

Đầy đủ các chức năng theo yêu cầu của người dùng

Khả năng tương tác nhanh

1.3. Tổng quan về android:

1.3.1. Hệ điều hành android

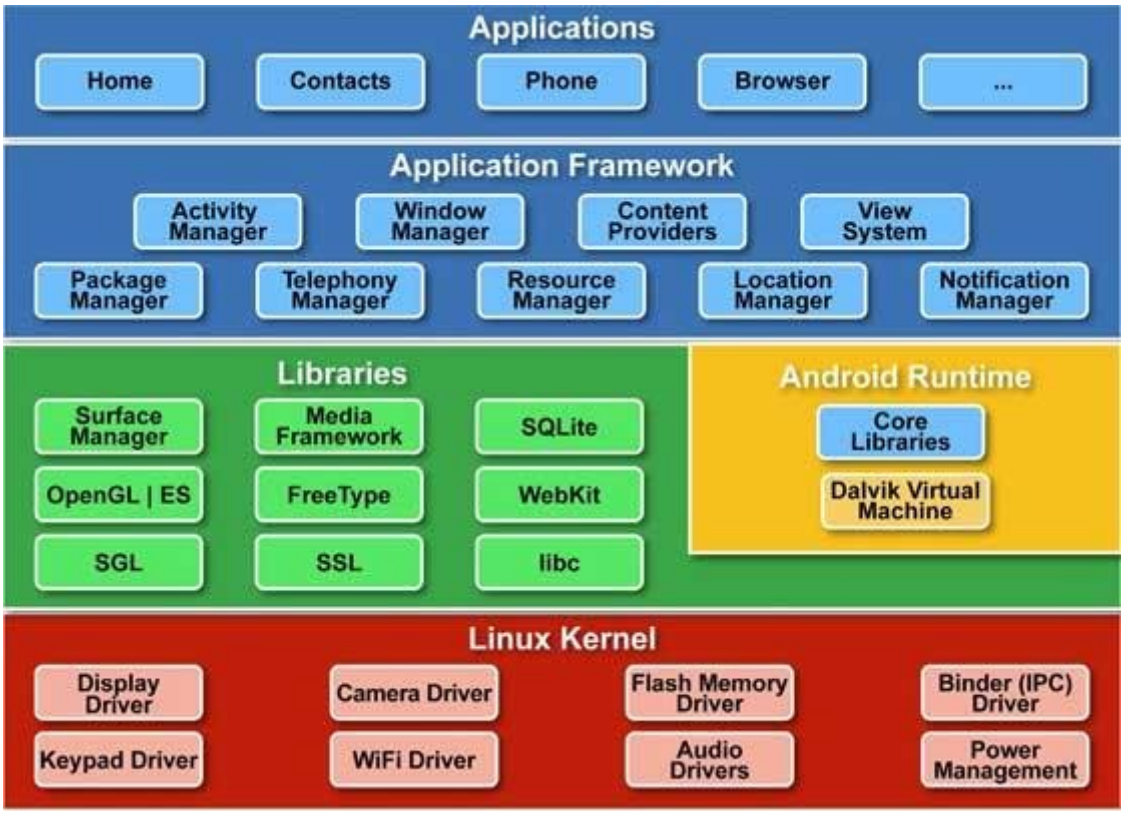
Android là một hệ điều hành có dạng mã nguồn mở, nó hoạt động dựa trên nền tảng Linux và được thiết kế dành riêng cho những thiết bị di động cảm ứng hoặc máy tính bảng. Trước đây, hệ điều hành này được phát triển bởi tổng công ty Android và được tài trợ bởi Google. Cho đến năm 2005 thì Google đã mua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

1.3.2. Tìm hiểu về kiến trúc của 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.

1.3.3. Các phiên bản của 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.

1.3.4. Một số ngôn ngữ 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.

1.3.5. 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…

2.1. Tổng quan về Java

2.1.1. Giống như C++, nhưng 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.

2.1.2. 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.

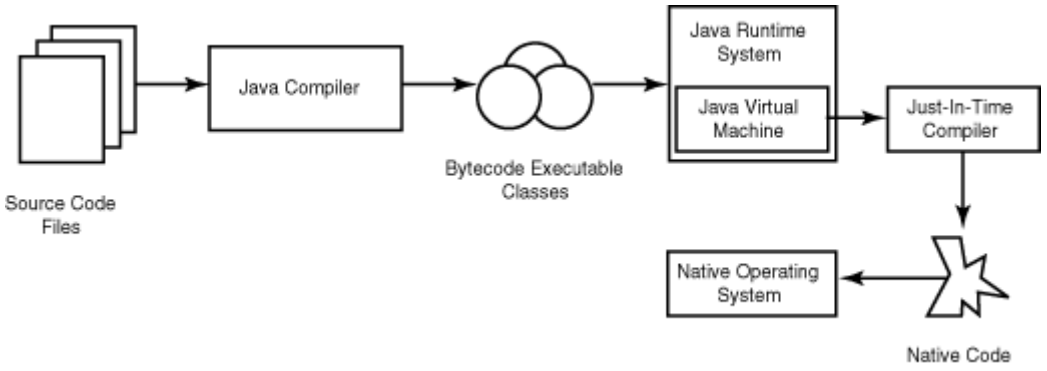
- Ở cấp độ mã nguồn: kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng 1 bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.

- 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.

2.1.3. 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.



.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).

2.1.4. Cơ thế thu gom rác tự động (GC)

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.

3.1.5. Đa luồng

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

3.1.6. 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 bytecode 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.

