**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**TP.Hồ Chí Minh - Năm 2021**

**GVHD: NGUYỄN TRỌNG TIẾN**

**SVTH: TRƯƠNG ĐĂNG QUANG**  **17069131**

**ĐỖ THÙY LINH 17028 671**

ỨNG DỤNG ÔN THI TIẾNG ANH VÀO LỚP 10 TRÊN DI ĐỘNG

**KHÓA LUẬN TỐT NGHIỆP**

**INDUSTRIAL UNIVERSITY OF HO CHI MINH CITY**

****FACULTY OF INFORMATION TECHNOLOGY**

**TRUONG DANG QUANG**

**DO THUY LINH**

**BUILDING A MOBILE APP (ANDROID) TO EXAM PREPARATION FOR 10TH GRADE ENGLISH**

Major: Software Engineering

**Supervisor: Dr. Nguyen Trong Tien**

HO CHI MINH CITY, 2020

# BUILDING A MOBILE APP (ANDROID) TO EXAM PREPARATION FOR 10TH GRADE ENGLISH

**Abstract:** Build a mobile application that can study for the 10th grade English exam anywhere and then send the results to the database to store the learning results and track the progress of each student.

**Reason for writing:** With the development of information technology today, learning and the needs of society are increasingly enhanced, making the best use of time for self-study is the top concern and necessity for every student. The e-exam application is a device that has been mentioned recently and it supports very well in the self-review of each individual. This is an interesting topic.

**Problem:** Build a mobile app to exam preparation for 10th grade english.

**Methods:** Mobile app uses React Native language with Android Studio IDE to make topics. Also use more external libraries to make the interface more user-friendly.

**Results:** Successfully building the mobile app theme, the functions are almost complete perfection.

**Conclusion:** Successfully building the mobile app theme and website, the functions are almost complete perfection. However, there are still some performance limitations and code optimization.

# LỜI CẢM ƠN

Lời đầu tiên nhóm chúng em xin gửi lời cảm ơn chân thành nhất đến Quý Thầy Cô Trường Đại học Công nghiệp Thành phố Hồ Chí Minh và đặc biệt là Quý Thầy Cô Khoa Công nghệ thông tin trường Đại học Công nghiệp Thành phố Hồ Chí Minh đã trang bị cho chúng em đầy đủ các kiến thức tổng quan và chuyên ngành để có thể vận dụng vào đề tài khóa luận này.

Nhằm củng cố và vận dụng những kiến thức đã tích lũy được trong 4 năm học, nhóm chúng em quyết định chọn đề tài “Xây dựng app ôn thi Tiếng Anh vào lớp 10” để làm đề tài khóa luận tốt nghiệp. Bên cạnh đó, chúng em cũng xin gửi lời cảm ơn sâu sắc nhất đến TS. Nguyễn Trọng Tiến, Thầy đã dành thời gian, tâm huyết và tận tâm hướng dẫn để bài khóa luận được hoàn thiện và đạt kết quả tốt nhất.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, bài báo cáo này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý Thầy Cô để có điều kiện bổ sung, nâng cao kiến thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Cuối cùng, chúng em xin kính chúc quý Thầy Cô dồi dào sức khỏe và thành công trong sự nghiệp trồng người cao quý.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn !

Thành phố Hồ Chí Minh, ngày 21 tháng 05 năm 2021

**Sinh viên thực hiện**

**Trương Đăng Quang, Đỗ Thùy Linh**

# NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... .......................................................................................................................................

Tp. HCM, ngày tháng năm 2021

**Giảng viên hướng dẫn**

**TS. Nguyễn Trọng Tiến**

# NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN

....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... ....................................................................................................................................... .......................................................................................................................................

Tp. HCM, ngày tháng năm 2021

**Giảng viên phản biện**

# MỤC LỤC

[**BUILDING A MOBILE APP (ANDROID) TO EXAM PREPARATION FOR 10TH GRADE ENGLISH 3**](#_Toc73405165)

[**LỜI CẢM ƠN 4**](#_Toc73405166)

[**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN 5**](#_Toc73405167)

[**NHẬN XÉT VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN 6**](#_Toc73405168)

[**MỤC LỤC 7**](#_Toc73405169)

[**DANH MỤC CÁC HÌNH VẼ 10**](#_Toc73405170)

[**DANH MỤC CÁC BẢNG BIỂU 10**](#_Toc73405171)

[**1.1 Tổng quan 11**](#_Toc73405172)

[**1.2 Mục tiêu đề tài 12**](#_Toc73405173)

[**1.3 Phạm vi đề tài 12**](#_Toc73405174)

[**1.4 Mô tả yêu cầu chức năng 12**](#_Toc73405175)

[**CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 13**](#_Toc73405176)

[**2.1 Android application [1] 13**](#_Toc73405177)

[**2.1.1 Giới thiệu về hệ điều hành android 13**](#_Toc73405178)

[**2.1.2 Kiến trúc hệ điều hành android 14**](#_Toc73405179)

[**2.2 React Native 17**](#_Toc73405180)

[**2.2.1 Tổng quan 17**](#_Toc73405181)

[**2.2.2 Ưu điểm 18**](#_Toc73405182)

[**2.2.3 Nhược điểm 18**](#_Toc73405183)

[**2.3 ReactJs 19**](#_Toc73405184)

[**2.3.1 Tổng quan 19**](#_Toc73405185)

[**2.3.2 Ưu điểm 19**](#_Toc73405186)

[**2.4 Redux 19**](#_Toc73405187)

[**2.4.1 Tổng quan về Redux 19**](#_Toc73405188)

[**2.4.2 Nguyên lý của Redux 20**](#_Toc73405189)

[**2.4.3 Thành phần của Redux 21**](#_Toc73405190)

[**2.4.4 Ưu điểm của Redux 22**](#_Toc73405191)

[**2.5 Axios 22**](#_Toc73405192)

[**2.5.1 Tổng quan về Axios 22**](#_Toc73405193)

[**2.5.2 Đặc điểm của Axios 23**](#_Toc73405194)

[**2.6 Ngôn ngữ lập trình C# 24**](#_Toc73405195)

[**2.6.1 Nền tảng .NET 24**](#_Toc73405196)

[**2.6.2 Tổng quan về ngôn ngữ C# 26**](#_Toc73405197)

[**2.6.3 Đặc trưng của ngôn ngữ lập trình C# 27**](#_Toc73405198)

[**2.7 Cơ sở dữ liệu MS SQL Server 28**](#_Toc73405199)

[**2.7.1 Tổng quan về SQL Server 28**](#_Toc73405200)

[**2.7.2 Các thành phần cơ bản của SQL Server 30**](#_Toc73405201)

[**CHƯƠNG 3 : PHÂN TÍCH VÀ ĐẶC TẢ CÁC YÊU CẦU CHỨ NĂNG 32**](#_Toc73405202)

[**3.1 Use case tổng quát 32**](#_Toc73405203)

[**3.2 Danh sách tác nhân và mô tả 32**](#_Toc73405204)

[**3.3 Danh sách các tình huống hoạt động (Use case) 32**](#_Toc73405205)

[**3.4 Danh sách Mock up 32**](#_Toc73405206)

[**3.4.1 Mock up đăng nhập 32**](#_Toc73405207)

[**3.4.2 Mock up làm đề thi 33**](#_Toc73405208)

[**3.4.3 Mock up cài đặt 33**](#_Toc73405209)

[**3.5 Tình huống hoạt động 34**](#_Toc73405210)

[**3.5.1 Đặc tả UC01\_ 34**](#_Toc73405211)

[**3.5.2 Đặc tả UC02\_ 34**](#_Toc73405212)

[**CHƯƠNG 4 : THIẾT KẾ VÀ HIỆN THỰC 35**](#_Toc73405213)

[**4.1 Class diagram 35**](#_Toc73405214)

[**4.2 Sitemap 35**](#_Toc73405215)

[**4.3 Database diagram 35**](#_Toc73405216)

[**4.4 Hiện thực 35**](#_Toc73405217)

[**CHƯƠNG 5 : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 36**](#_Toc73405218)

[**5.1 Kết quả đạt được 36**](#_Toc73405219)

[**5.2 Hạn chế của đồ án 36**](#_Toc73405220)

[**5.3 Hướng phát triển 36**](#_Toc73405221)

[**TÀI LIỆU THAM KHẢO 37**](#_Toc73405222)

[**PHỤ LỤC: TEST PLAN 38**](#_Toc73405223)

# DANH MỤC CÁC HÌNH VẼ

[Hình 2‑1 Android Application Development 12](#_Toc73321242)

[Hình 2‑2 Kiến trúc hệ điều hành Android 14](#_Toc73321243)

[Hình 2‑3 Logo React Native 17](#_Toc73321244)

[Hình 2‑4 Sự khác biệt khi sử dụng Redux 19](#_Toc73321245)

[Hình 2‑5 Luồng hoạt động của Redux 20](#_Toc73321246)

[Hình 2‑6 Ứng dụng của .NET 23](#_Toc73321247)

[Hình 2‑7 Thư viện lớp .NET 24](#_Toc73321248)

[Hình 2‑8 Logo ngôn ngữ .NET 25](#_Toc73321249)

[Hình 2‑9 C# compiler 26](#_Toc73321250)

[Hình 2‑10 Logo MSSQL Server 28](#_Toc73321251)

[Hình 3‑1 Mock up đăng nhập 32](#_Toc73321252)

[Hình 3‑2 Mock up làm đề thi 32](#_Toc73321253)

[Hình 3‑3 Mock up cài đặt 33](#_Toc73321254)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1 Danh sách tác nhân và mô tả 31](#_Toc73405149)

[Bảng 3‑2 Danh sách các tình huống hoạt động 31](#_Toc73405150)

[Bảng 6‑1 Môi trường kiểm thử 38](#_Toc73405151)

[Bảng 6‑2 Rủi ro 38](#_Toc73405152)

[Bảng 6‑3 Danh sách các chức năng kiểm thử 39](#_Toc73405153)

[Bảng 6‑4 Kiểm thử module/ chức năng 40](#_Toc73405154)

[Bảng 6‑5 Kiểm thử giao diện 40](#_Toc73405155)

# CHƯƠNG 1: GIỚI THIỆU

## Tổng quan

**Tên đề tài:** Xây dựng app ôn thi tiếng anh vào lớp 10 trên di động (Android)

Xã hội ngày càng phát triển, nhu cầu ứng dụng công nghệ thông tin vào đời sống thực tiễn ngày càng đa dạng và phong phú hơn, Trong thời đại công nghệ số, mọi thứ trong cuộc sống từ sinh hoạt hay học tập, công việc đều có sự hỗ trợ bởi các thiết bị hiện đại có ứng dụng công nghệ thông tin nhằm giải quyết hiệu quả các công việc và tăng năng suất mà trước đây phải cần rất nhiều nguồn nhân lực mới có thể hoàn thành. Hơn thế nữa, “Cách mạng Công nghiệp 4.0 đang diễn ra tại nhiều nước phát triển và một trong số đó Việt Nam cũng đang nỗ lực bắt kịp để thay đổi bộ mặt nền kinh tế đất nước cũng như rút ngắn thời gian đi đến chủ nghĩa xã hội . Có thể nói, ứng dụng công nghệ thông tin vào đời sống hiện nay là không thể thiếu đối với bất cứ một quốc gia nào muốn phát triển kinh tế một cách nhanh chóng. Trong lĩnh vực giáo dục hiện nay việc học trực tuyến trên các ứng dụng đã và đang được phổ biến trên tất cả mọi nơi.

