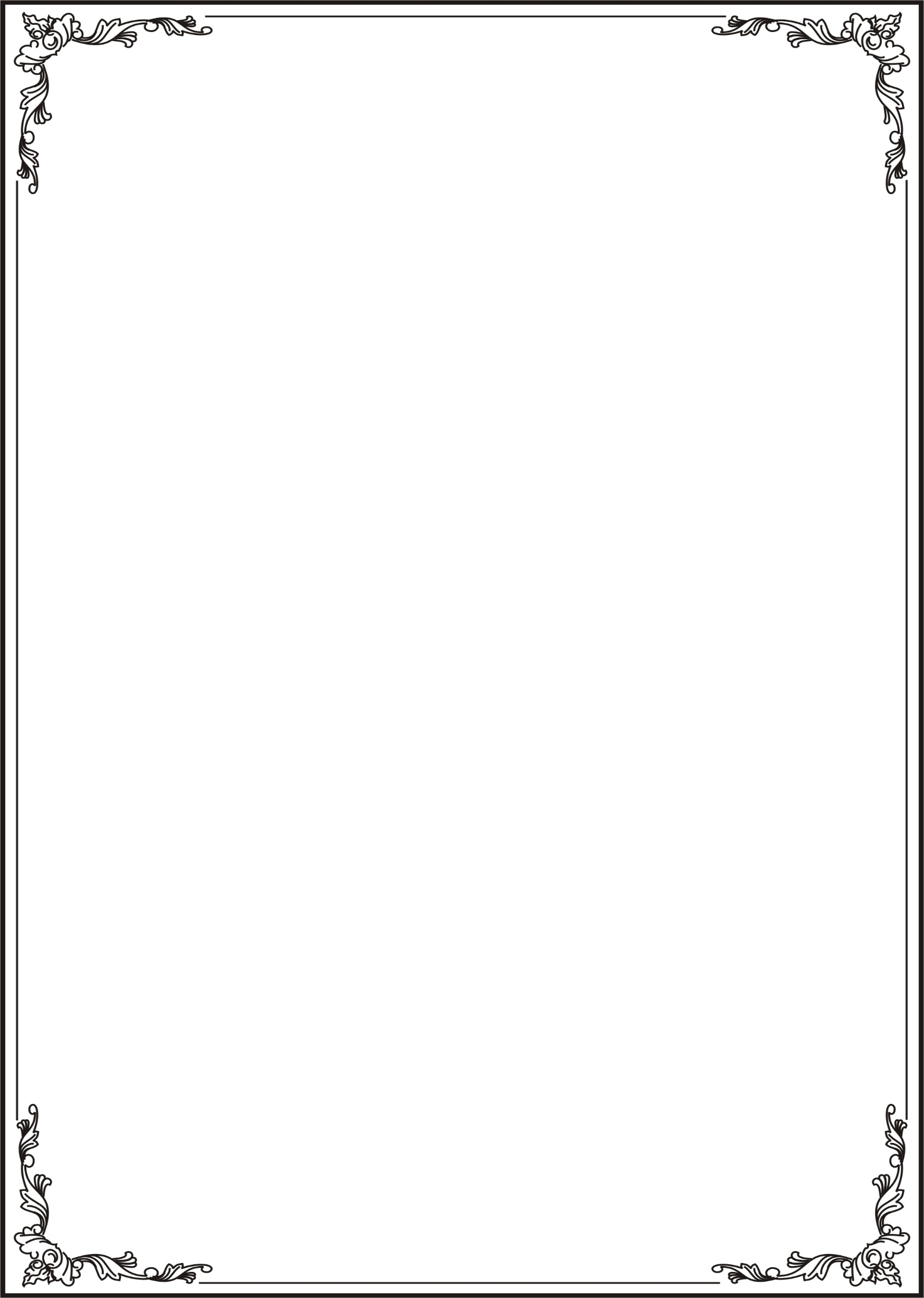
**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**QUẢN LÝ DỰ ÁN CNTT**

**Giảng viên hướng dẫn:**

NGUYỄN THỊ THANH TRÚC

**Sinh viên thực hiện:**

LÊ VĂN PHÚ – 13520628

NGUYỄN MINH TRÍ - 13520918

NGÔ SƠN LÂM – 13520429

NGUYỄN HOÀNG HIỆP - 13520265

**Lớp:** SE340.G11

*TP. Hồ Chí Minh, ngày 10 tháng 1 năm 2016*

**Lời Nói Đầu**

* Hiện nay, công nghệ thông tin đang phát triển hết sức mạnh mẽ. Điều này đồng nghĩa với việc nhu cầu về công nghệ của con người cũng ngày càng cao. Từ việc phải làm việc và giải trí trên máy tính, giờ đây các thiết bị di dộng cũng đang dần đáp ứng được các nhu cầu của con người và càng được dùng nhiều hơn do tính linh hoạt, tiện lợi và di động cao. Ngôn ngữ ngày càng đa dạng hơn, thông tin cần được chuyển đi chính xác hơn, người Việt chúng ta cần phần mềm hỗ trợ cho việc tra ngôn ngữ. Phần mềm Từ Điển từ đó đã ra đời, nó giúp cho người Việt dễ đọc, dễ thông tin. Từ khi phần mềm Từ Điển ra đời, nó là một phần mềm rất cần thiết cho chúng ta.
* Từ điển là một loại [phần mềm](http://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) hỗ trợ tra các ngôn ngữ trên [máy tính](http://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh), thường cần phải có phông ký tự [chữ Quốc ngữ](http://vi.wikipedia.org/wiki/Qu%E1%BB%91c_ng%E1%BB%AF) đã được cài đặt trong máy tính. Các từ điển khác nhau sẽ hỗ trợ các ngôn ngữ khác nhau.
* Nhờ có sự hỗ trợ của từ điển, ngày nay, chúng ta có thể hiểu được các đoạn văn bản trang trọng, có thể thông tin chính xác, có thể đọc hiểu văn bản một cách dễ dàng và trên hết nhờ vậy mà chúng ta có thể giữ được ngôn ngữ riêng của chính chúng ta.
* Hiện nay, người sử dụng tại Việt Nam đang dùng phổ biến các phần mềm từ điển như Lạc Việt, Babylon, Lingoes. Tuy nhiên, sử dụng các phần mềm này phải trả phí bản quyền, chỉ có từ điển Lingoes miễn phí cho mục đích học ngoại ngữ, nhưng phải trả phí nếu dùng cho công việc. Một số người chọn cách tra từ điển trực tuyến miễn phí trên các website, nhưng đây là cách không phù hợp cho nhu cầu tra cứu thường xuyên, đặc biệt không thể sử dụng được trong trường hợp bị mất kết nối mạng.
* Có rất nhiều nền tảng di động phổ biến như IOS, Android, Symbia, WindowPhone, Bada… nhưng Android của Google đang là hệ điều hành có hướng phát triển mạnh nhất. Do Android là một hệ điều hành mở nên nhận được rất nhiều sự hỗ trợ từ cộng động mạng trên khắp thế giới. Lập trình di động đã phát triển khá lâu trên thế giới nhưng vẫn còn khá mới mẻ ở Việt Nam, đặc biệt là lập trình ứng dụng trên Android. Từ những lý do trên, em đã suy nghĩ và quyết định lựa chọn đề tài tìm hiểu và xây dựng ra ứng dụng tra cứu Từ điển chạy trên hệ điều hành Android, tên đề tài là “XÂY DỰNG ỨNG DỤNG TỪ ĐIỂN ĐA NĂNG TRÊN ANDROID”.