3.1.6. JVM

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.

2.3. Github

2.3.1. Khái Niệm cơ bản

GitHub là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Nhưng cách sử dụng GitHub như thế nào? Nó sử dụng để hợp tác nhiều người lại với nhau, từ mọi nơi trên thế giới, lên kế hoạch, theo dõi và làm chung một dự án.

GitHub cũng là một nền tảng lưu trữ online lớn nhất trên thế giới về các dự án nhiều người làm.

Git là gì?

Trước tiên, chúng ta cần phải biết rõ Git là gì trước, vì nó là trái tim của GitHub. Git là một hệ quản trị phiên bản được phát triển bởi Linus Torvalds (tên rất quen phải không, người tạo ra Linux đó).

Vậy, hệ quản trị phiên bản – version control system là gì?

Khi lập trình viên tạo một dự án mới, họ sẽ cần liên tục cập nhật mã nguồn. Kể cả khi dự án đã được xuất bản, họ vẫn cần phải cập nhật các phiên bản mới cho nó, sửa lỗi, thêm tính năng, vâng vâng.

Hệ quản trị phiên bản sẽ giúp giám sát những thay đổi của code. Hơn thế nữa, nó còn lưu lại thông tin ai thay đổi gì để có thể khôi phục code cũ bị xóa hoặc code đã từng được sửa.

Codes không được ghi đè lên nhau vì Git lưu nhiều phiên bản copies trong repository (thư viện) của nó

Hub là gì?

Nếu Git là trái tim của GitHub thì Hub lại là phần hồn của nó. Hub trong GitHub là nơi biến những dòng lệnh, Git, thành một mạng xã hội khổng lồ cho lập trình viên.

Bên cạnh đóng góp vào những dự án chính, GitHub còn cho phép người dùng tương tác theo kiểu mạng xã hội. Bạn có thể theo dõi, và xem những người bạn thích làm gì, họ đang kết nối với ai, vâng vâng.

Repository

Repository hay repo là một thư viện nơi chứa các files của dự án. Nó có thể đặt trong bộ lưu trữ của GitHub hoặc trong repository của máy tính local. Bạn có thể chứa files code, hình ảnh, âm thanh hoặc mọi thứ liên quan đến dự án trong một repository.

Branch

Branch là một bản sao của repository. Bạn có thể sử dụng Branch để triển khai dự án theo hướng cô lập không ảnh hưởng đến dự án chính.

Làm việc với branch vì vậy sẽ không ảnh hưởng tới repository chính hoặc những branches khác. Nếu bạn hoàn tất công việc, bạn có thể "Merge" (nhập) branch vào những branch khác khoặc repository chính bằng cách dùng lệnh Pull Request

Pull Request

Pull request có nghĩa là bạn thông báo với những người khác rằng bạn đã đẩy những thay đổi của Branch lên Repository tổng (master repository). Các cộng tác viên của repository này sẽ có chấp nhận hoặc từ chối pull request này. Khi nó được mở ra, bạn có thể thảo luận và xem lại công việc với những người cùng làm khác.

Các bước để tạo một pull request trong GitHub là:

1. Chuyển tới repository và tìm menu branch
2. Trong branch menu, chọn branch chứa thay đổi của bạn
3. Nhấn vào nút **New pull request** bên cạnh menu branch
4. Thêm tiêu đề và mô tả vào pull request của bạn
5. Nhấn nút **Create pull request**

Fork một Repository

Fork một repository (forking a repository) có nghĩa là bạn tạo một dự án mới dựa trên dự án cũ. Tức là, sao chép hoàn toàn một repository đã tồn tại, tạo ra các thay

đổi cần thiết, và lưu phiên bản mới này dưới dạng một repository độc lập hoàn toàn mới và gọi nó là dự án của riêng bạn.

Tính năng này vô cùng tiện lợi để đẩy nhanh tiến độ dự án. Vì là một dứ án hoàn toàn mới, repository chính sẽ không bị ảnh hưởng. Nếu repostiory master được cập nhật, bạn cũng có thể áp dụng các cập nhật đó lên bản fork của bạn.

Các bước để fork một repository trong GitHub là:

1. Tìm repository bạn muốn fork
2. Nhấn vào nút **Fork**

2.3.2. Sử dụng git

2.3.2.1. Cơ bản về Git

Git là tên gọi của một Hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là Working Tree.

2.3.2.2. Flow cơ bản khi sử dụng Git

Sau đây là flow cơ bản khi bạn sử dụng Git:

- Clone project từ server về Local Repository
- Check-out 1 nhánh từ Local Repository về Working Space
- Bạn sẽ làm việc (thêm, sửa, xoá tại Working Space)
- Add : xác nhận sự thay đổi của các files (đưa đến vùng Staging Area)
- Commit: cập nhật sự thay đổi lên Local Repository

Về cơ bản đến đây là bạn đã hoàn thành 1 chu trình sử dụng Git. Lúc này, nếu như bạn muốn cập nhật sự thay đổi này lên server thì bạn sẽ dùng lệnh push để đẩy chúng lên server.

2.3.2.3. Những câu lệnh cơ bản thường sử dụng

Thiết lập chứng thực cá nhân

$ git config --global user.name "User Name"

$ git config --global user.email "username@gmail.com"

Lưu ý: --global được sử dụng để áp dụng cho tất cả các projects. Nếu bạn ko sử dụng --global thì settings sẽ chỉ dùng cho riêng project đó.