Trong tất cả các môn học thì Tiếng anh là một trong các môn bắt buộc trong chương trình học tập của tất cả học sinh trên cả nước. Vì vậy để hỗ trợ cho việc học tập của học sinh đạt hiệu quả tốt thì bên cạnh việc học trực tiếp với các thầy cô trên lớp học thì việc học sinh tự ôn tập là một điều cần thiết và quan trọng mà mỗi học sinh cần phải làm. Và việc chuyển giao giữa cấp 2 và cấp 3 là một chuyện quan trọng trên con đường học tập của mỗi học sinh để tiếp bước tới tương lai. Để có thể thi đạt được quả tốt thì ngoài việc học sinh phải ôn tập một cách chăm chỉ thì còn phải luyện giải những đề thi để có thể làm quen với các dạng đề các nhau. Ứng dụng ôn thi tiếng anh vào lớp 10 giúp các học sinh có thể làm quen với các đề đã được ra vào các năm trước trên các nơi. Điểm tiện lợi khi học sinh làm đề trên ứng dụng thì sẽ được thông báo kết quả mà học sinh đó vừa làm được. Ngoài ra ứng dụng có thể cập nhật bổ sung các đề mới để việc luyện tập và tiếp thu kiến thức thông qua các đề bài sẽ giúp học sinh tự tin hơn trong cuộc thi chính thức để đạt được kết quả tốt nhất.

## Mục tiêu đề tài

Nhằm giải quyết bài toán đặt ra tại phần tổng quan đề tài, mục tiêu của nhóm là xây dựng một app cho học sinh luyện tập các đề thi qua thiết bị di động với hệ điều hành Android bằng ngôn ngữ React, với phong cách thiết kế cũng như tối ưu hóa việc xử lý sẽ mang lại cho người dùng sự tiện dụng và nhẹ nhàng. Chỉ cần mang bên mình một chiếc điện thoại di động người dùng sẽ có thể luyện đề ở bất cứ đâu, bất cứ lúc nào. Để kết hợp và đồng bộ được dữ liệu, nhóm sẽ kết nối với dữ liệu qua API của C#.

Thiết bị di động sẽ thực hiện công việc như là một đề thi điện tử, ở đây sẽ hiển thị các đề thi đã được đưa ra ở các nơi trên cả nước giúp người dùng có thể lựa chọn các đề thi để làm thử. Khi hoàn thành một đề thi người dùng chỉ cần nhấn chọn hoàn thành thì ứng dụng sẽ tự động chấm điểm cho đề thi vừa được làm sau đó so sánh đáp án đúng sai để người dùng có thể dễ dàng biết được kết quả của đề thi. Ngoài ra ứng khi người dùng bắt đầu làm một đề hệ thống sẽ chủ động tính giờ khi hết 60 phút thì sẽ kết thúc đề thi giống như người dùng đang làm một bài thi thật.

## Phạm vi đề tài

Xây dựng app di động cho phép người dùng thực hành làm đề thi Tiếng anh sử dụng ngôn ngữ Java và IDE Android Studio để thực hiện đề tài. Ngoài ra sử dụng thêm các thư viện ngoài để giao diện trở nên thân thiện với người dùng.

Đề tài kết nối đồng bộ dữ liệu qua API. Những thay đổi của người dùng sẽ được trực tiếp cập nhật trong cơ sở dữ liệu và những thông tin mới mà admin đã cập nhật trong cơ sở dữ liệu thì người dùng cũng sẽ được cập nhật ngay tức thì.

## Mô tả yêu cầu chức năng

# : CƠ SỞ LÝ THUYẾT

## Android application [1]

### Giới thiệu về hệ điều hành android

Hiện nay, lĩnh vực công nghệ thông tin ngày càng phát triển, do đó việc trang bị cho mình những thiết bị Smartphone có hệ điều hành Android để hỗ trợ cho công việc trở nên hiệu quả hơn và nhẹ nhàng hơn. Nhưng để những chiếc mobile đó trở nên hữu ích nhất và được tin cậy nhất thì phải cần tích hợp những ứng dụng - phần mềm.



Hình ‑ Android Application Development

Android là hệ điều hành mã nguồn mở, dựa trên Linux Kernel, dành cho các thiết bị di động nói chung (điện thoại, máy tính bảng, đồng hồ thông minh, máy nghe nhạc,…). Có nghĩa là Android không chỉ giới hạn trong phạm vi một hệ điều hành cho điện thoại! Nó có thể được nhà sản xuất cài đặt lên đồng hồ, máy nghe nhạc, thiết bị định vị GPS, thậm chí là ô tô (các thiết bị Android Auto).

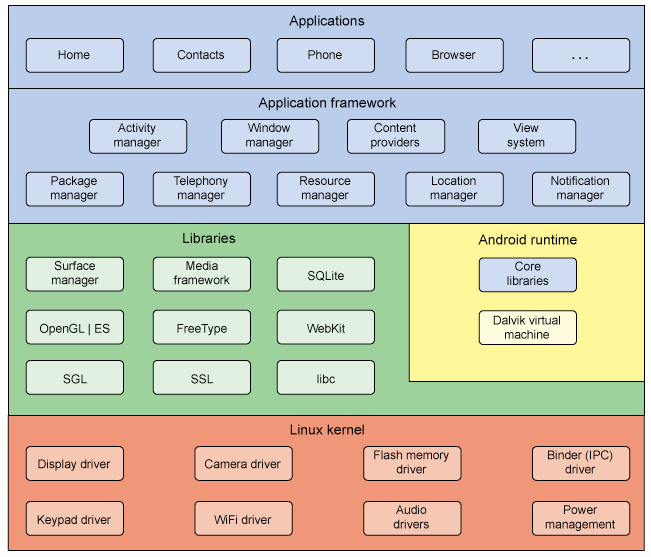
Android cũng không phải là một thiết bị hay sản phẩm cụ thể, nó là một hệ điều hành dựa trên Linux, nguồn mở, linh hoạt.

Hiện Android là một thương hiệu của Google. Có khả năng tùy biến rất cao và có thể chạy trên nhiều thiết bị, nhiều kiến trúc vi xử lý (ARM/ x86). Tính đến nay, Android đã có các phiên bản (kèm tên mã) lần lượt là:

* Android 1.5 Cupcake
* Android 1.6 Donut
* Android 2.1 Eclair
* Android 2.2 Froyo
* Android 2.3 Gingerbread
* Android 3.2 Honeycomb – Phiên bản Android đầu tiên được thiết kế cho máy tính bảng.
* Android 4.0 Ice Cream Sandwich
* Android 4.1 Jelly Bean
* Android 4.2 Jelly Bean
* Android 4.3 Jelly Bean
* Android 4.4 KitKat
* Android 5.0 Lollipop
* Android 6.0 Marshmallow
* Android 7.0 Nougat
* Android 8.0 Oreo
* Android 9.0 Pie
* Android 10 (Android Q)
* Android 11 (Android R)

### Kiến trúc hệ điều hành 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:



Hình 2‑2 Kiến trúc hệ điều hành Android

* **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.
* **Libraries:** 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 viện chơi và ghi âm audio, video, hay các thư viện SSL chịu trách nhiệm bảo mật Internet...

**Các thư viện Android:** 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. Một số thư viện chính của Android có thể kể đến như:

* 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 – được dùng 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 – được dùng 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.
* **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 Virtual 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 mọi 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.
* **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.
* **Applications:** Lớp trên cùng của kiến trúc là Application. Các ứng dụng tạo ra sẽ được cài đặt trên lớp này. Ví dụ như: danh bạ, nhắn tin, trò chơi,...

## React Native

### Tổng quan

React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.

Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái.

Giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là bạn không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ quá nhiều thời gian trong việc build và run ứng dụng nữa.

React Native sử dụng JS để phát triển được một ứng dụng di động hoàn chỉnh, đồng thời giải quyết được các vấn đề mà Native App gặp phải mà mình đã nêu ở trên. Và rồi còn cả kết hợp với code native như Swift, Java, v.v…



Hình ‑ Logo React Native

### Ưu điểm

Hiệu quả về mặt thời gian khi mà bạn muốn pháy triển một ứng dụng nhanh chóng. Hiệu năng tương đối ổn định. Cộng đồng phát triển mạnh, tiết kiệm được nhiều hơn về mặt kinh phí do chúng ta có thể lập trình trên 1 ngôn ngữ nhưng có thể chạy trên được cả hai hệ điều hành. Giúp trải nghiệm người dùng tốt hơn là hybrid app.

### Nhược điểm

Vẫn đòi hỏi native code, Hiệu năng sẽ thấp hơn so với app thuần native code. Về độ bảo mật thông tin không cao do dựa trên Js. Khả năng tùy biến cũng không được thực sự tốt đối với một vài module.

## ReactJs

### Tổng quan

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM.

Reactjs là một thư viện Javascript mã nguồn mở hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Bình thường các lập trình viên sẽ nhúng javascript vào code HTML thông qua các attribute như AngularJS nhưng với Reactjs làm việc như một thư viện cho phép nhúng HTML vào javascript thông qua JSX. Qua đó bạn có thể dễ dàng lồng các đoạn HTML vào trong JSX làm cho các component dễ hiểu và dễ sử dụng hơn.

### Ưu điểm

ReactJS giúp cho việc viết các đoạn code Javascript sẽ trở nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX. Thông qua JSX cho phép nhúng code HTML và Javascript.

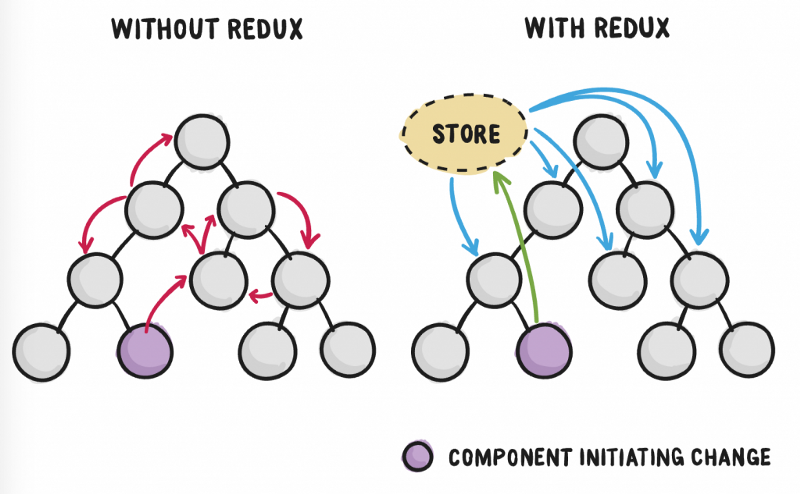
ReactJS cho phép Developer phá vỡ những cấu tạo UI phức tạp thành những component độc lập. Dev sẽ không phải lo lắng về tổng thể ứng dụng web, giờ đây Developer dễ dàng chia nhỏ các cấu trúc UI/UX phức tạp thành từng component đơn giản hơn. Đi kèm với ReactJS là rất nhiều các công cụ phát triển giúp cho việc debug code một cách dễ dàng hơn.