MỤC LỤC

[1. Giới thiệu: 4](#_Toc439979539)

[1.1. Giới thiệu về đề tài: 4](#_Toc439979540)

[1.2. Quy trình thực hiện các công việc chính: 4](#_Toc439979541)

[1.3. Mục tiêu: 4](#_Toc439979542)

[1.4. Chức năng chính: 4](#_Toc439979543)

[2. Phân tích đề tài: 5](#_Toc439979544)

[2.1. PARSE: 5](#_Toc439979545)

[**2.1.1. Parse là gì: 5**](#_Toc439979546)

[**2.1.2. Các dịch vụ của Parse: 6**](#_Toc439979547)

[**2.1.3. Bảng điều khiển Parse: 9**](#_Toc439979548)

[**2.1.4. Các đối tượng dữ liệu Parse: 10**](#_Toc439979549)

[**2.1.5. Các đối tượng dữ liệu Parse trên đám mây 11**](#_Toc439979550)

[2.2. API: 12](#_Toc439979551)

[**2.2.1. API là gì: 12**](#_Toc439979552)

[**2.2.2. API Bing 14**](#_Toc439979553)

[**2.2.3. API FaceBook 17**](#_Toc439979554)

[2.3. Xử lý tập tin trong Android 18](#_Toc439979555)

[**2.3.1. Internal Storage 18**](#_Toc439979556)

[**2.3.2. External Storage 19**](#_Toc439979557)

[**2.3.3. Saving Cache Files 20**](#_Toc439979558)

[3. Thiết kế 20](#_Toc439979559)

[3.1. Cơ sở dữ liệu 20](#_Toc439979560)

[3.2. Kiến trúc chương trình 21](#_Toc439979561)

[**3.2.1. Activity\_Login: 21**](#_Toc439979562)

[**3.2.2. Activity\_Login\_Normal 21**](#_Toc439979563)

[**3.2.3. Activity\_Resgister 21**](#_Toc439979564)

[**3.2.4. DictPlusApp 22**](#_Toc439979565)

[**3.2.5. FragmentDrawer 22**](#_Toc439979566)

[**3.2.6. FragmentStudy 22**](#_Toc439979567)

[**3.2.7. MainActivity 22**](#_Toc439979568)

[**3.2.8. SpeechEnglish 22**](#_Toc439979569)

[**3.2.9. TabAdapter: 22**](#_Toc439979570)

[**3.2.10. Fragment\_TraTu: 22**](#_Toc439979571)

[**3.2.11. Tab\_AnhViet 22**](#_Toc439979572)

[**3.2.12. Tab\_VietAnh 22**](#_Toc439979573)

[**3.2.13. Tab\_Online: 22**](#_Toc439979574)

[**3.2.14. FragmentAllHistory 23**](#_Toc439979575)

[**3.2.15. FragmentHistory 24**](#_Toc439979576)

[**3.2.16. History 24**](#_Toc439979577)

[**3.2.17. HistoryAdapter 25**](#_Toc439979578)

[**3.2.18. PopupEditHistory 25**](#_Toc439979579)

[**3.2.19. TabPublic: 26**](#_Toc439979580)

[**3.2.20. DataAdapter 26**](#_Toc439979581)

[**3.2.21. DataOffline 26**](#_Toc439979582)

[**3.2.22. Popup\_TraTu 26**](#_Toc439979583)

[**3.2.23. Constants 26**](#_Toc439979584)

[**3.2.24. DataTracNghiem 26**](#_Toc439979585)

[**3.2.25. DauNhan 26**](#_Toc439979586)

[**3.2.26. DichNhanh: 26**](#_Toc439979587)

[**3.2.27. FloatingWindow\_NhacNho 27**](#_Toc439979588)

[**3.2.28. FragmentTracNghiem 28**](#_Toc439979589)

[**3.2.29. TestSpeechVoca 28**](#_Toc439979590)

[**3.2.30. ActivityQuestion 28**](#_Toc439979591)

[**3.2.31. Activity\_Comment 29**](#_Toc439979592)

[**3.2.32. AdapterComment 29**](#_Toc439979593)

[**3.2.33. ParseComment 29**](#_Toc439979594)

[**3.2.34. Activity\_ListQuestion 29**](#_Toc439979595)

[**3.2.35. ActivitySearchImg 29**](#_Toc439979596)

[**3.2.36. ThumbnailAdapter: 30**](#_Toc439979597)

[2.1. Giao diện 30](#_Toc439979598)

[**2.1.1. Danh sách các màn hình 30**](#_Toc439979599)

[**2.1.2. Màn hình Tra Từ 32**](#_Toc439979600)

[**2.1.3. Màn hình Lịch Sử 33**](#_Toc439979601)

[**2.1.4. Màn hình Học Từ Vựng 34**](#_Toc439979602)

[**2.1.5. Màn hình Trắc Nghiệm 35**](#_Toc439979603)

[**2.1.6. Màn hình Kiểm Tra Phát Âm 36**](#_Toc439979604)

[**2.1.7. Màn hình Hỏi Đáp 37**](#_Toc439979605)

[**2.1.8. Màn hình Tìm Ảnh Trên Ping 38**](#_Toc439979606)

[**2.1.9. Màn hình Danh Sách Các Câu Hỏi 39**](#_Toc439979607)

[**2.1.10. Màn hình Đăng Nhập 40**](#_Toc439979608)

[**2.1.11. Màn hình Đăng Ký 41**](#_Toc439979609)

[**2.1.12. Màn hình Chờ 42**](#_Toc439979610)

[**2.1.13. Màn hình Menu 43**](#_Toc439979611)

[3. Cài đặt và thử nghiệm: 44](#_Toc439979612)

[4. Hướng phát triển: 44](#_Toc439979613)

[5. Đánh giá - Kết luận: 44](#_Toc439979614)

[6. Mục tiêu - Giới hạn: 44](#_Toc439979615)

[6.1. Mục tiêu: 45](#_Toc439979616)

[6.2. Giới hạn: 45](#_Toc439979617)

[7. Tài liệu tham khảo: 45](#_Toc439979618)

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**QUẢN LÝ DỰ ÁN CNTT**

1. **Giới thiệu:**

## Giới thiệu về đề tài:

* Xây dựng phần mềm từ điển đa năng.

## Quy trình thực hiện các công việc chính:

* Xác định các yêu cầu.
* Thiết kế hệ thống.
* Thiết kế giao diện.
* Cài đặt và thử nghiệm.

## Mục tiêu:

* Tra từ vựng.
* Học từ vựng.
* Dịch các đoặn văn bản.
* Cộng đồng từ vựng chia sẻ các từ vựng và kinh nghiệm học tập với nhau.