Tạo một kho chứa Git

$ git init

Nếu như bạn muốn theo dõi một dự án cũ trong Git, bạn cần ở trong thư mục của dự án đó. Lệnh này sẽ tạo một thư mục mới có tên .git, thư mục này chứa tất cả các tập tin cần thiết cho kho chứa.

Sao chép một kho chứa đã tồn tại

$ git clone https://github.com/user/repository.git

Câu lệnh trên sẽ tạo một thư mục mới có tên giống trên của repo.

Nhánh trong git

Khi sử dụng Git, bạn có thể tạo ra nhiều nhánh (branch) khác nhau. Câu lệnh Git này dùng để kiểm tra branch hiện tại:

$ git branch

Để tạo mới một branch:

$ git branch <name_branch>

Để chuyển và tạo mới:

$ git branch -b <name_branch>

Chuyển nhánh

Trước khi muốn thay đổi source code, điều đầu tiên mà bạn cần phải làm là checkout một nhánh. Để checkout một nhánh, bạn dùng câu lệnh Git sau:

$ git checkout <name_branch>

Cập nhật thay đổi

Sau khi bạn thay đổi source code: thêm mới, sửa, xoá files,… Bạn cần phải cập nhật lên Staging Area. Để cập nhật hết các files:

$ git add .

Sau lệnh add, bạn cần sử dụng câu lệnh Commit để đây thông tin thay đổi lên Local Repository:

$ git commit -m "Message"

Cập nhật lên server

Sau câu lệnh Commit, thông tin mới chỉ được cập nhật lên Local Repository. Nếu muốn cập nhật lên server thì bạn phải sử dụng câu lệnh push:

$ git push origin <name_branch>

Ngoài ra, nếu chưa tồn tại remote trên server thì bạn cần phải add mới một remote trước rồi mới push:

$ git remote add origin <remote_url>

$ git push origin <name_branch>

Gộp nhánh

Sau một thời gian cập nhật các file và push lên git trên branch mới, bây giờ mình cần ghép (merge) code lại vào nhánh gốc (master). Trước tiên, cần phải checkout ra khỏi branch hiện tại cần gộp để vào branch master, sau đó thì dùng lệnh merge để ghép branch mới vào master:

$ git checkout master

$ git merge <new_branch>

Xem lại lịch sử commit

$ git log

Lệnh git log sẽ cho bạn biết về người commit, ngày giờ, message của những lần commit đó.

Xem thay đổi trước khi push

$ git diff

Lệnh này giúp bạn biết những gì đã được thay đổi giữa nhánh hiện tại và nhánh trước nó.

Gộp commit

$ git rebase -i HEAD~

Sau dấu ~ là số commit bạn muốn gộp. Sau khi gõ lệnh này một cửa sổ trình soạn thảo hiện ra. Thay đổi ký tự pick của dòng các dòng sau dòng đầu thành s rồi lưu lại/kết thúc. Khi đó, trình soạn thảo để chỉnh sửa giải thích commit thiết lập cho commit sau khi đã tổng hợp sẽ được hiển thị, nên hãy chỉnh sửa lưu lại/kết thúc.

Pull từ remote repository

$ git pull origin master

Lệnh trên sẽ gộp những thay đổi mới kéo về từ máy chủ từ xa với nhánh hiện tại trên máy local.

2.4. Công cụ 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:

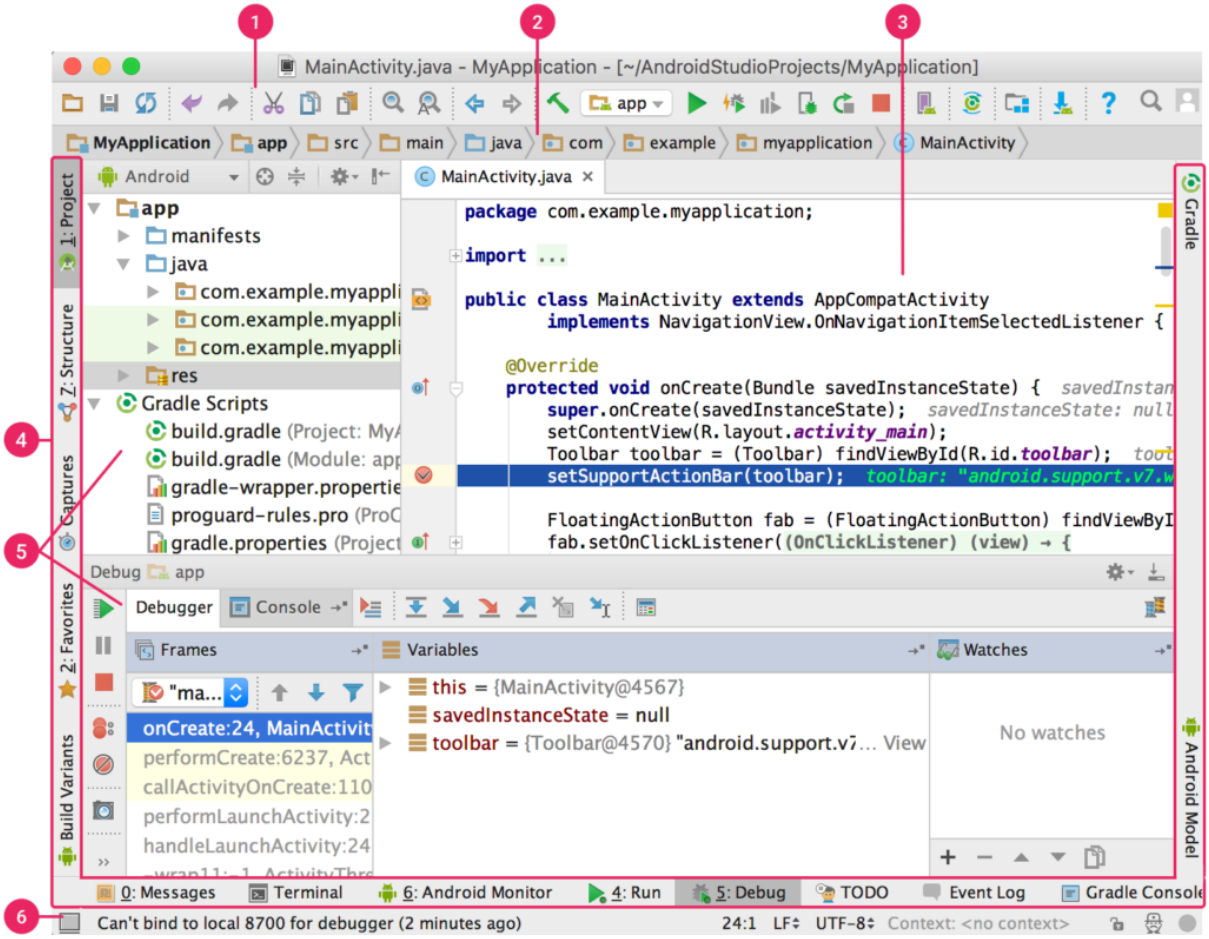
Android Studio là Môi trường phát triển tích hợp chính thức (IDE) dành cho phát triển ứng dụng Android, dựa trên IntelliJ IDEA . Trên trình soạn thảo mã và công cụ phát triển mạnh mẽ của IntelliJ, Android Studio cung cấp nhiều tính năng nâng cao hiệu suất của bạn khi xây dựng ứng dụng Android, chẳng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