## Redux

### Tổng quan về Redux

**Redux** là một predictable state management tool cho các ứng dụng Javascript. Nó giúp bạn viết các ứng dụng hoạt động một cách nhất quán, chạy trong các môi trường khác nhau (client, server, and native) và dễ dàng để test. [**Redux**](https://topdev.vn/blog/?s=redux) ra đời lấy cảm hứng từ tư tưởng của ngôn ngữ **Elm** và kiến trúc **Flux**của Facebook. Do vậy Redux thường dùng kết hợp với React.

Sự khác biệt với việc sử dụng Redux và không sử dụng Redux được miêu tả bằng hình vẽ dưới đây :



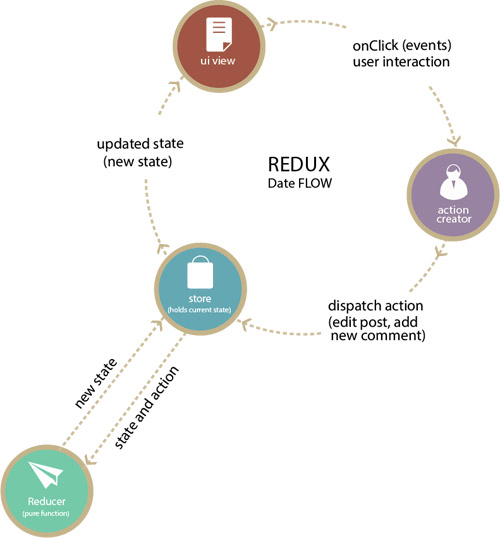
Hình ‑ Sự khác biệt khi sử dụng Redux

* Với việc không sử dụng Redux, các component sẽ giao tiếp với nhau bằng props. Nếu chúng ta cần lấy state của một component cách nhau 3 tầng, chúng ta phải gọi 3 lần, điều đó sẽ khiến việc code và quản lý state rất phức tạp, và to dần lên theo thời gian.
* Với việc sử dụng Redux, chúng ta sẽ lưu state của các component vào 1 store chung ở bên ngoài. Sau đó nếu muốn dùng ở component nào chúng ta chỉ cần gọi nó và sử dụng.

### Nguyên lý của Redux

Redux được xây dựng dựa trên 3 nguyên lý:

* Nguồn dữ liệu tin cậy duy nhất: State của toàn bộ ứng dụng được chứa trong một object tree nằm trong Store duy nhất. Trạng thái chỉ được phép đọc cách duy nhất để thay đổi State của ứng dụng là phát một Action (là 1 object mô tả những gì xảy ra).
* Thay đổi bằng hàm thuần túy: Để chỉ ra cách mà State được biến đổi bởi Action chúng ta dùng purefunction gọi là Reducer.
* Trạng thái chỉ có thể đọc: Chỉ có 1 cách duy nhất để thay đổi state đó là tạo ra một action (là 1 object mô tả những gì xảy ra).



Hình ‑ Luồng hoạt động của Redux

### Thành phần của Redux

Về cơ bản Redux có 3 thành phần như sau:

* Action: là các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.
* Reducer: là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.
* Store: lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Bạn có thể access các state được lưu, update state, và đăng ký or hủy đăng ký các listeners thông qua helper methods.
* View: Hiển thị dữ liệu được cung cấp bởi Store

### Ưu điểm của Redux

Khả năng dự báo về kết quả. (Predictability of outcome). Khả năng maintain bảo trì, nghiêm ngặt trong cấu trúc làm cho code trở nên dễ dàng bảo trì hơn. Khả năng tổ chức redux chặt chẽ hơn về việc tổ chức code, điều đó làm cho code trở nên nhất quán và dễ dàng hơn khi làm việc nhóm. Server rendering: Nó rất hữu ích đặc biệt là cho việc khởi tạo render, người dùng sẽ có trải nghiệm tốt hơn hoặc cho việc tối ưu bộ mấy tìm kiếm. Developer tools: Developer có thể theo dõi mọi thứ xảy ra trong ứng dụng từ các actions đến sự thay đổi của state. Dễ dàng test: Bởi Redux code hầu hết là các hàm nhỏ và độc lập.

## Axios

### Tổng quan về Axios

Axios là một HTTP client dựa được phát triển trên đối tượng [Javascript Promise](https://allaravel.com/tutorials/biet-tuot-javascript-promise-trong-60-giay/), nó có thể sử dụng trong các ứng dụng font-end Vue.js, React, Angular… Sử dụng Axios dễ dàng gửi đi các request HTTP bất đồng bộ đến các REST endpoint và thực hiện các chức năng CRUD.

Các khái niệm:

* HTTP client là có thể là phần mềm, thư viện có thể thực hiện các yêu cầu (request) dạng HTTP đến máy chủ HTTP và nhận về các hồi đáp (reponse). Đơn giản hơn bạn có thể coi nó gần với một trình duyệt web.
* Javascript Promise là một đối tượng giúp kiểm soát kết quả hoàn thành hay thất bại của một hành động bất đồng bộ trong Javascript (Tham khảo thêm [Kiến thức về Javascript Promise](https://allaravel.com/tutorials/biet-tuot-javascript-promise-trong-60-giay/)). [Vue.js](https://allaravel.com/tutorials/vuejs-framework/vue-js-la-gi/), React, Angular là những framework Javascript hiện đang rất hot, giúp xây dựng những ứng dụng font-end linh hoạt hoạt động nhanh và mạnh mẽ.
* REST endpoint là những điểm (URL) cung cấp các chức năng API cho phép chúng ta tương tác với một hệ thống, ví dụ khi chúng ta muốn tương tác với Lazada chúng ta có thể gửi các request HTTP đến các [REST API](https://allaravel.com/laravel-tutorials/co-ban-ve-rest-va-restfull-api/) do Lazada cung cấp.
* [CRUD](https://allaravel.com/laravel-tutorials/xay-dung-ung-dung-crud-su-dung-vue-js-trong-laravel/) viết tắt của Create, Read, Update, Delete là một thuật ngữ lập trình nói đến 4 phương thức quen thuộc khi làm việc với kho dữ liệu.

Axios giúp chúng ta thực hiện thao tác với các website hoặc API giúp xây dựng những ứng dụng font-end linh hoạt mạnh mẽ hơn.

Việc tạo ra một HTTP request dùng để fetch hay lưu dữ liệu là một trong những nhiệm vụ thường thấy mà một ứng dụng Javascript phía client cần phải làm khi muốn giao tiếp với phía server. Các thư viện bên thứ 3, đặc biệt là jQuery từ xưa đến nay vẫn là một trong những cách phổ biến để giúp cho các browser API tương tác tốt hơn, rõ ràng mạch lạc hơn và xóa đi những điểm khác biệt giữa các browser với nhau.

Khi ngày càng có nhiều developer thích dùng các native DOM API được nâng cấp và cải thiện hơn jQuery, hay các thư viện lập trình front-end UI như React và Vue.js thì việc sử dụng cả một thư viện jQuery chỉ với một giá trị sử dụng đó là tính năng của hàm $.ajax đã trở nên bất hợp lý hơn bao giờ hết.

### Đặc điểm của Axios

Một trong những đặc điểm lớn ta có thể thấy đó là cách mà thư viện xử lý với các [HTTP error code](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes#4xx_Client_errors) Axios sẽ reject tất cả các promise của request nếu một trong các mã lỗi trên được trả về.

Axios sẽ tự động gủi trả cookies về cho server khi tạo một request. Ta không cần phải truyền một cách trực tiếp các option để cho cookies có thể được include.

**Axios** được xây dựng dựa trên các XHR API cũ hơn, bạn có thể khai báo các hàm callback cho onUploadProgress và onDownloadProgress để hiện thị phần trăm thành công tại giao diện cho app của bạn.

**Axios** có thể được xử dụng ở cả browser và Node.js và chính điều này đã tạo cơ hội cho việc chia sẻ code Javascript giữa các trình duyệt và phần back-end hay việc thực hiện render cho ứng dụng front-end của bạn ở phía server.

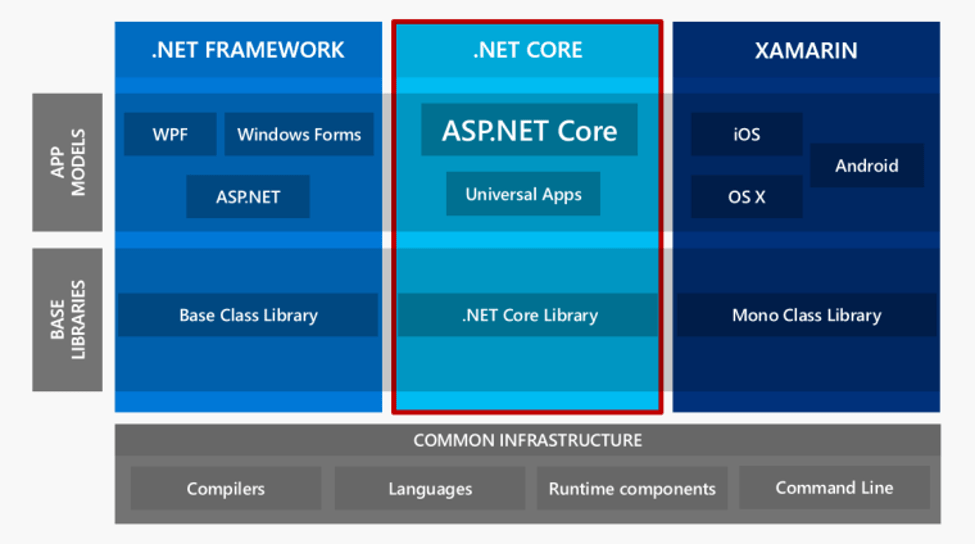
## Ngôn ngữ lập trình C#

### Nền tảng .NET

NET Framework được Microsoft đưa ra chính thức từ năm 2002. .NET Framework chỉ hoạt động trên Windows. Những nền tảng ứng dụng như WPF, Winforms, ASP.NET(1-4) hoạt động dựa trên .NET Framework.