## Chức năng chính:

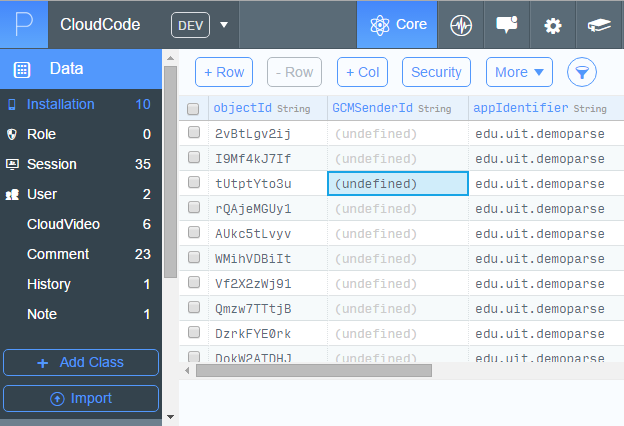
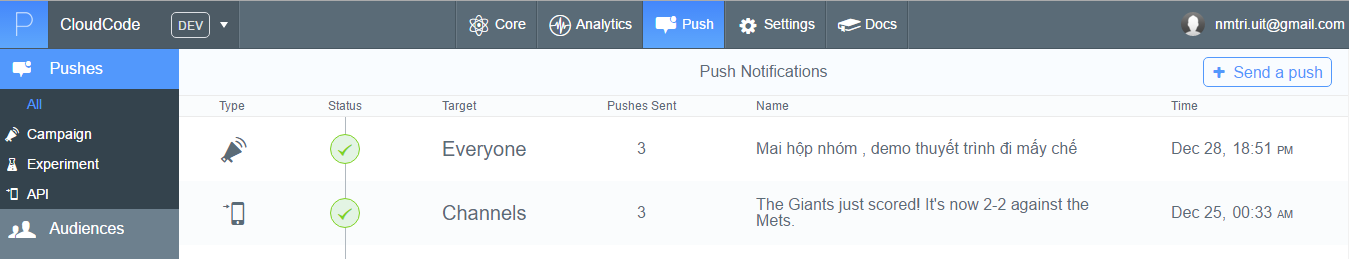
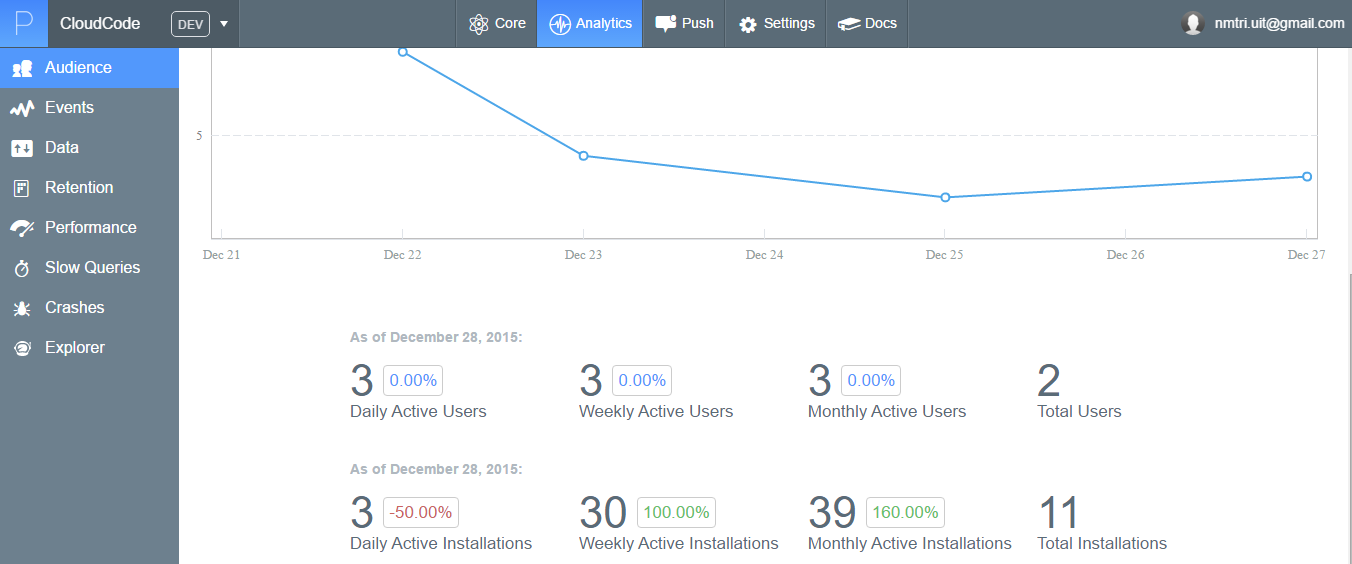
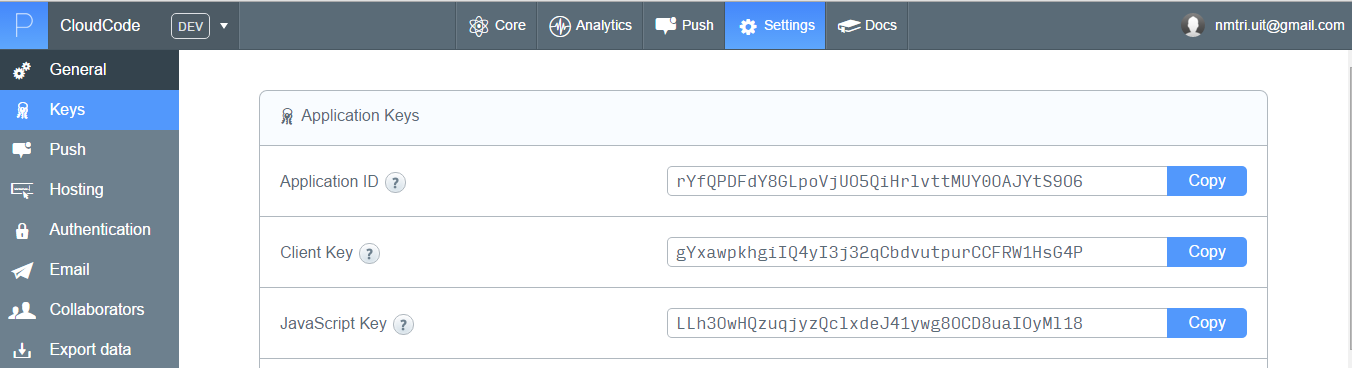
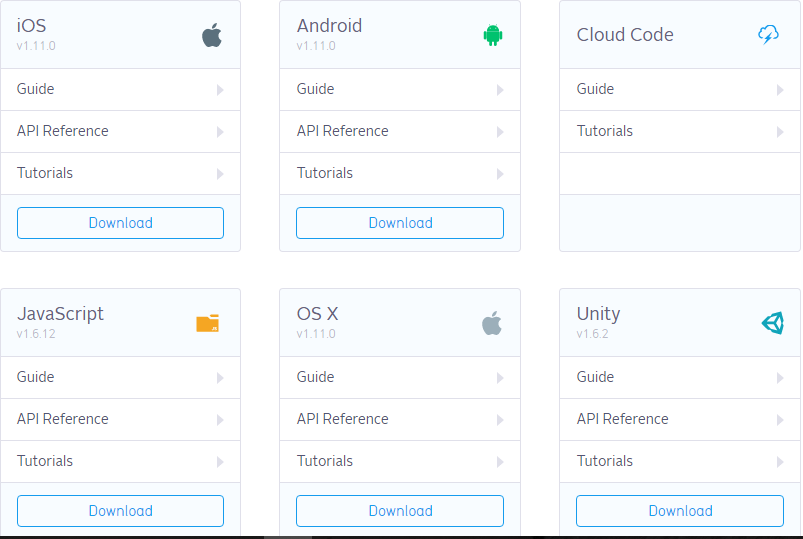
* Tra từ vựng.
* Học từ vựng.
* Nghe đọc từ vựng.
* Dịch các đoặn văn bản.
* Cộng đồng từ vựng chia sẻ các từ vựng và kinh nghiệm học tập với nhau.
* Tra từ vựng nhanh qua clipboard.
* Tra từ theo chuyên ngành.
* Lưu trữ từ vựng online.
* Đồng bộ với máy tính.
* Tạo tài khoản đăng nhập dành riêng cho người sử dụng và đồng bộ dữ liệu lên cloud.
* Đăng nhập bằng facebook

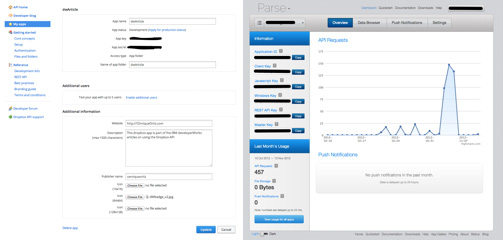
1. **Phân tích đề tài:**
   1. **PARSE:**
      1. **Parse là gì:**

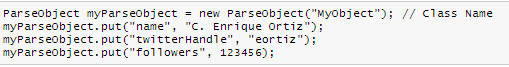
* Parse là một nền tảng đám mây (PaaS & IaaS) cung cấp các API và các dịch vụ đám mây dành cho các ứng dụng iOS, Android và Windows®. Parse SDK còn cung cấp thư viện cho JavaScript và các API REST.
* Khi sử dụng Parse API, Ta có thể chạy các ứng dụng di động của mình trên đám mây một cách nhanh chóng và hao tốn ít tài nguyên.
* Một ứng dụng di động được tích hợp với Parse API có thể dễ dàng lưu trữ các đối tượng và các tập tin dữ liệu trên Parse Cloud, gửi và lắng nghe các tin nhắn push (VD: tin nhắn dạng push notification, cho phép một ứng dụng thông báo cho Ta các tin nhắn hoặc sự kiện mới mà không cần phải mở ứng dụng ra).
* Parse SDK còn cho phép quản lý người dùng, xử lý dữ liệu vị trí địa lý và sử dụng các nền tảng truyền thông mạng xã hội như Twitter và Facebook. Đối với các ứng dụng di động cần mở rộng quy mô, Parse SDK cung cấp tất cả tính co giãn của một nền tảng đám mây.
* Hình 2.1.1 PARSE hỗ trợ hầu hết các nền tảng lập trình và phát triển ứng dụng sau:

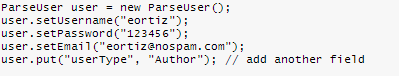


Nguồn: parse.com

* + 1. **Các dịch vụ của Parse:**
* **Core**: Đây là nơi chúng ta quản lý những dữ liệu đã được upload lên thông qua service của Parse: lưu trữ dữ liệu, chạy các tác vụ ngầm theo lịch, quản lý cấu hình ứng dụng với Config, sử dụng Webhooks để tương tác với server người dung.
* Hình 2.1.2 Core
* **Push**: gửi push notification tới các thiết bị đa nền tảng, lọc tin nhắn theo từng đối tượng khách hàng.
* Hình 2.1.3 Push
* **Analytics**: Đây là nơi chúng ta quản lý và thống kê các thông tin về việc sử dụng service trong app: phân tích việc sử dụng dữ liệu của người dùng, phân tích việc hiệu quả của sử dụng các push notification, tìm và fix các lỗi của ứng dụng.
* Hình 2.1.4 Analytics
* **Settings**: Đây là nơi lưu các keys của tất cả API. Ngoài ra, chúng ta có thể chỉnh setting cho ứng dụng của chúng ta tại đây, quản lý bảo mật, xuất dữ liệu,...
* Hình 2.1.5 Setting
* **Docs**: Đây là nơi cung cấp các tài liệu về API để chúng ta tham khảo và sử dụng trong ứng dụng
* Hình 2.1.6 Docs
  + 1. **Bảng điều khiển Parse:**
* Bảng điều khiển Parse hỗ trợ các nhà phát triển trong việc quản lý các ứng dụng. Bảng điều khiển cung cấp các số liệu thống kê về cách sử dụng chung và cách sử dụng ứng dụng-đặc trưng cho các API, các tệp và các tin nhắn push. Các khóa và các giá trị thiết lập ứng dụng được quản lý thông qua bảng điều khiển. Bảng điều khiển này cũng cung cấp một trình duyệt dữ liệu để cho phép các nhà phát triển duyệt và thậm chí chỉnh sửa các đối tượng Parse đã lưu. Trình duyệt dữ liệu rất có ích cho việc gỡ lỗi. Hình 1 là một ảnh chụp màn hình của bảng điều khiển Parse**:**
* Hình 2.1.7 Bảng điều khiển Parse

****

* Các ứng dụng được xác thực thông qua một Application ID (mã định danh ứng dụng) và một Client ID (mã định danh khách hàng). Để có được các ID ứng dụng và ID khách hàng của mình, Ta phải [đăng ký ứng dụng của mình](https://www.parse.com/?#signup) qua bảng điều khiển Parse. Ta sẽ sử dụng các khóa này khi khởi tạo thư viện Parse trên ứng dụng của mình.
  + 1. **Các đối tượng dữ liệu Parse:**
* **ParseObject**: ParseObject biểu diễn một đối tượng dữ liệu trên đám mây Parse. Nó cung cấp các phương thức để thêm các cặp tên-giá trị, kiểm tra xem một khóa cụ thể đã có mặt chưa và xóa hoặc tìm nạp một ParseObject cụ thể của máy chủ. ParseObject cũng cho phép Ta sử dụng các phương thức get…() và put…() khác nhau để xử lý dữ liệu ParseObject, sát nhập các ParseObject, lưu một ParseObject trong máy chủ và nhiều hơn nữa.
* **Parse users**: Một lớp có tên là ParseUser đại diện cho một người dùng và cung cấp chức năng tài khoản-người dùng cho các ứng dụng Parse. Mỗi ứng dụng Parse có những người dùng Parse liên kết với nó. Một lớp ParseUser là một ParseObject nhưng nhưng có các thuộc tính bổ sung là username, password và email. Ta có thể thêm vào bất kỳ giá trị dữ liệu bổ sung nào mà Ta thấy phù hợp.



* **Parse ACL**: Một *ACL* là một danh sách các giấy phép truy cập (hoặc các quyền điều khiển truy cập) gắn liền với một đối tượng dữ liệu. Lớp ParseACL cho phép Ta định nghĩa các giấy phép cho một ParseObject cụ thể. Với các ACL, Ta có thể định nghĩa quyền truy cập công khai vào các đối tượng dữ liệu ứng dụng của mình và Ta có thể hạn chế quyền truy cập cho những người dùng hoặc các nhóm những người dùng (thông qua các vai trò) cụ thể.
  + 1. **Các đối tượng dữ liệu Parse trên đám mây**
* Một khi đã tạo ra và điền vào một ParseObject, ta có thể lưu nó trên Parse cloud. Việc lưu các đối tượng dữ liệu trên Parse cloud thực sự là một trong những điều đơn giản nhất để làm với Parse; Parse hoàn toàn che giấu đi sự phức tạp thường liên quan đến việc biểu diễn dữ liệu, sắp xếp dữ liệu (marshaling), truyền thông mạng và vận chuyển và v.v. Sẽ cần sử dụng các phương thức của trình trợ giúp để ánh xạ cá thể đối tượng dữ liệu của mình vào một ParseObject và ngược lại và ta sẽ cần quyết định xem ta có muốn gửi hoạt động lưu trữ của Parse trên luồng riêng của mình không hoặc có sử dụng lưu trữ theo phương thức nền không.
* ParseObject cung cấp hai loại trong số các phương thức lưu trữ: save() và saveInBackground() (lưu trữ trong nền) là phương thức lưu trữ nên dùng do nó chạy hoạt động lưu trữ trên luồng công việc riêng của nó.
  1. **API:**
     1. **API là gì:**
* API là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng ) là một [giao diện](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)) mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. Chẳng hạn, một chương trình máy tính có thể (và thường là phải) dùng các hàm API của [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) để xin cấp phát bộ nhớ và truy xuất tập tin. Nhiều loại hệ thống và ứng dụng hiện thực API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính.
* Đây là phần mềm hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các lập trình viên có thể rút ra từ đó để tạo nên các tính năng giao tiếp người- máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ. Một trình ứng dụng có thể sử dụng nó để yêu cầu và thi hành các dịch vụ cấp thấp do hệ điều hành của máy tính thực hiện. Hệ giao tiếp lập trình ứng dụng giúp ích rất nhiều cho người sử dụng vì nó cho phép tiết kiệm được nhiều thời gian tìm hiểu các chương trình mới, do đó khích lệ mọi người dùng nhiều ứng dụng hơn.
* Một trong các mục đích chính của một API là cung cấp khả năng truy xuất đến một tập các [hàm](https://vi.wikipedia.org/w/index.php?title=Ch%C6%B0%C6%A1ng_tr%C3%ACnh_c%C3%B2n&action=edit&redlink=1) hay dùng — ví dụ, hàm để vẽ các [cửa sổ](https://vi.wikipedia.org/w/index.php?title=C%E1%BB%ADa_s%E1%BB%95(l%E1%BA%ADp_tr%C3%ACnh)&action=edit&redlink=1) hay các [icon](https://vi.wikipedia.org/w/index.php?title=Icon&action=edit&redlink=1) trên màn hình. Các API, cũng như hầu hết các interfaces, là trừu tượng (abstract). Phần mềm mà muốn cung cấp truy xuất đến chính nó thông qua các API cho sẵn, phải hiện thực API đó. Trong nhiều tình huống, một API thường là một phần của bộ SDK, hay [software development kit](https://vi.wikipedia.org/w/index.php?title=B%E1%BB%99_ph%C3%A1t_tri%E1%BB%83n_ph%E1%BA%A7n_m%E1%BB%81m&action=edit&redlink=1). Một bộ SDK có thể bao gồm một API cũng như các công cụ/phần cứng, vì thế hai thuật ngữ này không thay thế cho nhau được.
* Các hàm API cũng như các hàm bình thường mà chúng ta hay viết và trên từng ngôn ngữ khác nhau cũng có các bộ API tương ứng khác nhau. Ví dụ: Google API có bộ cho .NET, PHP,.. Và mỗi bộ API hầu như đều có hướng dẫn sử dụng rất cụ thể và chi tiết.
* Có nhiều [mô hình thiết kế](https://vi.wikipedia.org/w/index.php?title=M%C3%B4_h%C3%ACnh_thi%E1%BA%BFt_k%E1%BA%BF&action=edit&redlink=1) khác nhau cho các APIs. Interfaces nhằm là cách [thực thi](https://vi.wikipedia.org/w/index.php?title=Th%E1%BB%B1c_thi_(m%C3%A1y_t%C3%ADnh)&action=edit&redlink=1) nhanh nhất thường gồm các tập các [hàm](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_con), [thủ tục](https://vi.wikipedia.org/w/index.php?title=Th%E1%BB%A7_t%E1%BB%A5c&action=edit&redlink=1), [biến](https://vi.wikipedia.org/wiki/Bi%E1%BA%BFn_s%E1%BB%91) và các [cấu trúc dữ liệu](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u). Tuy nhiên, các mô hình khác vẫn tồn tại, như bộ [thông dịch](https://vi.wikipedia.org/w/index.php?title=Th%C3%B4ng_d%E1%BB%8Bch_(ph%E1%BA%A7n_m%E1%BB%81m_m%C3%A1y_t%C3%ADnh)&action=edit&redlink=1) dùng để ước giá biểu thức trong [ECMAScript](https://vi.wikipedia.org/w/index.php?title=ECMAScript&action=edit&redlink=1)/[JavaScript](https://vi.wikipedia.org/wiki/JavaScript).
* Một API tốt thường cung cấp một "hộp đen" hay là một lớp trừu tượng (abstraction layer) bao bọc nó, nhằm đảm bảo là nhà lập trình không thể biết cách hiện thực cụ thể bên trong của mỗi hàm trong API. Điều này làm cho việc thiết kế lại hay cải tiến hàm của API đó trở nên dễ dàng hơn vì nó không làm đổ ỗ các đoạn mã khác mà có sử dụng các hàm đó.
  + 1. **API Bing**
* Bing API Version 2.0 cho phép các nhà phát triển để nhúng một công cụ tìm kiếm linh hoạt và mạnh mẽ như một thành phần tìm kiếm tùy chỉnh trong các trang web của họ và các ứng dụng. Bing API cung cấp mở, tùy chọn linh hoạt cho việc xây dựng và tăng cường trang web hoặc ứng dụng của bạn. Với API người dùng có thể tìm kiếm nội dung web, hình ảnh, những câu chuyện và nhiều hơn nữa.Người dùng cũng có thể kiếm tiền từ các ứng dụng của họ với quảng cáo. Nó sử dụng một giao thức RESTful với kết quả định dạng hoặc trong XML hoặc JSON và cũng cung cấp hỗ trợ cho các giao thức SOAP.
* Bing API cho phép bạn nhúng một công cụ tìm kiếm linh hoạt và mạnh mẽ như một tìm kiếm tùy chỉnh thành phần trong các trang web và các ứng dụng của bạn. Các phiên bản mới bao gồm:
* Thiết bị đầu cuối HTTP có thể cung cấp kết quả trong hoặc XML hoặc JSON định dạng media
* Tăng cường hỗ trợ cho SOAP
* Một đầy đủ [OpenSearch](http://www.opensearch.org/Home) giao diện RSS compliant mà cung cấp truy cập tới nhiều nguồn thông tin của API
* Bing API Version HTTP tạo điều kiện cho quá trình sử dụng HTTP GET để gửi yêu cầu. Trong khi điều này không phải là cơ chế mà bạn đang có khả năng để sử dụng trong một môi trường sản xuất, nó là một cách hữu ích để có được một cảm giác như thế nào API hoạt động. Sử dụng các API từ một trình duyệt có thể bao gồm:
* Bắt một AppID.
* Xác định các giá trị cho các thông số cần thiết.
* Làm việc với các thông số tùy chọn.
* Lựa chọn một giao thức.
* Gửi yêu cầu.
* Làm việc với kết quả.
* Yêu cầu mẫu khác.
* Kết quả phân tích cú pháp JSON.
* Các chức năng của API Bing:

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả** |
| **Web** | Tìm kiếm các nội dung web |
| **Image** | Tìm kiếm ảnh trên web |
| **News** | Tìm kiếm tin tức |
| **Spell** | Tìm kiếm từ điển Encarta cho gợi ý chính tả |
| **Translate** | Tiềm kiếm và dịch văn bản |
| **Maps** | Thông tin vị trí kết hợp với bản đồ và tọa độ |
| **Knowledge** | mang lại các kiến ​​thức chuyên môn của Bing vào ứng dụng |
| **Phonebook** | Tìm kiếm danh bạ |
| **RelatedSearch** | Trả về chuỗi truy vấn giống của bạn |
| **Video** | Tìm kiếm các video trên web |

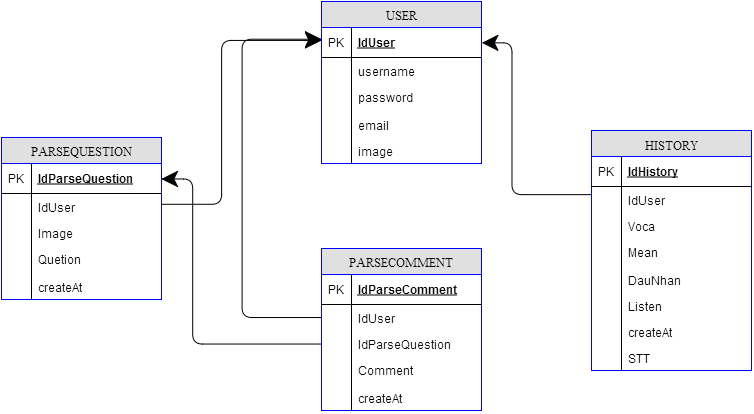
* **Microsoft Translate**
* Microsoft Translator API là một đám mây dựa trên [bản dịch tự động](https://www.microsoft.com/en-us/translator/at.aspx) dịch vụ (aka máy dịch) hỗ trợ nhiều ngôn ngữ cao đến hơn 95% tổng sản phẩm quốc nội của thế giới (GDP). Dịch có thể được sử dụng để xây dựng các ứng dụng, các trang web, các công cụ, hoặc bất kỳ giải pháp đòi hỏi hỗ trợ đa ngôn ngữ.
* Được xây dựng cho doanh nghiệp, Microsoft Translator là một chứng minh, tùy chỉnh, và giải pháp mở rộng cho dịch tự động. Microsoft công nghệ dịch tính năng quyền hạn dịch trên sản phẩm của Microsoft, bao gồm Office, SharePoint, Lync, Yammer, [Visual](https://www.microsoft.com/en-us/translator/mat.aspx)Studio, [Bing, và](https://www.bing.com/translator)Skype. Đơn giản chỉ cần tích hợp dịch sang web, máy tính để bàn, hoặc các ứng dụng điện thoại di động, Translator API cung cấp một chức năng phong phú này cho bất kỳ nhà phát triển, không phải đề cập đến một sự lựa chọn rộng của giao diện:. AJAX, HTTP, SOAP, OData, và Translator Web Widget [Tìm hiểu thêm về dịch tự động và làm thế nào Microsoft Translator làm việc.](https://www.microsoft.com/en-us/translator/at.aspx)
* Microsoft Translator có thể được tùy chỉnh với cả prepublishing, [Microsoft Hub](https://www.microsoft.com/en-us/translator/hub.aspx) Translate, và postpublishing, với các [dịch Khung Collaborative](https://www.microsoft.com/en-us/translator/ctf.aspx)(CTF).
* Microsoft Hub Translate cho phép khách hàng để xây dựng một hệ thống dịch điều chỉnh các thuật ngữ và phong cách của họ. CTF cho phép khách hàng sử dụng một mô hình CrowdSourced kiểm soát hoặc mở để cải thiện bản dịch sau khi xuất bản.
* **Image**
* Trả về một danh sách các hình ảnh có liên quan đến thuật ngữ truy vấn. Kết quả hình ảnh chứa các thuộc tính về các phương tiện truyền thông tập tin hình ảnh như chiều rộng, chiều cao và kích thước tập tin, cùng với các URL cho các trang có chứa hình ảnh.
  + 1. **API FaceBook**
* Facebook hiện đang chiếm lượng người dùng đông đảo nhất trên thế giới. Với 1 tỷ người dùng, bạn có thể có rất nhiều lợi thế khi sử dụng cho social login (dùng Stormpath), điều này giúp việc chia sẻ và kiếm tiền nhanh hơn.
* Facebook hiện đang hạn chế loại data được xuất hiện trên ứng dụng, nhưng vẫn còn rất nhiều đất cho các nhà lâp trình phát triển tiếp. Hơn nữa, Facebook cũng có hệ thống định dạng ngôn ngữ riêng, mobile backed, Parse và nhiều thứ khác nữa đang chờ bạn tìm hiểu.
* Những ứng dụng đơn giản như: đăng nhập qua tài khoản Facebook, tích hợp Facebook like ,share , comment vào website,tạo quảng cáo… đều được hầu hết các website áp dụng. Tạo ứng dụng trên nền canvas facebook. Allows anyone to "build social applications on Facebook and the Web."
* Thành phần của Facebook API
  + **Graph API** : dựa vào URl để truy xuất vào tài nguyên trong facebook.
  + **Authentication** :chứng thực người dung thong qua OAuth 2.0(là 1 chuẩn ) Ví dụ login bằng facebook.
  + **Social Plugins** : cách đơn giản nhất( copy ,dán) để tích hợp api facebook vào trang web của mình. Ví dụ : nút like …
  + **Open Graph Protoco**l: This is an open standard Facebook set up to identify pages and things on the Web. ( kéo tag meta vào xài) tham khảo trên trang <http://opengraphprotocol.org/>
  + **FQL** (Facebook Query Language) : giống sql dung khi mà Graph API bó tay khi lấy dữ liệu.
  + **Dialogs** : facebook cung cấp hộp thoại đăng nhập, post bài vào timeline ,..
  + **Ads API** : Cung cấp api để xây dựng app quảng cáo.
  + **Chat** : bạn có thể tích hợp facebook chat vào trang web của bạn hoặc desktop , ….
  + **Localization and translation**
  + **PublicFeedAPI**
  + **KeywordInsights API**
  1. **Xử lý tập tin trong Android**
     1. **Internal Storage**
* Có 2 trường hợp: Thứ nhất: đọc Resource File (Read only), Thứ nhì : đọc và ghi tập tin (Read & Write).
  + **Resource File** (Read only): tạo 1 tập tin và nhập đại vào một số dữ liệu rồi kéo thả tập tin đó vào thư mục **drawable**. Nếu như ứng dụng của bạn không có thư mục này thì bạn tự tạo nó nhớ là viết y xì “**drawable**“, tên file cũng đừng để khoảng trắng, đừng để chữ số đằng trước. Sử dụng InputStream, InputStreamReader, BufferedReader, StringBuilder.
  + **Read & Write:** phần Read giống như Resource File. Write sử dụng thêm các kiễu dữ liệu: FileOutputStream, OutputStreamWriter.
    1. **External Storage**
* Là tương tác với tập tin được lữu trữ trên SD Card và bộ nhớ trong Để thao tác được trên SD Card, bạn phải cấp quyền cho ứng dụng (**uses – permission**)
* Để vừa ghi vừa đọc trên **SD Card** thì ta phải sử dụng 2 permission bên dưới trong**AndroidManifest**:

<**uses-permission** android:name=”android.permission.WRITE\_EXTERNAL\_STORAGE” />  
<**uses-permission** android:name=”android.permission.READ\_EXTERNAL\_STORAGE” />

* Để lấy được đường dẫn của SD Card dùng lệnh:
* String sdcard = **Environment**.**getExternalStorageDirectory**().**getAbsolutePath**();
* Sau đó mọi thứ tương tác với tập tin thì y xì như Java thuần túy, ở đây Tôi dùng **Scanner** để đọc và **OutputStreamWriter** để ghi.
  + 1. **Saving Cache Files**
* Lợi ích của cache files là: Để tăng tốc độ xử lý của ứng dụng khi thường xuyên truy cập internet thì ta nên lưu cache.
* Để lấy được đường dẫn của thư mục Cache trong ứng dụng, ta dùng lệnh : **getCacheDir**() .

# Thiết kế

* 1. **Cơ sở dữ liệu**



* 1. **Kiến trúc chương trình**
     1. **Activity\_Login:**
* Chức năng: Là màn hình đăng nhập chính gồm có đăng nhập bằng tài khoảng Facebook và tài khoản bình thường
* Các phương thức chính :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | saveUser | void | Bitmap | -Khởi tạo user lưu trên parse , nhận hình ảnh từ đại diện từ Facebook |
| 2. | getUserdDetailFromFB | void | null | -Nhận dữ liệu từ facebook ( email, tên người đăng nhập) |
| 3. | getUserDetailFromParse | void | null | -Nhận dữ liệu người dùng từ parse, sử dụng khi đã đăng khởi động lại app |

* + 1. **Activity\_Login\_Normal**
* Chức năng: là màn hình đăng nhập phụ với tài khoảng và mật khẩu.
  + 1. **Activity\_Resgister**
* Chức năng: là màn hình cho người dung đăng kí tài khoản để sử dụng app
* Các phương thức chính:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | @Override  onActivityResult | void | Int,int,Intent | -Nhận dữ liệu file ảnh được chọn làm ảnh đại diện |

* + 1. **DictPlusApp**
* Chức năng: Chức năng chính của lớp này là khởi tạo 1 cloud database và đăng kí database cho ứng dụng trên parse.com, và Facebook
  + 1. **FragmentDrawer**
* Chức năng: là cầu nối với MainActivy để xử lý các sự kiện của Drawer
  + 1. **FragmentStudy**
* Chức năng: chứa các button để truy cập các dịch vụ học từ vựng, tra từ nhanh và tuy chọn cài đạt cho các dịch vụ
  + 1. **MainActivity**
* Chức năng : nơi đây sẽ xử lý các sự kiện chuyển fragment ( Tra từ, lịch sử, hoc từ vựng), chuyển đến activity mới.
  + 1. **SpeechEnglish**
* Chức năng : phục vụ cho việc phát âm tiếng anh (Text to speech ) đươc dùng cho trong từ điển Anh-Việt.
  + 1. **TabAdapter:**
* Chức Năng : là cầu nối với ViewPager để khởi tạo các Tab trong ViewPager.
  + 1. **Fragment\_TraTu:**
* Chức Năng : nơi khởi tạo các Tab (Anh-Việt, Việt – Anh, Chuyên Ngành)
  + 1. **Tab\_AnhViet**
* Chức năng: khởi tạo dữ liệu từ điển offline, ghi dữ liệu lịch sử tra từ, hiện dữ liệu Anh-Việt lên Tab Anh-Việt.
* Các phương thức chính
  + 1. **Tab\_VietAnh**
* Chức năng: Khởi tạo dữ liệu cho phần từ điển Việt-Anh
  + 1. **Tab\_Online:**
* Chức năng: Tra từ theo chuyên nghành, dịch câu, văn bản
* Các phương thức chính:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | readData | void | null | -Đọc dữ liệu từ file txt gồm (anhviet.txt và vietanh.txt) |
| 2. | createHistory | Static void | String , String | -Nhận từ vựng và nghĩa để ghi vào lịch sử tra từ |
| 3. | getShortMean | String | String | -Rút ngọn nghĩa của từ trước khi ghi vào lịch sử |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | translate | String | String | -Sử dụng API của Bing để dịch câu Tiếng Anh |
| 2. | @Override  onEditorAction | boolean | TextView,int,KeyEvent | -Nhận file HTML nghĩa của từ, từ CloudCode hiện lên WebView |
| 3. | SavePase | void | String,String | -Chuyển file HTML thành text lưu vào lịch sử Parse |

* + 1. **FragmentAllHistory**
* Chức năng : lấy toàn bộ dữ liệu lịch sử của các user đã được tạo về hiện lên listview trong Tab Cộng Đồng giúp cho mọi người có thể chia sẽ các từ vựng với nhau , giúp ích cho việc học từ
* Các phương thức chính :
  + 1. **FragmentHistory**
* Chức năng: Lưu lại lịch sử tra từ dưới dạng offline, khi có mạng thì sẽ tự đồng bộ lên mây, khác với FragmentAllHistory là không có mạng vẫn sử dụng được
* Các phương thức chính : tương tự FragmentAllHistory có điểm khác là có thể edit được lịch sự và xóa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | fetchMovies() | void | null | -Dùng để load lại dữ liệu từ parse, khi kéo listview xuống |
| 2. | getDataTracNghiem | Static  List<DataTracNghiem> | null | -Lấy danh sách các từ được đánh dấu check trong lịch sử sau đó ramdom nghĩa với nhau tạo thanh dữ liệu để phục vụ trong phần Học từ ( Làm trắc nghiệm) |
| 3. | animateRemvoal | void | ListView,View | -Tạo hiệu ứng swipe listview dùng để xóa từ trong lịch sử |