(Giao diện người dùng 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.

3. Ứng dụng dự án thực tế

3.1. Mô tả bài toán thực tế

Trong quá trình tự học tập tiếng Trung, gặp phải không ít khó khăn, như không hiểu ý nghĩa các từ, không biết dùng từ như nào là hợp lý, học kiểu truyền thống còn khá khô khan, Và dựa vào các kiến thức đã được tìm hiểu, vậy nên em đã quyết định xây dựng ứng dụng từ điển học tiếng Trung.

3.2. Ứng dụng Mandarin Learning

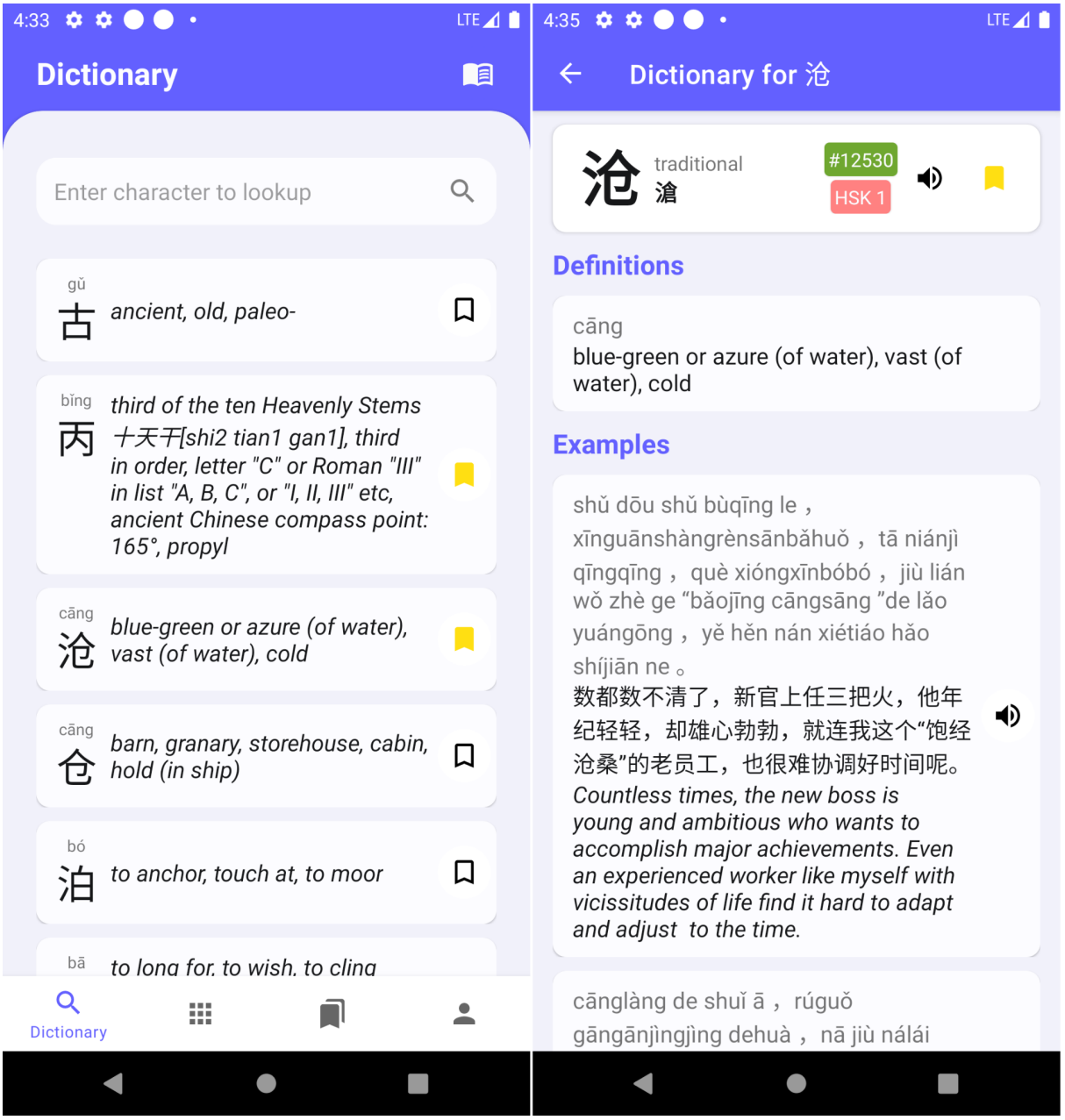
Ứng dụng này được xây dựng theo mô hình MVP, một trong ba architecture phổ biến của ứng dụng Android