Mono là phiên bản cộng đồng nhằm mang .NET đến những nền tảng ngoài Windows. Mono được phát triển chủ yếu nhằm xây dựng những ứng dụng với giao diện người dùng và được sử dụng rất rộng rãi: Unity Game, Xamarin…

Cho đến năm 2016, Microsoft định hướng đi đa nền tảng và phát triển .NET core. .NET core hiện được sử dụng trong các ứng dụng Universal Windows platform và ASP.NET Core. Từ đây, C# có thể được sử dụng để phát triển các loại ứng dụng đa nền tảng trên các hệ điều hành khác nhau (Windows, Linux, MacOS,…)



Hình 2‑6 Ứng dụng của .NET

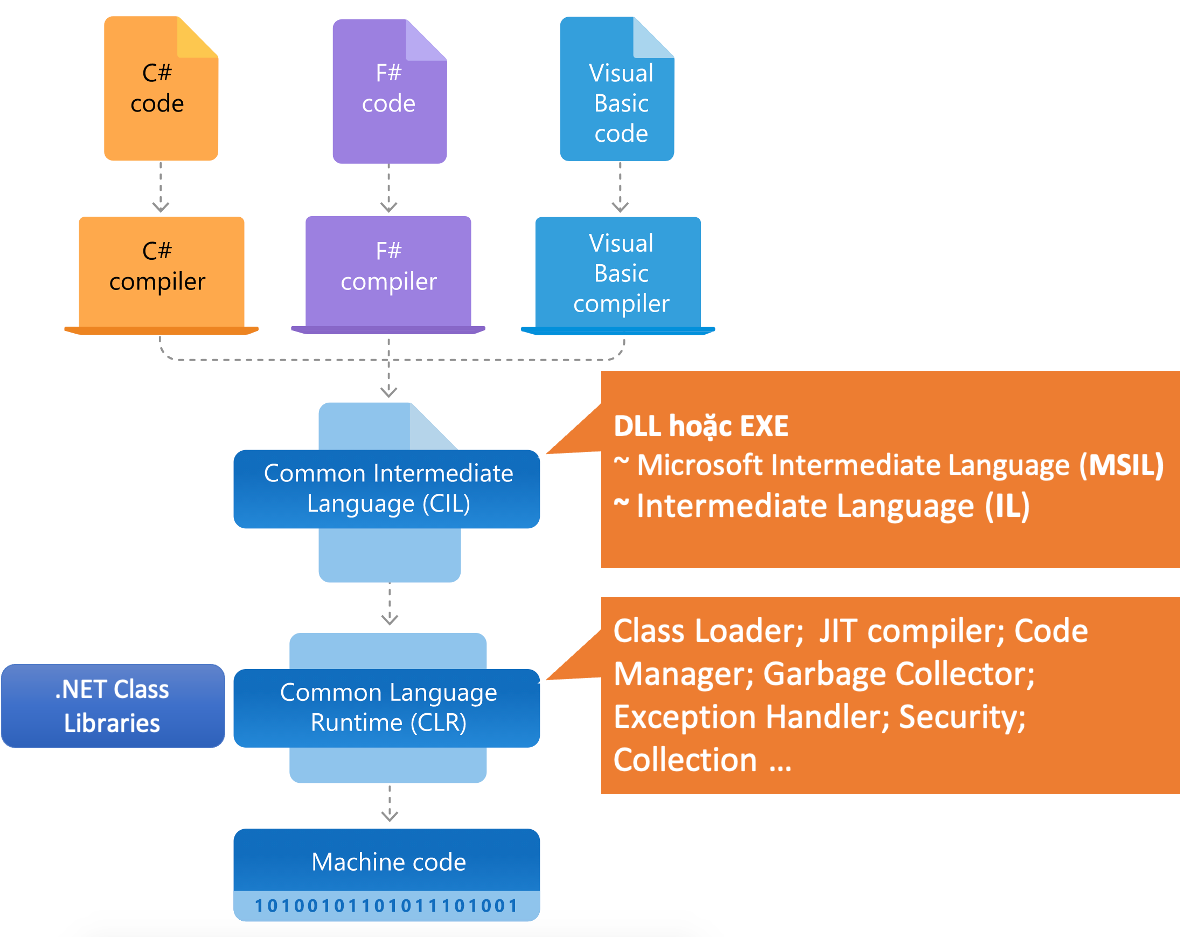
**.NET** bao gồm thành phần **CLR** (Common Language Runtime) và các thư viện lớp Framework Class Library (**FCL**).

Một chương trình viết cho .NET thì được thực thi (chạy) trong một môi trường do .NET tạo ra (phần mềm), nó có tên là Common Language Runtime (CLR) - có thể coi nó như một máy ảo cung cấp các dịch vụ về an ninh phần mềm (security), quản lý bộ nhớ (memory management), xử lý lỗi ngoại lệ (exception handling). Vì code viết cho .NET sẽ chạy trọng CLR nên code đó gọi là "managed code" (code được quản lý bởi .NET).

.NET bao gồm tập các thư viện lập trình lớn, và những thư viện này hỗ trợ việc xây dựng các chương trình phần mềm như lập trình giao diện; truy cập, kết nối cơ sở dữ liệu; ứng dụng web; các giải thuật, cấu trúc dữ liệu; giao tiếp mạng... CLR cùng với bộ thư viện này là 2 thành phần chính của.NET framework.

.NET framework đơn giản hóa việc viết ứng dụng bằng cách cung cấp nhiều thành phần được thiết kế sẵn, người lập trình chỉ cần học cách sử dụng và tùy theo sự sáng tạo mà gắn kết các thành phần đó lại với nhau. Nhiều công cụ được tạo ra để hỗ trợ xây dựng ứng dụng.NET, và IDE (Integrated Developement Environment) được phát triển và hỗ trợ bởi chính Microsoft là Visual Studio.

Thư viện lớp (FCL) là một tập hợp các lớp (class), giao diện (interface), kiểu giá trị, chúng cho phép bạn sử dụng để thi hành rất nhiều tác vụ trong lập trình: ví dụ như xử lý dữ liệu, truy cập file, làm việc với văn bản ... Từ thư viện lớp cung cấp sẵn này, các chương trình C# phát triển và mở rộng để thêm vào các chức năng khác nhau.



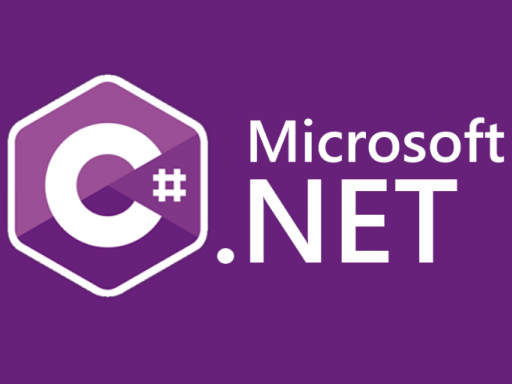
Hình 2‑7 Thư viện lớp .NET

### Tổng quan về ngôn ngữ C#

C# là một ngôn ngữ lập trình hiện đại được phát triển bởi Microsoft và được phê duyệt bởi European Computer Manufacturers Association (ECMA) và International Standards Organization (ISO). C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# được phát triển bởi Anders Hejlsberg và nhóm của ông trong việc phát triển .Net Framework.

C# được thiết kế cho các ngôn ngữ chung cơ sở hạ tầng (Common Language Infrastructure – CLI), trong đó bao gồm các mã (Executable Code) và môi trường thực thi (Runtime Environment) cho phép sử dụng các ngôn ngữ cấp cao khác nhau trên đa nền tảng máy tính và kiến trúc khác nhau.

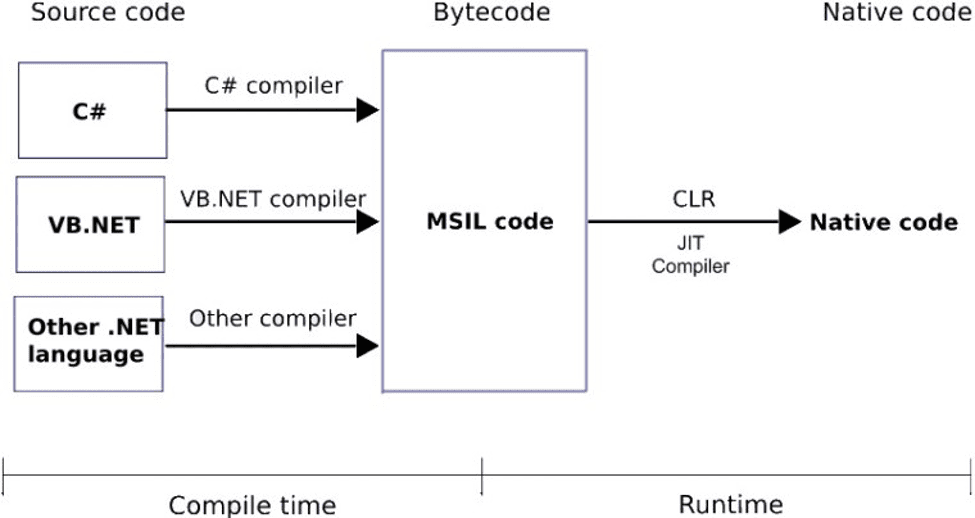


Hình 2‑8 Logo ngôn ngữ .NET

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.

Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.



Hình 2‑9 C# compiler

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

### Đặc trưng của ngôn ngữ lập trình C#

* **C# là ngôn ngữ đơn giản**

C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).

Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.

* **C# là ngôn ngữ hiện đại**

Điều gì làm cho một ngôn ngữ hiện đại? Những đặc tính như là xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng, và bảo mật mã nguồn là những đặc tính được mong đợi trong một ngôn ngữ hiện đại. C# chứa tất cả những đặc tính trên. Nếu là người mới học lập trình có thể chúng ta sẽ cảm thấy những đặc tính trên phức tạp và khó hiểu. Tuy nhiên, cũng đừng lo lắng chúng ta sẽ dần dần được tìm hiểu những đặc tính qua các nội dung khoá học này.

* **C# là một ngôn ngữ lập trình thuần hướng đối tượng**

Lập trình hướng đối tượng (OOP: Object-oriented programming) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên.

* **C# là một ngôn ngữ ít từ khóa**

C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

## Azure

### Tổng quan về Azure

Microsoft Azure được định nghĩa là một nền tảng điện toán đám mây và được phát hành vào năm 2010 bởi chuyên gia công nghệ Microsoft. Azure mang lại giải pháp Cloud tích hợp toàn diện để sử dụng cho việc xây dựng, triển khai cũng như quản lý cho các ứng dụng thông qua mạng lưới trung tâm toàn cầu của Microsoft.

Nó là nền tảng có phạm vi sử dụng các dịch vụ trực tuyến hoặc ứng dụng vô cùng rộng lớn. Giá cả của các dịch vụ này sẽ được tính dựa trên con số tiêu thụ tài nguyên từ phía người dùng.

### Các dịch vụ của Azure và cách hoạt động

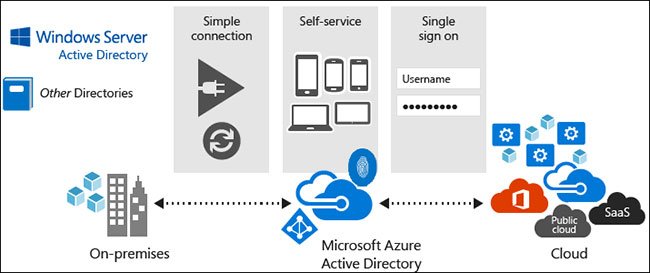
Azure cung cấp hơn 200 dịch vụ, được chia thành 18 loại. Các danh mục này bao gồm máy tính, mạng, lưu trữ, IoT, di chuyển, di động, phân tích, container, [trí tuệ nhân tạo và machine learning](https://quantrimang.com/su-khac-biet-giua-ai-hoc-may-va-hoc-sau-157948), tích hợp, công cụ quản lý, công cụ nhà phát triển, bảo mật, cơ sở dữ liệu, DevOps, nhận dạng media và web service. Hãy xem xét một số dịch vụ Azure chính theo danh mục:



Hình ‑ Azure sử dụng trên các thiết bị

* **Dịch vụ tính toán**
* Máy ảo: Dịch vụ này cho phép bạn tạo một máy ảo trong Windows, Linux hoặc bất kỳ cấu hình nào khác trong vài giây.
* Dịch vụ điện toán đám mây: Dịch vụ này cho phép bạn tạo các ứng dụng có thể mở rộng trong đám mây. Sau khi ứng dụng được triển khai, mọi thứ, bao gồm việc cung cấp, cân bằng tải và theo dõi sức khỏe, đều do Azure đảm nhận.
* Service Fabric: Với Service Fabric, quá trình phát triển một microservice được đơn giản hóa rất nhiều. Microservice là một ứng dụng có chứa các ứng dụng nhỏ, được đóng gói khác.
* Các hàm: Với các hàm, bạn có thể tạo ứng dụng bằng bất kỳ ngôn ngữ lập trình nào. Phần tốt nhất về dịch vụ này là bạn không cần phải lo lắng về các yêu cầu phần cứng trong khi phát triển ứng dụng, vì Azure sẽ lo điều đó. Tất cả những gì bạn cần làm là cung cấp code mà thôi.
* **Kết nối mạng**
* Azure CND: Azure CDN (Content Delivery Network) dùng để cung cấp nội dung cho người dùng. Nó sử dụng băng thông cao và nội dung có thể được truyền đến bất kỳ người nào trên toàn cầu. Dịch vụ CDN sử dụng một mạng lưới các máy chủ được đặt ở những vị trí chiến lược trên toàn cầu để người dùng có thể truy cập vào dữ liệu càng sớm càng tốt.
* Express route: Dịch vụ này cho phép bạn kết nối mạng tại chỗ của mình với đám mây Microsoft hoặc bất kỳ dịch vụ nào khác mà bạn muốn, thông qua kết nối riêng tư. Vì vậy, thông tin liên lạc duy nhất sẽ xảy ra ở đây sẽ là giữa mạng doanh nghiệp và dịch vụ mà bạn muốn.
* Mạng ảo: Mạng ảo cho phép bạn để bất kỳ dịch vụ Azure nào giao tiếp với nhau một cách riêng tư và an toàn.
* Azure DNS: Dịch vụ này cho phép bạn host các DNS domain hoặc domain hệ thống của mình trên Azure.
* **Lưu trữ**
* Disk storage: Dịch vụ này cho phép bạn chọn từ [HDD (Hard Disk Drive) hoặc SSD (Solid State Drive)](https://quantrimang.com/diem-khac-nhau-giua-o-cung-ssd-va-hdd-152904) làm tùy chọn lưu trữ cùng với máy ảo.
* Blob storage: Dịch vụ này được tối ưu hóa để lưu trữ một lượng lớn dữ liệu phi cấu trúc, bao gồm văn bản và thậm chí cả dữ liệu nhị phân.
* File storage: Đây là một dịch vụ lưu trữ file được quản lý, có thể được truy cập thông qua giao thức SMB (Server Message Block).
* Queue storage: Đây là dịch vụ lưu trữ số lượng lớn tin nhắn. Dịch vụ này có thể được truy cập từ bất kỳ đâu trên thế giới này.

### Lý do nên sử dụng Azure



Hình ‑ Lý do để sử dụng Azure

* **Phát triển ứng dụng**: Bạn có thể tạo bất kỳ ứng dụng web nào trong Azure.
* **Thử nghiệm**: Sau khi phát triển thành công một ứng dụng trên nền tảng, bạn có thể thử nghiệm nó.
* **Host ứng dụng**: Sau khi quá trình thử nghiệm hoàn tất, Azure có thể giúp bạn host ứng dụng.
* **Tạo máy ảo**: Bạn có thể tạo máy ảo ở bất kỳ cấu hình nào bạn muốn với sự trợ giúp của Azure.
* **Tích hợp và đồng bộ các tính năng**: Azure cho phép bạn tích hợp và đồng bộ các thiết bị và thư mục ảo.
* **Thu thập và lưu trữ số liệu**: Azure cho phép bạn thu thập và lưu trữ số liệu, có thể giúp bạn tìm thấy những gì hoạt động hiệu quả.
* **Ổ cứng ảo**: Đây là phần mở rộng của máy ảo. Chúng cung cấp một lượng lớn dung lượng lưu trữ dữ liệu.

### Lợi ích của việc sử dụng Azure

* **Azure là sản phẩm tuyệt vời dành cho bộ phận phát triển**

Azure có nhiệm vụ hỗ trợ các hệ điều hành, ngôn ngữ lập trình , thiết bị, khuôn khổ và cơ sở dữ liệu, như sau:

* Thực hiện đổi mới liên tục và tiến hành cung cấp những ứng dụng có chất lượng cao.
* Cung cấp quá trình trải nghiệm chéo cho thiết bị với sự hỗ trợ ở tất cả nền tảng di động.
* Chạy stack bất kỳ dựa trên Linux hoặc Windows nhờ các tính năng tiên tiến là: Kubernetes cluster trong Azure Container Service.
* **Azure là đám mây duy nhất**
* Bạn có thể mở rộng Azure ngay tại chỗ và thực hiện xây dựng ứng dụng lai tạo với Azure Stack.
* Thực hiện kết nối dữ liệu và ứng dụng ngay tại chỗ để vượt qua sự phức tạp cũng như tiến hành tối ưu hóa cho các tài sản hiện có.
* Phân phối và phân tích dữ liệu liền mạch trên đám mây và tại chỗ
* Azure là đám mây có thể xây dựng ứng dụng thông minh

Bạn có thể sử dụng Azure để tạo ra ứng dụng thông minh hoặc dữ liệu Bạn có thể sử dụng Azure và trí thông minh nhân tạo để cảm nhận các trải nghiệm mới mẻ trong việc học tập cũng như mô phỏng HPC.

* Phát những ứng dụng mới mẻ và đột phá với AI tích hợp.
* Triển khai và xây dựng những mô hình AI tùy biến trên dữ liệu bất kỳ.
* Kết hợp các yếu tố tốt nhất của Microsoft và dữ liệu mã nguồn mở cùng với đổi mới AI.



Hình ‑ Azure là giải pháp Điện toán đám mây nổi tiếng của Microsoft

* **Azure rất được tin tưởng**

90% công ty Fortune 500 đều tin tưởng dịch vụ [điện toán đám mây](https://blog.itnavi.com.vn/cloud-computing-la-gi/) của Microsoft. Azure có khả năng bảo mật, riêng tư minh bạch của Microsoft.

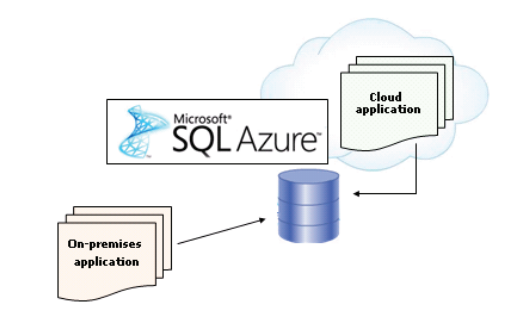
* Azure đạt quy mô toàn cầu với các trung tâm dữ liệu đều do chính Microsoft quản lý trên 44 khu vực đều đã được công bố.
* Azure có thể phát hiện và giảm thiểu các mối đe dọa đến tài nguyên Azure của bạn nhờ Trung tâm Bảo mật Azure.
* Azure dựa vào đám mây để tạo ra phạm vi tuân thủ toàn  diện nhất. Và nó được công nhận bởi các tổ chức từ chính phủ Hoa Kỳ.

## Cơ sở dữ liệu SQL Azure

### Tổng quan về SQL Azure

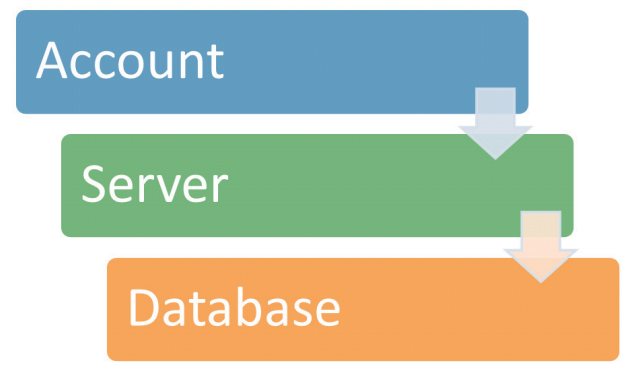
SQL Azure là một dịch vụ cơ sở dữ liệu quan hệ dựa trên đám mây, thúc đẩy các công nghệ SQL Server hiện có. Microsoft SQL Azure mở rộng chức năng của Microsoft SQL Server để phát triển các ứng dụng dựa trên web, có khả năng mở rộng và được phân phối. SQL Azure cho phép người dùng thực hiện các truy vấn quan hệ, hoạt động tìm kiếm và đồng bộ hóa dữ liệu với người dùng di động và các office từ xa. SQL Azure có thể lưu trữ và lấy cả dữ liệu có cấu trúc và phi cấu trúc.

Giao diện đơn giản của kiến trúc SQL Azure được trình bày



Hình ‑ Kiến trúc SQL Azure

Quy trình hoạt động của SQL Azure được giải thích trong mô hình như được trình bày bên dưới:



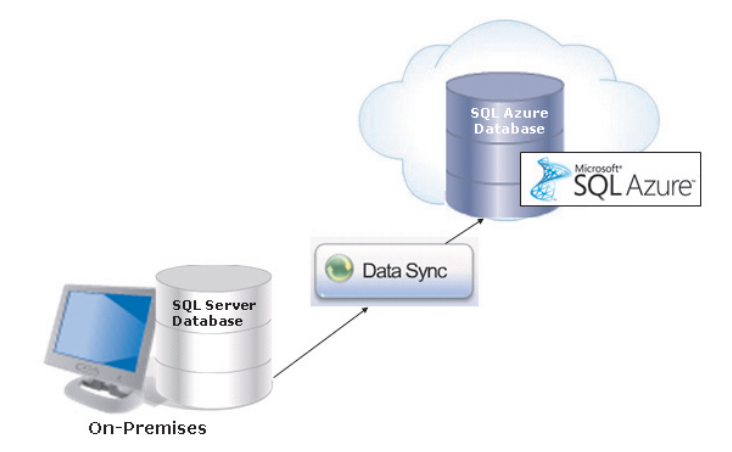
Hình ‑ Quy trình hoạt động của SQL Azure

Ba đối tượng cốt lõi trong mô hình hoạt động của SQL Azure như sau:

* **Tài khoản:** đầu tiên phải tạo một tài khoản SQL Azure. Tài khoản này được tạo ra cho mục đích thanh toán. Thuê bao tài khoản được ghi lại và đo lường, được tính tiền theo lượng sử dụng. Sau khi tài khoản người dùng được tạo ra, các yêu cầu cần phải được cung cấp cho cơ sở dữ liệu SQL Azure, bao gồm số lượng cơ sở dữ liệu cần thiết, kích thước cơ sở dữ liệu, v.v...
* **Server:** Máy chủ SQL Azure là đối tượng giúp tương tác giữa tài khoản và cơ sở dữ liệu. Sau khi tài khoản được đăng ký, cơ sở dữ liệu được cấu hình sử dụng máy chủ SQL Azure. Các thiết lập khác như thiết lập tường lửa và gán tên miền (DNS) cũng được cấu hình trong máy chủ SQL Azure.
* **Database:** Cơ sở dữ liệu SQL Azure lưu trữ tất cả dữ liệu theo cách tương tự như bất kỳ cơ sở dữ liệu SQL Server tại chỗ. Mặc dù lưu trữ bằng công nghệ đám mây, cơ sở dữ liệu SQL Azure có tất cả các chức năng của một RDBMS bình thường như table, view, query, function, thiết lập bảo mật, v.v...

Ngoài những đối tượng cốt lõi thì còn một đối tượng bổ sung trong SQL Azure. Đối tượng này là công nghệ Đồng bộ dữ liệu SQL Azure. Công nghệ Đồng bộ dữ liệu SQL Azure được xây dựng trên Microsoft Sync Framework và cơ sở dữ liệu SQL Azure.