* + 1. **History**
* Chức năng : Đây là 1 lớp để lưu thông tin dữ liệu của mỗi từ khi được đưa lên mây gôm: Từ Vựng, Nghĩa, STT, Trạng Thái, File Nghe Mp3
* Các Phương Thức Chính
  + 1. **HistoryAdapter**
* Chức năng: Là cầu nối với ListView trong FragmentHistory va FragmentAllHistory dùng để custom lại ListView, giúp thực hiện việc đỗ dữ liệu từ Parse lên ListView
* Các phương thức chính

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | put | void | String,Object | -Ghi dữ liệu lên đám mây theo cột với tham sô tên cột được truyền vào, và dữ liệu được đưa lên Parse. |
| 2. | getString | String | String | -Nhận dữ liệu từ đám mây Parse, thông qua tên cột được truyền vào. |
| 3. | getBoolean | boolean | String | -Nhận dữ liệu từ đám mây Parse, thông qua tên cột được truyền vào |
| 4. | getParseFile | Byte[] | String | -Nhận dữ liệu từ đám mây Parse, thông qua tên cột được truyền vào |

* + 1. **PopupEditHistory**
* Chức năng: Là cửa sổ thu nhỏ trên xuất trên FragmentHistory khi chọn từ trong ListView, dùng để thay đổi dữ liệu các từ, nghĩa trong lịch sử
  + 1. **TabPublic:**
* Chức năng: nơi khởi tạo các Tab (Lịch Sử, Cộng Đồng)
  + 1. **DataAdapter**
* Chức Năng: Là cầu nối với ListView trong Tab\_AnhViet và Tab\_VietAnh , giúp thực hiện đỗ dữ liệu vào ListView.
  + 1. **DataOffline**
* Chức năng : Đây là lớp để lưu trữ dữ liệu offline phục vụ cho tra từ, bao gồm: Từ Vựng và Nghĩa. Dự liệu Offline được tao trong hàm readData trong lớp Tab\_AnhViet
  + 1. **Popup\_TraTu**
* Chức năng : Là cửa sổ thu nhỏ , khi click vào các từ vựng trong ListView trong Tab\_AnhViet,Tab\_VietAnh. Giúp hiện đầy đủ nghĩa của từ đó.
  + 1. **Constants**
* Chức năng: lưu trữ các biến static dùng để xác định các service được tạo ra đang hoạt động hay đã dừng.
  + 1. **DataTracNghiem**
* Chức Năng : Đây là lớp để lưu trữ dữ liệu dùng trong TracNghiemTuVung , bao gồm: Từ Vựng, Câu A,B,C,D, Đáp Án.
  + 1. **DauNhan**
* Chức năng: Tìm các phiên âm thích hợp với từ vựng được ghi vào lịch sử
  + 1. **DichNhanh:**
* Chức năng: Đây là 1 lớp service hỗ trợ tra từ nhanh thông qua clipboard.
* Các Phương Thức Chính: void handleStart(){}: khởi tạo cửa sổ WindowManager xuất hiện trên màn hình điện thoại ( Giống như Messenger của Facebook khi có tin nhắn đến sẽ có 1 cửa sổ nhỏ trên màn hình), và kích hoạt timer dùng để kiểm tra thay đổi của ClipBoard 100ms 1 lần. Nếu có thay đổi thì sẽ lấy đoạn text trong Clipboad dịch nghĩa và hiện lên WindowManager
  + 1. **FloatingWindow\_NhacNho**
* Chức năng: Đây là 1 lớp service hỗ trợ nhắc nhỡ từ vừng hiện lên màn hình thông qua WindowManager. Hỗ trợ text to speech tiếng anh, và tiếng việt mỗi khi từ nhắc nhở thay đổi.
* Các phương thức chính

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | playMp3 | void | Byte[] | -Phát âm từ vựng với dữ liệu âm thanh lưu trên mây là byte[] |
| 2. | speakVNOF | void | String | Phát âm tiếng việt tương ứng với tham số nghĩa của từ truyền vào. Sử dụng hỗ trợ cua công cụ vnSpeakTTS-HongVi |
| 3. | speakENOF | void | String | -Phát âm tiếng anh với tham số là từ vựng đang hiện nhắc nhở, sử dụng công cụ TextToSpeech mặc định của máy (Google) |
| 4. | @Override  onUtteranceCompleted | void | String | Phương thức này được gọi khi TextToSpeech phát âm kết thúc, dùng để chuyển qua từ mới |
| 5. | autoNextVoca | Void | null | -Chuyển qua từ nhắc nhở kế tiếp |
| 6. | handleStart | Void | Null | -Khởi tạo cửa sổ nhắc nhở trên màn hình |
| 7. | NofitionMute | Void | Null | -Được sử dụng khi nhắc nhở không cần phát âm từ, ở đây tính toán thời gian chuyển sang từ tiếp theo và update thanh thông báo |
| 8. | DelayAnhViet,DelayEnglish | Void | Null | -Tính toán thời gian để chuyển sang thông báo mới. |
| 9. | updateNotification | Void | String,String | -Dùng để update thanh thông báo khi chuyển sang từ mới. |
| 10. | showNotification | Void | String,String | -Hiện thông báo lên. |

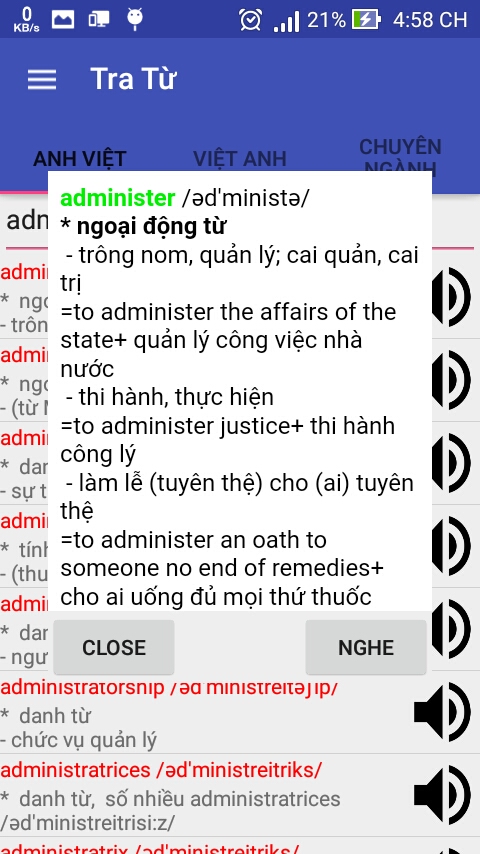
* + 1. **FragmentTracNghiem**
* Chức năng: Hiện thị dữ liệu trắc nghiệm đã được tạo lên Activity TracNghiem giúp người dùng làm bài tập trắc nghiệm để nhớ từ vựng hơn.
  + 1. **TestSpeechVoca**
* Chức năng : Giúp người sử dụng kiểm tra phát âm của từ vựng với sự hỗ trợ của công cụ Google Speech Recognition.
  + 1. **ActivityQuestion**
* Chức năng: Là nơi cho người dùng để đăt câu hỏi lên diễn đàn hỏi đáp
* Các phương thức chính :
* @Override
  + Public void onActivityResult(int requesCode,int resultCode,Intent data){}
  + Chức năng: nhận dữ liệu trả về của Activity tìm ảnh trong bộ nhớ và tìm ảnh trên Internet.
    1. **Activity\_Comment**
* Chức năng : nơi các người dùng giải đáp câu hỏi mà bạn đã đặt trên diễn đàn theo thời gian thực với sự hỗ trợ của ParsePush và Google Message
* Các phương thức chính:
  + public void processReceive(Context context,Intent inten){}
  + Chức năng: nhận dữ liệu từ các comment mà các người dùng khác đang tương tác vào câu hỏi mà bạn truy cập thông qua server ParsePush gửi về.
    1. **AdapterComment**
* Chức năng: là cầu nối với ListView ở Activity\_Comment để đỗ dự liệu comment vào hiện thị 1 dãy comment lên.
  + 1. **ParseComment**
* Chức năng: Đây là 1 lớp để lưu thông tin dữ liệu của mỗi comment khi được đưa lên mây gôm: Chuỗi Comment., và nhận dữ liệu Comment từ sever gửi về.
  + 1. **Activity\_ListQuestion**
* Chức năng: Hiện thị danh sách các câu hỏi mà người dùng đã đăng lên
  + 1. **ActivitySearchImg**
* Chức năng: Tìm ảnh trên Bing với từ khóa với sự hỗ trợ của API Bing
* Các phương thức chính:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên phương thức chính | Kiểu dữ liệu trả về | Tham số truyền vào | Chức năng |
| 1. | doSeach() | void | null | -Gọi đến lớp AsyncTask jsonTask nới đây sẽ kết nối đến server search ảnh của Bing , sau đó trả về dữ liệu json chứa thông tin của ảnh cần tìm |
| 2. | getDataTracNghiem | Static  List<DataTracNghiem> | null | -Lấy danh sách các từ được đánh dấu check trong lịch sử sau đó ramdom nghĩa với nhau tạo thanh dữ liệu để phục vụ trong phần Học từ ( Làm trắc nghiệm) |
| 3. | Class  getThumbTask | null | URL | -Lớp AsyncTask này sẽ nhận địa chỉ URL từ file json, sau đó tải ảnh về đổ vào GirdView |