MVP là một User Interface Architectural Pattern (kiến trúc giao diện người dùng) được thiết kế để tạo điều kiện cho Automated Unit Testing và cải tiến **Separation of Concerns** trong việc trình bày logic (presentation logic).

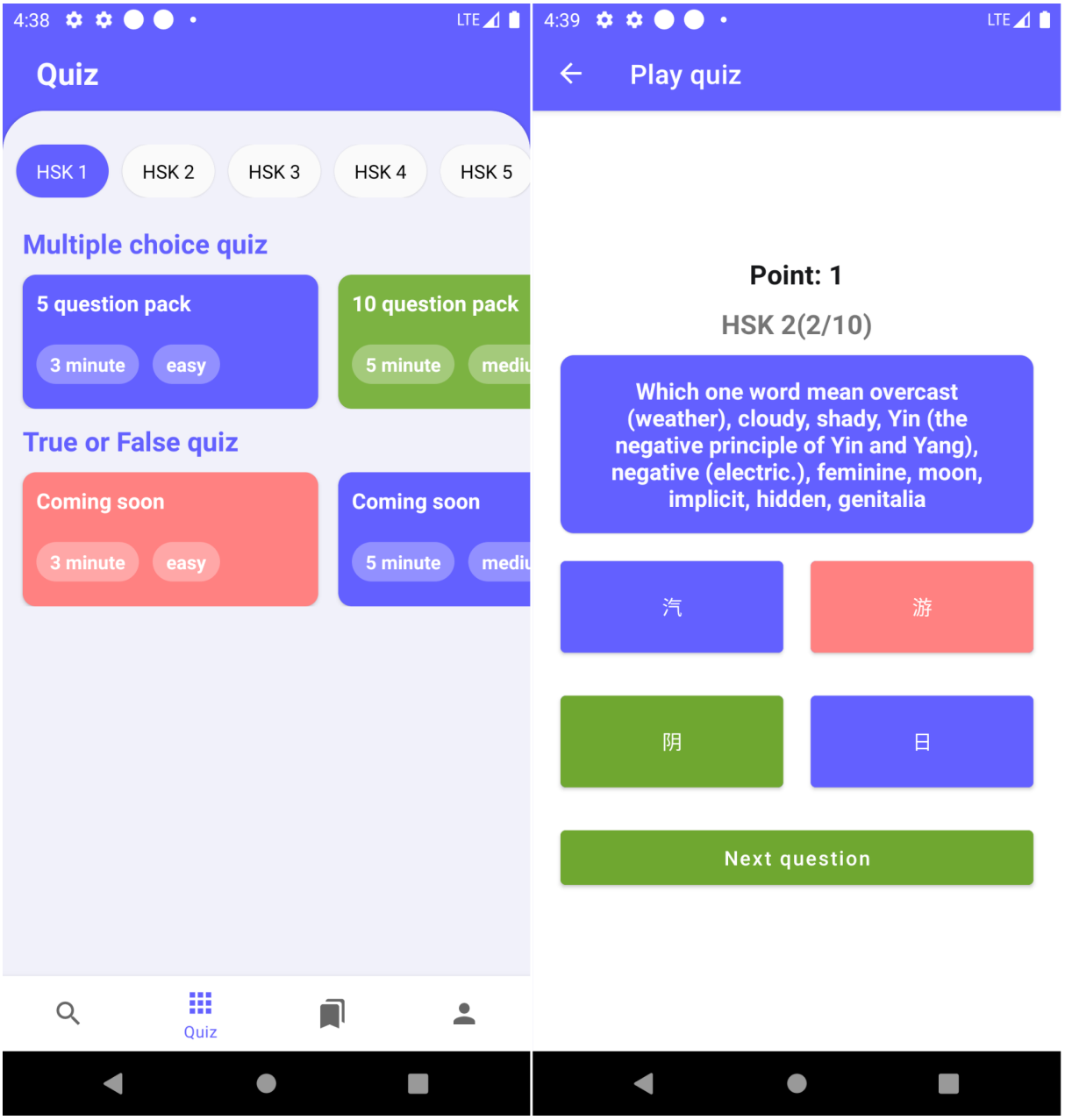
Separation of Concerns có thể hiểu là nguyên tắc để tách một chương trình phần mềm thành các thành phần riêng biệt, phần này thay đổi thì không phục thuộc vào phần kia và ngược lại.