**SQL Azure Data Sync** giúp đồng bộ hóa dữ liệu trên SQL Server cục bộ với các dữ liệu trên SQL Azure như được trình bày trong hình dưới:



Hình ‑ Đồng bộ hóa dữ liệu trên SQL Azure

Data Sync còn có khả năng quản lý dữ liệu giúp chia sẻ dữ liệu dễ dàng giữa các cơ sở dữ liệu SQL khác nhau. Data Sync không chỉ được sử dụng để đồng bộ hóa tại chỗ với SQL Azure, mà còn để đồng bộ hóa một tài khoản SQL Azure với tài khoản khác.

### Các lợi ích của SQL Azure

* **Chi phí thấp hơn**: SQL Azure cung cấp một số hàm tương tự như trên SQL Server tại chỗ với chi phí thấp hơn so với SQL Server tại chỗ. Ngoài ra, khi SQL Azure trên nền tảng đám mây, nó có thể được truy cập từ bất kỳ vị trí nào. Do đó, không có thêm chi phí cần thiết để phát triển một cơ sở hạ tầng CNTT chuyên dụng và phòng ban để quản lý cơ sở dữ liệu.
* **Sử dụng TDS:** TDS được sử dụng trong các cơ sở dữ liệu SQL Server tại chỗ cho các thư viện máy khách. Do đó, hầu hết các nhà phát triển đã quen thuộc với TDS và cách sử dụng tiện ích này. Cùng một loại giao diện TDS được sử dụng trong SQL Azure để xây dựng các thư viện máy khách. Do đó, các nhà phát triển làm việc trên SQL Azure dễ dàng hơn
* **Biện pháp chuyển đổi dự phòng tự động:** SQL Azure lưu trữ nhiều bản sao dữ liệu trên các vị trí vật lý khác nhau. Thậm chí khi có lỗi phần cứng do sử dụng nhiều hoặc tải quá mức, SQL Azure giúp duy trì các hoạt động kinh doanh bằng cách cung cấp khả năng sẵn sàng của dữ liệu thông qua các địa điểm vật lý khác.
* **Tính linh hoạt trong việc sử dụng dịch vụ:** Ngay cả các tổ chức nhỏ cũng có thể sử dụng SQL Azure bởi mô hình định giá cho SQL Azure được dựa trên khả năng lưu trữ được tổ chức sử dụng. Nếu tổ chức cần lưu trữ nhiều hơn, giá có thể thay đổi cho phù hợp với nhu cầu. Điều này giúp các tổ chức có được sự linh hoạt trong việc đầu tư tùy thuộc vào việc sử dụng dịch vụ.
* **Hỗ trợ Transact-SQL:** Do SQL Azure hoàn toàn dựa trên mô hình cơ sở dữ liệu quan hệ, nó cũng hỗ trợ các hoạt động và truy vấn Transact-SQL. Khái niệm này cũng tương tự như hoạt động của các SQL Server tại chỗ. Do đó, các quản trị viên không cần bất kỳ đào tạo hoặc hỗ trợ bổ sung nào để sử dụng SQL Azure

### Sự khác biệt giữa SQL Azure và SQL Server

Một số khác biệt quan trọng khác giữa SQL Azure và SQL Server phía khách hàng như sau:

* **Các công cụ** – SQL Server phía khách hàng cung cấp một số công cụ để theo dõi và quản lý. Tất cả những công cụ này có thể không được hỗ trợ bởi SQL Azure bởi có một số tập hợp công cụ hạn chế có sẵn trong phiên bản này
* **Sao lưu** – Sao lưu và phục hồi chức năng phải được hỗ trợ trong SQL Server phía khách hàng để khắc phục thảm họa. Đối với SQL Azure, do tất cả các dữ liệu là trên nền tảng điện toán đám mây, sao lưu và phục hồi là không cần thiết
* **Câu lệnh USE** – Câu lệnh USE không được SQL Azure hỗ trợ. Do đó, người dùng không thể chuyển đổi giữa các cơ sở dữ liệu trong SQL Azure so với SQL Server phía khách hàng.
* **Xác thực** – SQL Azure chỉ hỗ trợ xác thực SQL Server và SQL Server phía khách hàng hỗ trợ cả xác thực SQL Server và xác thực của Windows
* **Hỗ trợ Transact-SQL** – Không phải tất cả các chức năng Transact-SQL đều được SQL Azure hỗ trợ
* **Tài khoản và đăng nhập** – Trong SQL Azure, các tài khoản quản trị được tạo ra trong cổng thông tin quản lý Azure. Do đó, không có thông tin đăng nhập người dùng mức thể hiện cấp riêng biệt
* **Tường lửa** – Các thiết lập tường lửa cho các cổng và địa chỉ IP cho phép có thể được quản lý trên máy chủ vật lý cho SQL Server phía khách hàng. Bởi cơ sở dữ liệu SQL Azure có mặt trên điện toán đám mây, xác thực thông qua các thông tin đăng nhập là phương pháp duy nhất để xác minh người dùng.

# : PHÂN TÍCH VÀ ĐẶC TẢ CÁC YÊU CẦU CHỨ NĂNG

## Use case tổng quát

## Danh sách tác nhân và mô tả

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Người dùng | Người dùng đăng nhập vào ứng dụng để có thể thực hiện các chức năng trong ứng dụng. |
| Admin | Admin đăng nhập vào ứng dụng để có thể thực hiện các chức nặng quả ký tài khoản, quả ký dữ liệu. |

Bảng 3‑1 Danh sách tác nhân và mô tả

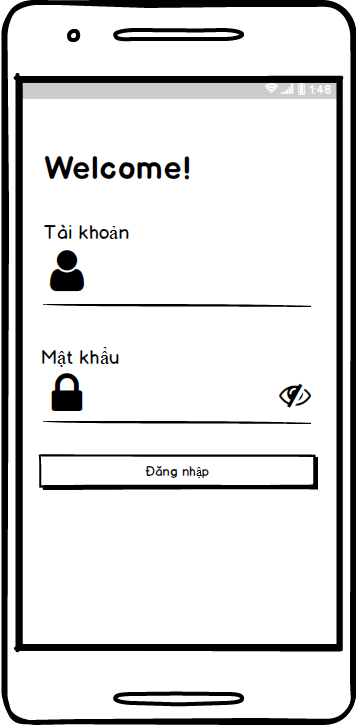
## Danh sách các tình huống hoạt động (Use case)

|  |  |
| --- | --- |
| ID | Tên use case |
| UC01 | Đăng nhập |
| UC02 | Đăng xuất |
| UC03 | Xem danh sách tài khoản(web) |
| UC04 | Thêm tài khoản(web) |
| UC05 | Đặt lại mật khẩu cho tài khoản(web) |
| UC06 | Xóa tài khoản(web) |
| UC07 | Xem danh sách đề thi |
| UC08 | Xem chi tiết đề thi(web) |
| UC09 | Cập nhập đề thi(web) |
| UC10 | Thêm đề thi(web) |
| UC11 | Xóa đề thi(web) |
| UC12 | Làm đề(mobile) |
| UC13 | Xem kết quả của từng bài thi(mobile) |
| UC14 | Hoàn thành bài thi(mobile) |

Bảng 3‑2 Danh sách các tình huống hoạt động

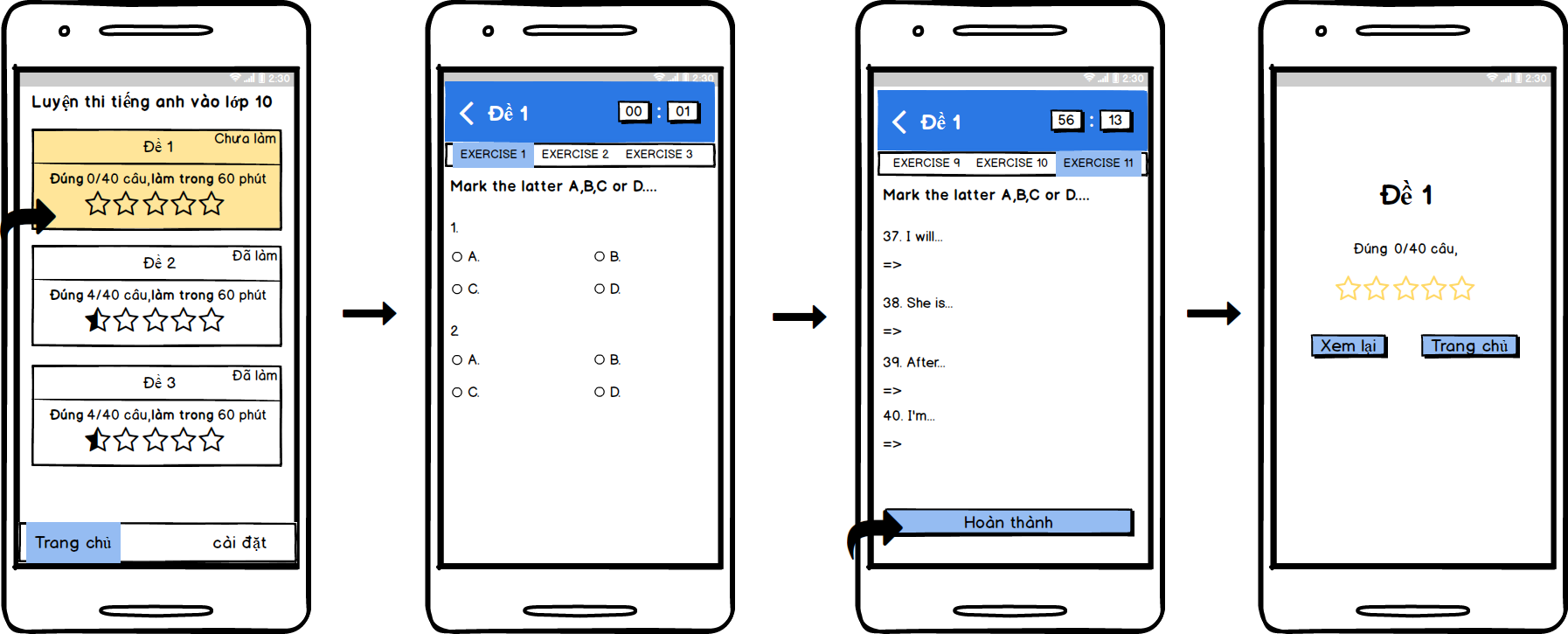
## Danh sách Mock up

### Mock up đăng nhập



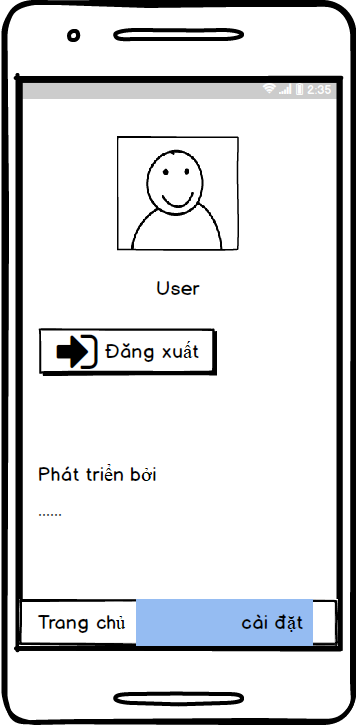
Hình ‑ Mock up đăng nhập

### Mock up làm đề thi



Hình ‑ Mock up làm đề thi

### Mock up cài đặt



Hình ‑ Mock up cài đặt

## Tình huống hoạt động

### Đặc tả UC01\_ Đăng nhập

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Đăng nhập | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) |  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC02\_ Đăng xuất