Ứng dụng được xây dựng với 7 màn hình chính:

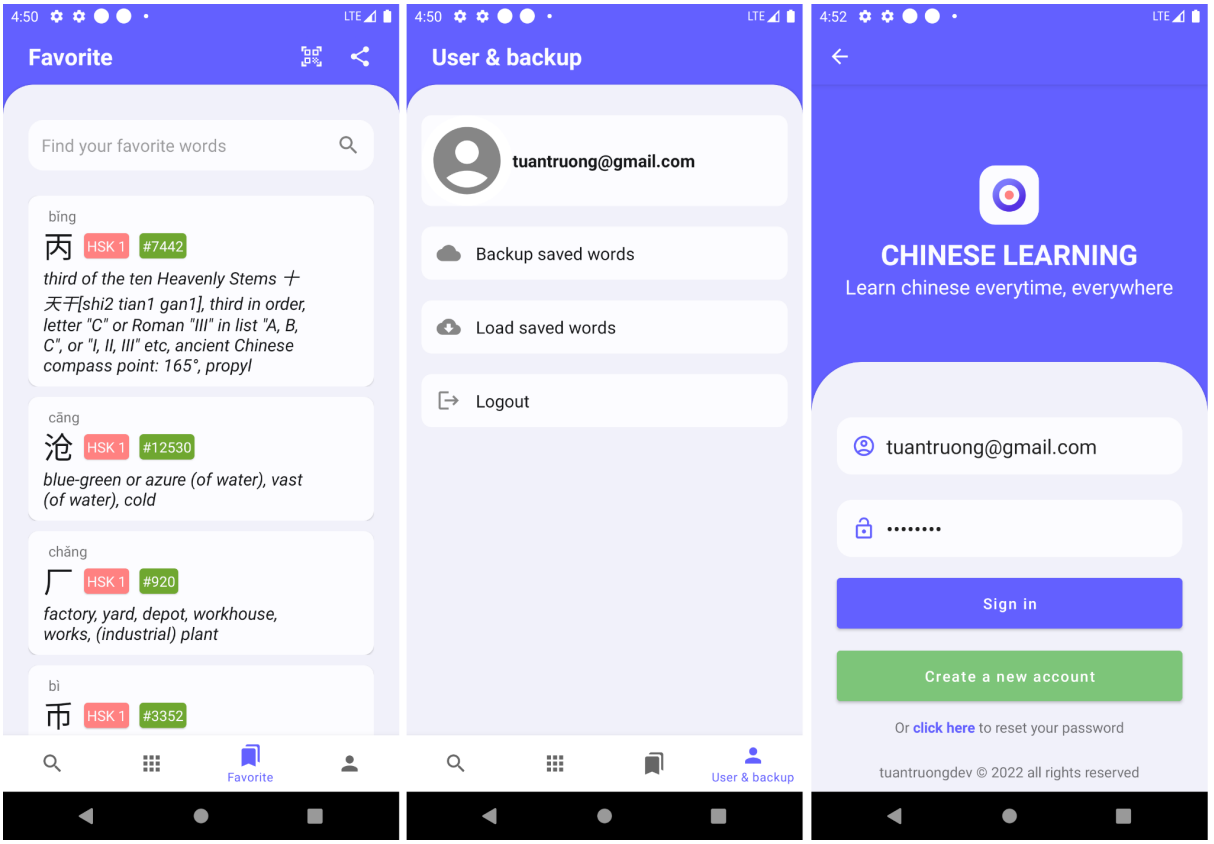
- Màn hình từ điển (Màn hình chính)



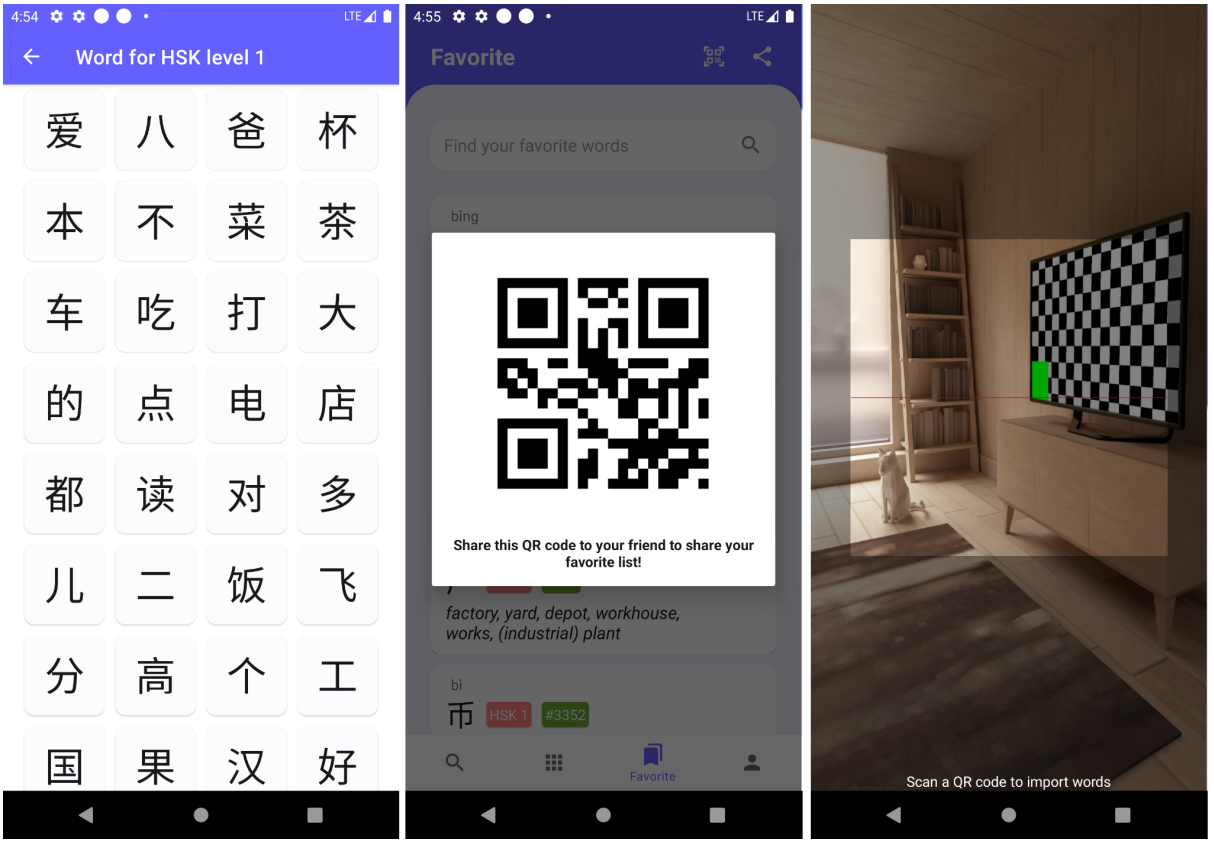
- Màn hình Quiz



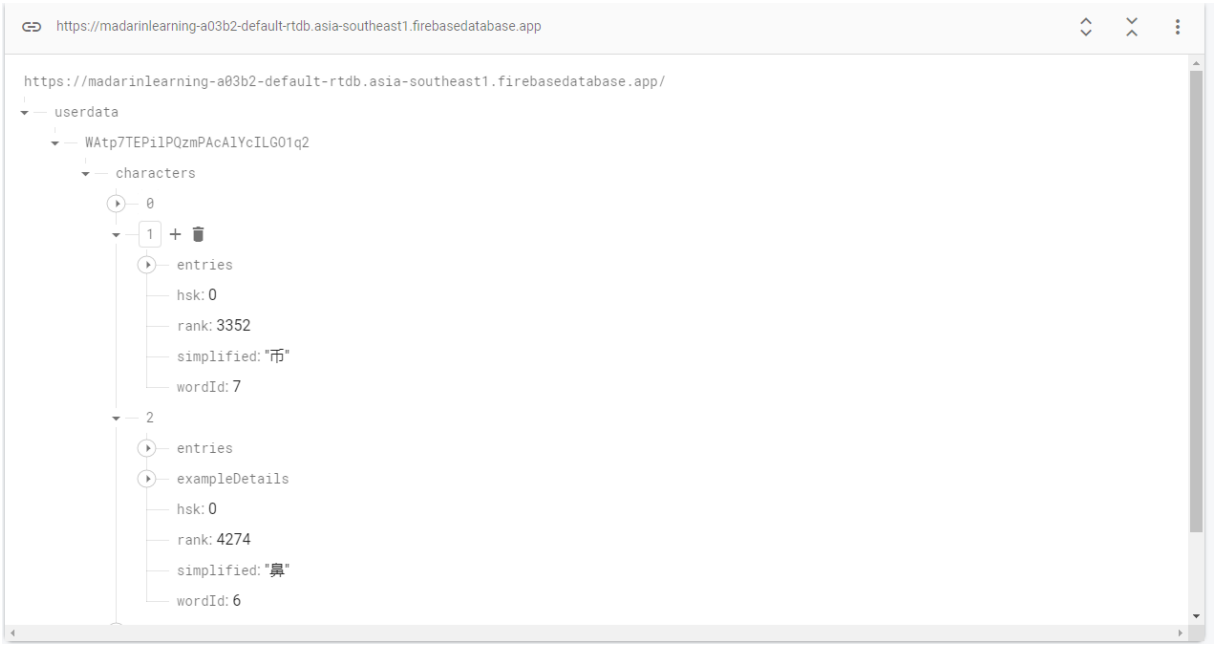
- Màn hình favorite, user, và login



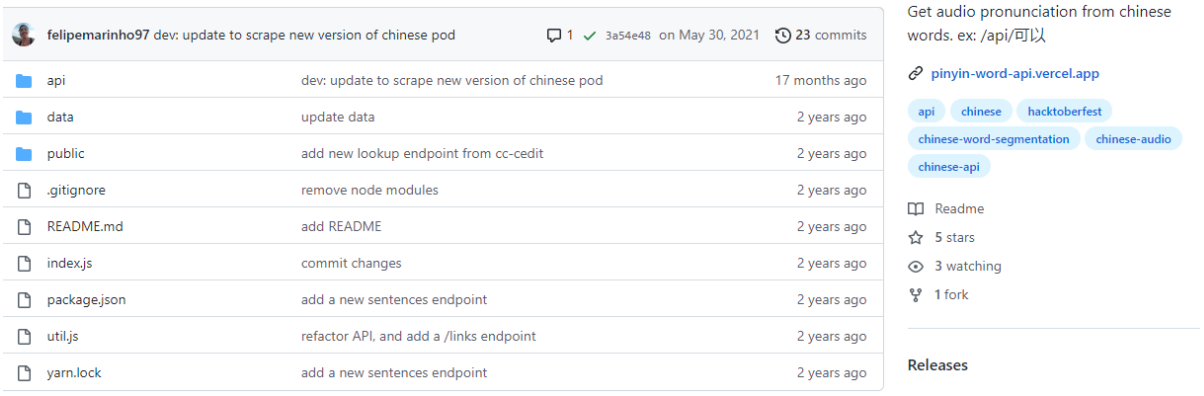
- Một số màn hình phụ (Danh sách từ, share, quét từ được share)



- Ứng dụng sử dụng REST API open source và lưu trữ dữ liệu trên Google Firebase



- REST API open source trên github



(https://github.com/felipemarinho97/pinyin-word-api)

CHƯƠNG III
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

I. Kết luận

1. Kỹ năng mềm

Sau 2 tháng thực tập tại Công ty VCCORP giúp em học hỏi được rất nhiều điều mà tự học và môi trường trước đó không có được

- Kỷ luật và chấp hành đúng giờ làm việc

Trong quá trình thực tập ở công ty, kỷ luật là một điều rất quan trọng. Nếu bản thân không thực hiện tốt được những kỷ luật đó, công việc dù giỏi đến đâu cũng không thể hoàn thành và đánh giá cao.