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) |  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC03\_Xem danh sách tài khoản(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Xem danh sách tài khoản(web) | | |
| Mục đích: | Admin có thể xem danh sách tài khoản người dùng. | |
| Mô tả: | Admin chọn chức năng xem danh sách tài khoản để xem danh sách tài khoản và có thể thực hiện quản lý tài khoản. | |
| Tác nhân | Admin | |
| Điều kiện trước: | Đăng nhập bằng tài khoản admin. | |
| Điều kiện sau: | Hiển thị danh sách tải khoản. | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
| 1.Click chọn chức năng quản lý user. | 2.Hiển thị danh sách các user. |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC04\_Thêm tài khoản(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** Thêm tài khoản(web) | | |
| Mục đích: | Tạo mới tài khoản cho người dùng. | |
| Mô tả: |  | |
| Tác nhân | Admin | |
| Điều kiện trước: | Đăng nhập bằng tài khoản admin, chọn chức năng quả lý người dùng. | |
| Điều kiện sau: | Tạo thành công tài khoản cho người dùng | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
| 1.Click chọn Thêm tài khoản.  3.Admin nhập đầy đủ thông tin tài khoản(user name, tên, password, confirm password, chọn quyền).  4.Người dùng chọn nút tạo tài khoản. | 2.Hệ thống chuyển qua giao diện thêm mới tài khoản  5.Hệ thống kiểm tra thộng tin hợp lệ không.  6.Thêm tài khoản mới vào cơ sở dữ liệu.  7.Thông báo tạo tại khoản thành công. |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC05\_Đặt lại mật khẩu cho tài khoản(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC06\_Xóa tài khoản(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC07\_Xem danh sách đề thi

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC08\_Xem chi tiết đề thi(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC09\_Cập nhập đề thi(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC10\_Thêm đề thi(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC11\_Xóa đề thi(web)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC12\_Làm đề(mobile)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC13\_Xem kết quả của từng bài thi(mobile)

#### Mô tả

|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

### Đặc tả UC14\_Hoàn thành bài thi(mobile)

#### Mô tả

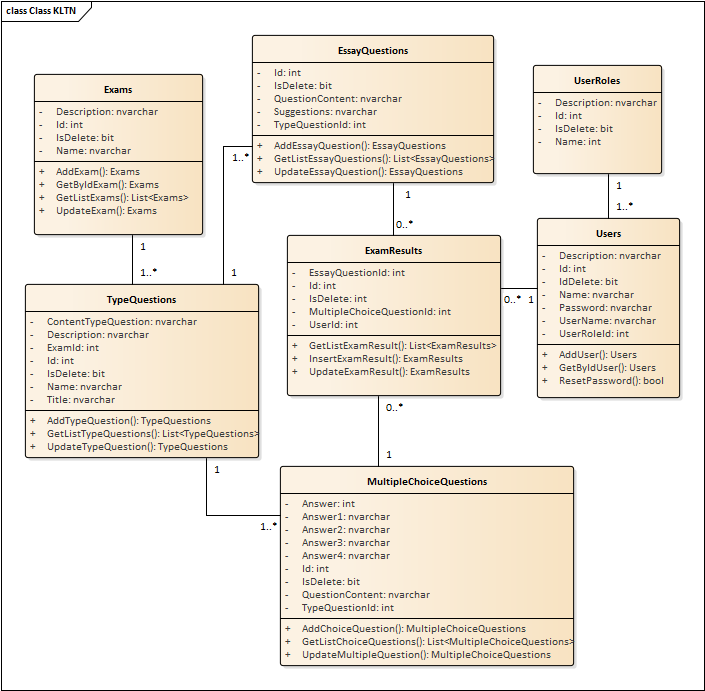
|  |  |  |
| --- | --- | --- |
| **Use case:** | | |
| Mục đích: |  | |
| Mô tả: |  | |
| Tác nhân |  | |
| Điều kiện trước: |  | |
| Điều kiện sau: |  | |
| Luồng sự kiện chính  (Basic flows) | Tác nhân | Hện thống |
|  |  |
| Luồng sự kiện  phụ  (Alternative Flows) |  |  |

#### Activity diagram

#### Sequence diagram

# : THIẾT KẾ VÀ HIỆN THỰC

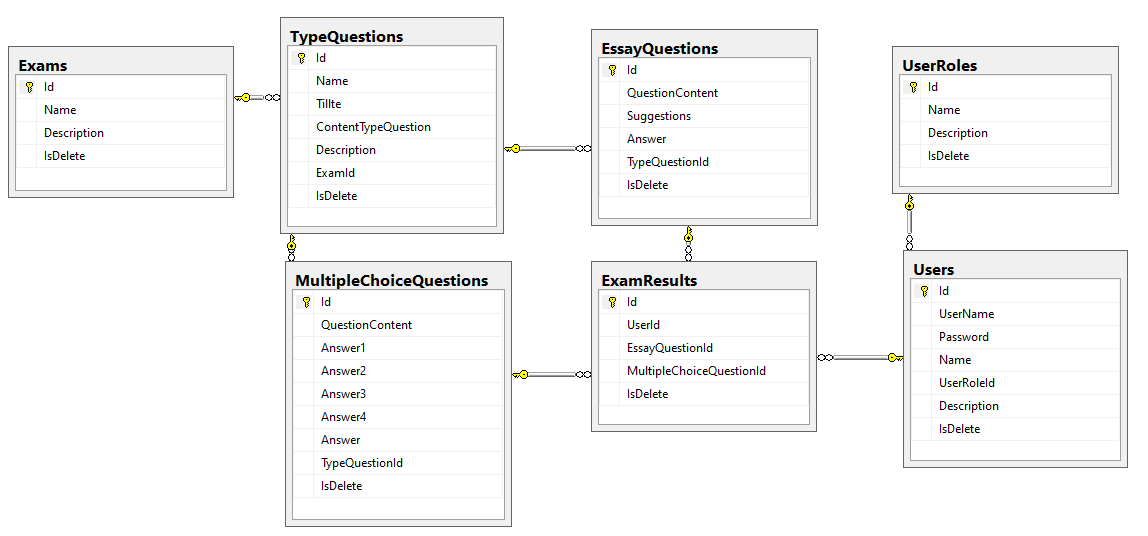
## Class diagram



Hình ‑ Class diagram

## Sitemap

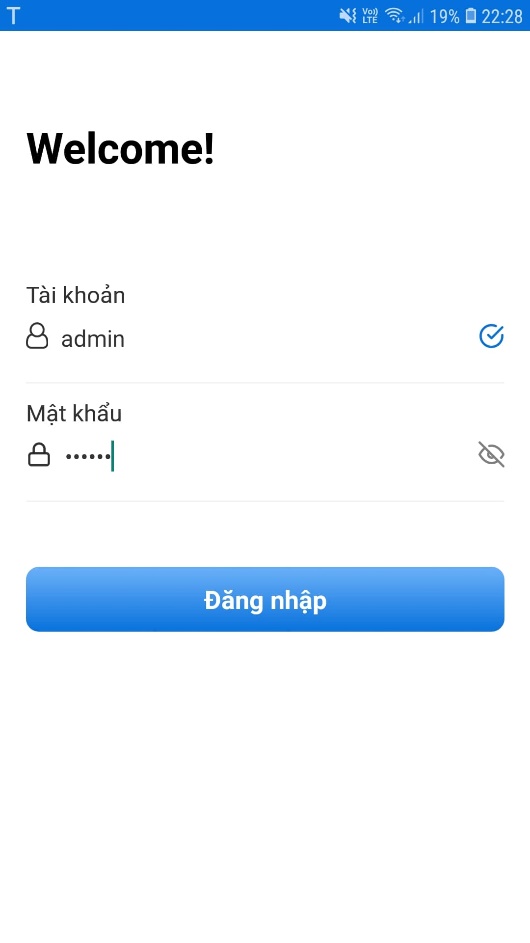
## Database diagram



Hình ‑ Database Diagram

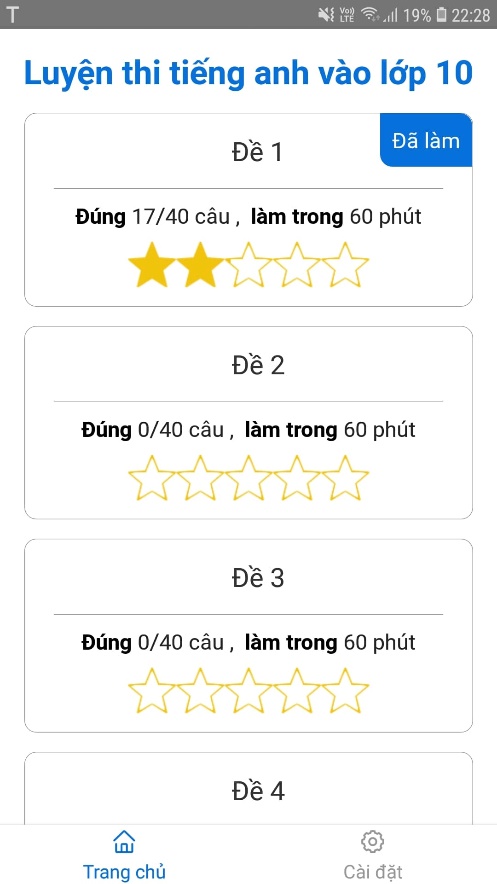
## Hiện thực

* **Giao diện app mobile**
* Màn hình đăng nhập

****

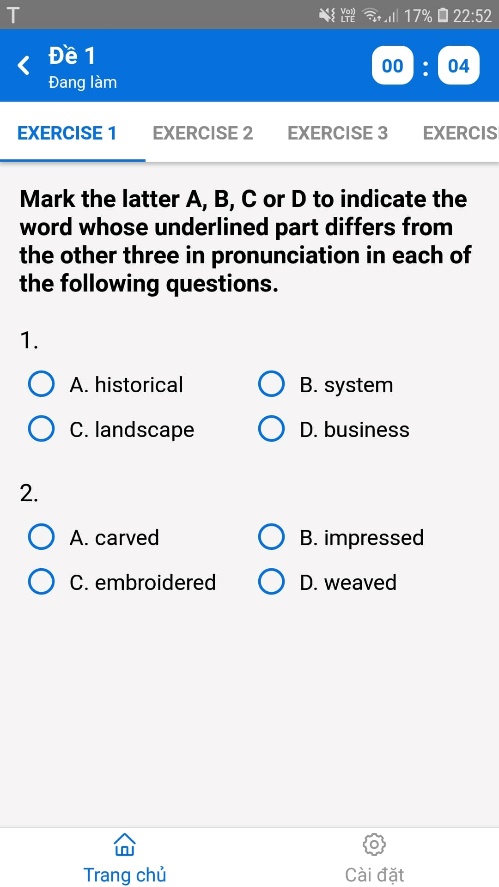
Hình ‑ Màn hình đăng nhập (mobile)