Ở VCCORP, mỗi ngày làm việc 8 tiếng, bắt đầu từ 9h -17h30 các ngày từ thứ 2 đến thứ 7. Thời gian chính là yếu tố cần thiết nhất cho công việc.

- Kỹ năng giao tiếp, tác phong ứng xử chuyên nghiệp hơn

Kỹ năng giao tiếp luôn được xem là yếu tố then chốt đối với sự phát triển toàn diện của một người. Nhất là làm việc trong môi trường công nghệ thông tin, cần phải biết trình bày và giải thích rõ ràng, cùng những người khác tìm ra và thực hiện giải pháp, giao nhiệm vụ cho cả nhóm một cách hiệu quả.

VCCORP cũng giúp thay đổi những tác phong ứng xử trong môi trường công sở:

- Biết cách sử dụng Email, Telegram để liên lạc các bộ phận trong công ty.

- Tạo được mối quan hệ thân thiết với những anh chị đồng nghiệp, được anh chị nhiệt tình chỉ bảo.

- Hòa nhã với các nhân viên nơi thực tập, tham gia sôi nổi các hoạt động ngoài giờ làm việc cùng mọi người trong dự án.

- Chủ động tiếp cận công việc và sẵn sàng hỗ trợ đồng nghiệp để có thể hoàn thành công việc chung, tự khẳng định năng lực bản thân.

- Phong cách, trang phục luôn chỉnh tề, phù hợp, lịch sự

- Khả năng làm việc nhóm

Làm việc theo nhóm giúp cho bản thân em rất nhiều. Giúp em có thể lắng nghe ý kiến từ nhiều phía khác nhau, trợ giúp và tôn trọng ý kiến của mọi người. Bên cạnh

đó, làm việc nhóm cũng giúp em rèn luyện khả năng Thuyết phục- thuyết phục các thành viên phải trao đổi, suy xét những ý tưởng đã đưa ra. Đồng thời em cũng nâng cao khả năng tự bảo vệ và thuyết phục người khác đồng tình với ý kiến của mình.

- Lên kế hoạch công việc và quản lý thời gian

Lên kế hoạch cho công việc mình làm là một trong những kỹ năng em học được ở công ty. Để làm tốt công việc được giao, bản thân em đã biết lập kế hoạch cho các vấn đề, những phần quan trọng cần giải quyết được sắp xếp lên đầu cứ như thế theo thứ tự giảm dần

Việc này không chỉ giúp bản thân em điều chỉnh thời gian cho phù hợp mà còn giúp tiến độ công việc được nâng cao.

2. Áp dụng thực tiễn

Trong quá trình thực tập, Em đã được giao làm 4 dự án trong vòng 2 tháng, dự án mà em báo cáo ở trên là dự án cuối cùng, cũng như dự án được em đầu tư thời gian và công sức nhất.

Áp dụng những kiến thức cơ bản đã được học ở trường và các kiến thức nâng cao được học ở đợt thực tập này để xây dựng được ứng dụng học tiếng Trung, dự án này có những kiến thức căn bản về lập trình, từ CTDL & GT đến những công nghệ nâng cao đang được dùng phổ biến như : Firebase,Rest API, Git

3. Hạn chế

Do thời gian thực tập còn hạn chế, và kinh nghiệm thực tế chưa có nhiều nên không thể tránh khỏi những thiếu sót về kiến thức, kỹ năng cũng như kết quả đạt được. Em sẽ cố gắng khắc phục những hạn chế đó để có thể hoàn thành tốt bài báo cáo trong đợt Bảo vệ đồ án sắp tới.

II. Hướng phát triển

- Về ứng dụng học tiếng Trung, thay vì sử dụng backend là dự án open source và google firebase, Em sẽ viết lại backend bằng ngôn ngữ golang, nắm chắc 100% code dự án cả 2 phía backend và frontend
- Về bản thân, tiếp tục công việc tại công ty VCCORP để học hỏi, trau dồi kinh nghiệm ngành nghề, liên tục học hỏi các ngôn ngữ, công nghệ để nâng cao trình độ bản thân.

TÀI LIỆU THAM KHẢO

[1] Giáo trình môn Thiết kế giao diện phần mềm (ĐH Công nghệ thông tin và truyền thông Thái Nguyên)

[2] Ian Darwin (2011) Android Cookbook

[3] David Griffiths (2016) Head First Android Development

[4] https://firebase.google.com/docs

[5] https://developer.android.com

[6] https://www.figma.com/file/m8Yj9KdvSWRs5rG6lmkKVD/Online-Learning-UI-Kit-Free---SINAU-(Community)?node-id=5%3A0

[7] https://www.behance.net/gallery/96147213/Dictionary-app-design-concept

[8] https://viblo.asia/p/nhung-lenh-git-co-ban-can-nho-V3m5W1OyZO7