* Màn hình trang chủ hiển thị danh sách đề

****

Hình ‑ Màn hình trang chủ hiển thị danh sách đề thi

* Màn hình hiển thị chi tiết đề

****

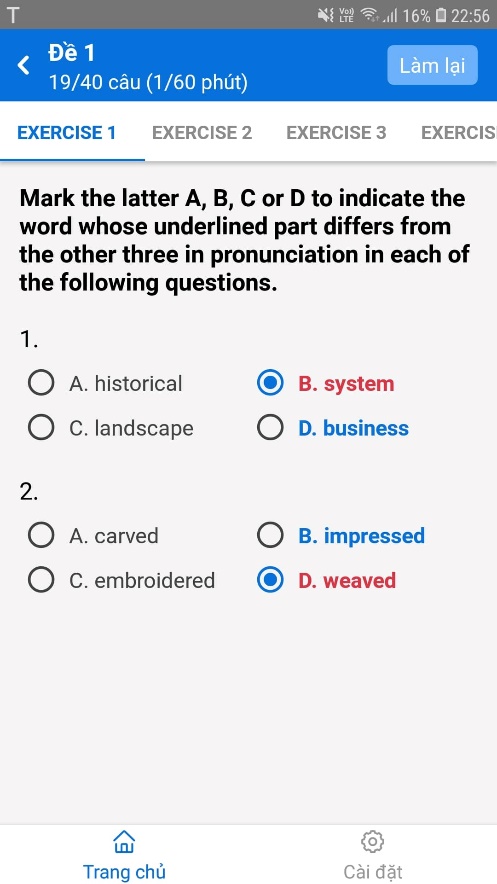
Hình ‑ Màn hình hiển thị chi tiết đề (mobile)

* Màn hình hiển thị thông tin khi làm hoàn thành đề

****

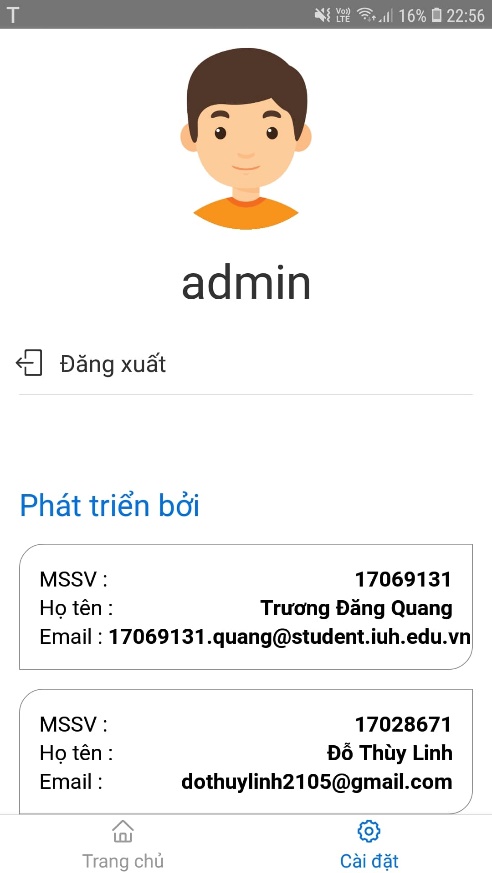
Hình ‑ Màn hình hiển thị thông tin khi làm hoàn thành đề

* Màn hình hiển thị chi tiết đáp án của đề (khi chọn Xem lại)

****

Hình ‑ Màn hình hiển thị chi tiết đáp án của đề (khi chọn Xem lại)

* Màn hình Cài đặt

****

Hình ‑ Màn hình Cài đặt (mobile)

# : KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

Tìm hiểu các công nghệ mới như:

* React Native

## Hạn chế của đồ án

Trình bày những hạn chế, công việc chưa hoàn tất trong đồ án

## Hướng phát triển

Trình bày các định hướng phát triển cho hệ thống hoặc hướng nghiên cứu trong tương lai.

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Việt

1. Giới thiệu lập trình Android và cài đặt môi trường (9/12/2019), <https://bitly.com.vn/wyt7e>.
2. Kiến trúc của hệ điều hành Android (12/9/2016), https://bitly.com.vn/cRP6X.

Các tài liệu Tiếng Anh

1. Android – Architecture, <https://bitly.com.vn/1vSbQ>.

Các tài liệu từ Internet

1. Website:https://docs.microsoft.com/en-us/sql/sql-server/
2. ...

# PHỤ LỤC: TEST PLAN

# Giới thiệu

# Mục tiêu

*Mục tiêu của kế hoạch kiểm thử:*

* Việc lập kế hoạch kiểm thử giúp phân chia công việc hợp lý cho các thành viên trong nhóm. Ngoài ra giúp đảm bảo việc thực hiện đầy đủ và đúng kế hoạch.
* Đảm bảo độ chính xác cao của các chức năng mobile application.
* Nâng cao mức độ tiện lợi và hiệu quả của phần mềm mobile.
* Giảm thiểu tối đa khả năng gặp rủi ro của phần mềm trong thời gian sử dụng.

# Phân tích ứng dụng

* *Đối tượng sử dụng mobile app:* người dùng
* *Mục tiêu của mobile app thực hiện các chức năng:*
* Xem thông tin chi tiết các dạng đề thi
* Thi thử đối với các dạng đề thi
* *Hoạt động của người dùng mobile app:* người dùng có thể xem chi tiết các đề thi đã được đưa ra tại các tỉnh thành khác nhau và thi thử với các đề đó.
* *Phần mềm/phần cứng/thiết bị mà ứng dụng sử dụng:*
* Phần mềm: Android Studio, Visual Studio 2019, SQL Server Management Studio, Visual Studio Code, Posman
* Phần cứng: mobile có kết nối mạng internet hoặc wifi

# Phạm vi kiểm thử

* Theo yêu cầu ứng dụng, đề tài app ông thi tiếng anh vào lớp 10 trên di động chỉ tập trung vào kiểm thử tất cả các chức năng và giao diện bên ngoài của app.
* Không thực hiện kiểm thử mức độ hiệu suất ứng dụng, logic cơ sở dữ liệu.

# Các ràng buộc về quy trình kiểm thử

* *Môi trường kiểm thử, các điều kiện liên quan:*

|  |  |  |
| --- | --- | --- |
| **STT** | **Tài nguyên** | **Mô tả** |
| 1 | Mạng | Mạng wifi hoặc internet |
| 2 | Thiết bị di động | Chạy hệ điều hành android 6.0 trở lên |

Bảng 6‑ Môi trường kiểm thử

* *Các ràng buộc về tài nguyên, lịch trình, công cụ:*
* Dùng ngôn ngữ React Native là ngôn ngữ chính cho việc xây dựng app android
* Sử dụng Visual Studio Code thực hiện code app mobile
* Dùng ngôn ngữ C# là ngôn ngữ chính để xây dựng các API
* Sử dụng Visual Studio 2019 thực hiện code API
* Cơ sở dữ liệu SQL Server 2012
* Dùng Posman để kiểm tra các API

# Rủi ro

|  |  |
| --- | --- |
| **Rủi ro** | **Giải pháp** |
| Các thành viên nhóm còn thiếu các kinh nghiệm về kiểm thử. | Tự học hỏi, tìm hiểu và đọc lại giáo trình môn Đảm bảo chất lượng và kiểm thử phần mềm để bổ sung kiến thức kiểm thử |
| Tiếp cận công nghệ mới; gây khó khăn trong quá trình hoàn thiện đề tài. | Ưu tiên các công việc kiểm thử cần thiết theo trình tự. |
| Trình độ cá nhân của các thành viên trong nhóm có sự chênh lệch nhất định dẫn đến khó khăn trong việc phân chia nhiệm vụ. | Họp nhóm, phân chia lại chức năng phù hợp với khả năng của từng thành viên. |

Bảng 6‑ Rủi ro

# Yêu cầu kiểm thử

* *Danh sách các chức năng kiểm thử*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả vắn tắt chức năng(outline)** | **Ước lượng số lượng tình huống kiểm kiểm thử (test case).** | **Ghi chú** |
| 1 | Đăng nhập | Người dùng thực hiện đăng nhập vào ứng dụng để tiến hành thực hiện các chức năng | 3 |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

Bảng 6‑ Danh sách các chức năng kiểm thử

* *Điều kiện chấp nhận*

Danh sách các tiêu chí để xác định mức chất lượng kiểm thử là đủ để chuyển sang giai đoạn thử nghiệm tiếp theo

* + Phạm vi bao phủ của Test (Test coverage)
  + Successful Test coverage
  + Số lượng các trường hợp kiểm thử (Đơn vị/ Tích hợp/ Các trường hợp thử nghiệm hệ thống).
  + Số lượng lỗi/ Trọng số lỗi

# Kỹ thuật kiểm thử

# Kiểm thử đơn vị

# Kiểm thử module/chức năng

|  |  |
| --- | --- |
| **Mục tiêu** | Đảm bảo chức năng kiểm thử với mục tiêu thích hợp, bao gồm dữ liệu đầu vào, navigation, quá trình xử lý và kết quả nhận được. |
| **Kỹ thuật** | Dùng các vùng dữ liệu đúng, vùng dữ liệu không đúng, giá trị biên thực hiện trên mỗi use case, để xác định:   * Kết quả mong muốn khi dữ liệu đưa vào là đúng. * Các thông báo lỗi, cảnh báo hiển thị khi dữ liệu không chính xác đưa vào. * Quy tắc nghiệp vụ áp dụng cho trường hợp test. * Sử dụng các công cụ kiểm thử (test tool) |
| **Tiêu chí hoàn thành chức năng** | * Tất cả các kế hoạch kiểm thử cho chức năng được thực hiện. * Chỉnh sửa các lỗi đã phát hiện. |
| **Điều kiện đặc biệt** | Xác định hoặc mô tả các mục hoặc vấn đề (bên trong hoặc bên ngoài) mà có ảnh hưởng đến việc triển khai và thực hiện kiểm thử chức năng. |

Bảng 6‑ Kiểm thử module/ chức năng

# Kiểm thử giao diện

|  |  |
| --- | --- |
| **Mục tiêu** | Kiểm thử các vấn đề:  Di chuyển từ tab này sang tab khác, từ field này sang field khác. Đối tượng cửa sổ, menu, kích cỡ, vị trí, trạng thái. |
| **Kỹ thuật** | Tạo hoặc sửa đổi các kiểm thử cho mỗi cửa sổ để xác minh điều hướng và trạng thái đối tượng thích hợp cho từng cửa sổ ứng dụng. |
| **Tiêu chí hoàn thành chức năng** | Mỗi cửa sổ phải được xác minh thành công để phù hợp với phiên bản chuẩn hoặc trong tiêu chuẩn chấp nhận được. |
| **Điều kiện đặc biệt** | Để thực hiện các chức năng cần có tài khoản đăng nhập không phải đối tượng bất kì không có tài khoản mà sử dụng được |

Bảng 6‑ Kiểm thử giao diện

# Kiểm thử hệ thống

Kiểm thử hệ thống thuộc loại Black Box Testing

* Kiểm thử hệ thống được thực hiện khi hệ thống đã được tích hợp đầy đủ các chức năng bao gồm cả các thiết bị bên ngoài, kiểm thử các thành phần tương tác với nhau và với toàn bộ hệ thống.
* Tạo kịch bản kiểm thử cuối cùng: kiểm thử mọi đầu vào và đầu ra mong muốn.
* Kiểm thử trải nghiệm của người sử dụng với ứng dụng.

# Kế hoạch nguồn nhân lực thực hiện kiểm thử cho ứng dụng

# 