* + 1. **ThumbnailAdapter:**
* Chức Năng: Là cầu nối với GirdView ở ActivitySearcgImg đê đổ dữ liệu ảnh vào và hiện thị lên.
  1. **Giao diện**
     1. **Danh sách các màn hình**

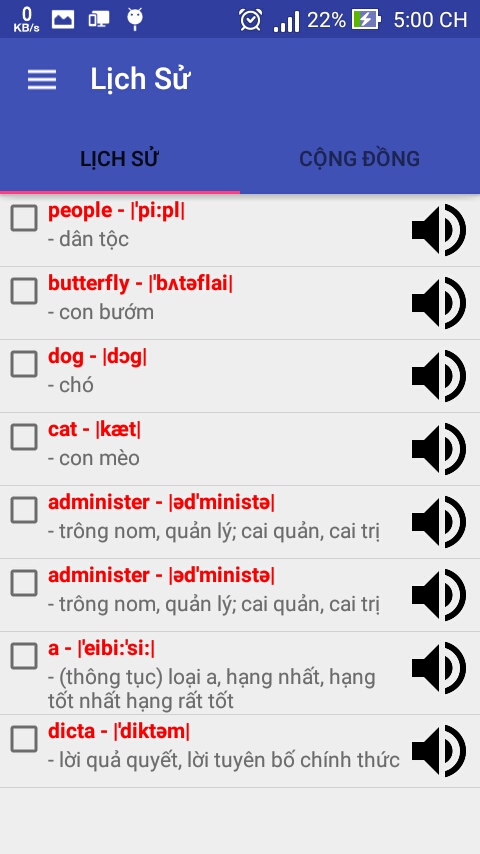
|  |  |  |
| --- | --- | --- |
| STT | Màn hình | Chức năng |
| 1 | Tra từ | Người dùng tra từ vựng theo từ điểm Anh-Việt, Việt-Anh và Chuyên Ngành. |
| 2 | Lịch sử | Ghi chép lại các từ vựng đã tra của người dùng và cộng đồng. |
| 3 | Học từ vựng | Người dùng chọn dữ liệu từ lịch sử hoặc cộng đồng. Sau đó chọn các chức năng tra nhanh, làm trắc nghiệm, nhắc nhở và kiểm tra phát âm. |
| 4 | Trắc Nghiêm | Làm trắc nghiêm về từ vựng |
| 5 | Kiểm Tra Phát Âm | Người dùng phát âm từ hiện trên màn hình |
| 4 | Hỏi đáp | Đặt câu hỏi để cộng đồng giải đáp, có thể miêu ta câu hỏi bằng hình ảnh trên máy hoặc Internet sau đó đăng lên New Feed |
| 5 | Tìm ảnh trên Ping | Gõ từ khóa để tìm hình ảnh |
| 6 | Danh sách các câu hỏi | Xem các câu hỏi và câu trả lời của cộng đồng. |
| 7 | Đăng nhập | Người đăng nhập qua FaceBook hoặc tài khoản đã tạo |
| 8 | Đăng ký | Người dùng tạo tài khoản |
| 9 | Chờ | Người dùng chờ nạp đữ liệu vào app |
| 7 | Menu | Người dùng có thể cập nhật Avatar và trỏ tới các màn hình Tra Từ, Lịch Sử, Học Từ Vựng, Hỏi Đáp, Đăng Xuất. |

* + 1. **Màn hình Tra Từ**



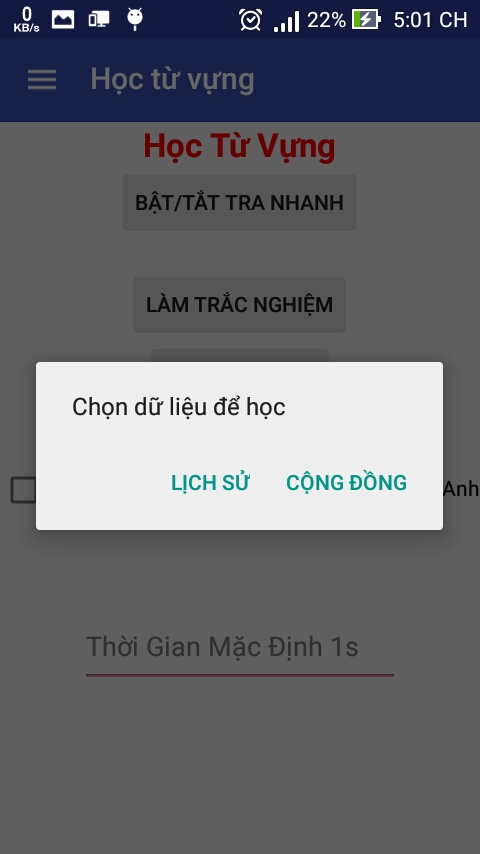
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | listView | ListView | Các từ liên quan với từ cần tra |
| 2 | edtVocaOnline | EditText | Gõ nội dung tra |
| 3 | rbToanTin | RadioButton | Chọn chuyên ngành toán tin |
| 4 | rbThongdung | RadioButton | Chọn thông dụng |
| 5 | rbKinhte | RadioButton | Chọn chuyên ngành kinh tế |
| 6 | wvResult | WebView | Màn hình kết quả |

* + 1. **Màn hình Lịch Sử**



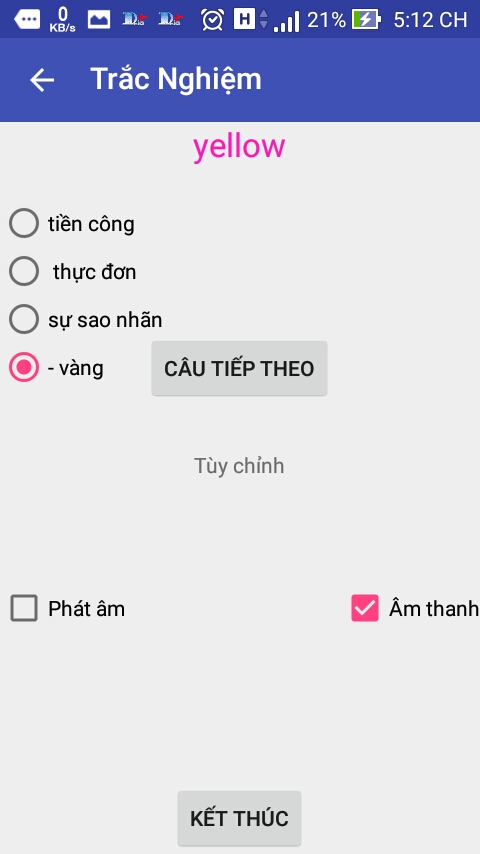
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | listViewHistory | ListView | Chứa cá từ đã tra |

* + 1. **Màn hình Học Từ Vựng**



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | btnStartTracNghiem | Button | Các từ liên quan với từ cần tra |
| 2 | textView | TextView | Gõ nội dung tra |
| 3 | btnCheckPhatAm | Button | Chọn chuyên ngành toán tin |
| 4 | btnNotifition | Button | Chọn thông dụng |
| 5 | cbsdTV | CheckBox | Chọn chuyên ngành kinh tế |
| 6 | cbsdTA | CheckBox | Màn hình kết quả |
| 7 | btnCheckPhatAm | Button | Kiễm tra phát âm |
| 8 | btnsdTranhanh | Button | Tra nhanh |

* + 1. **Màn hình Trắc Nghiệm**



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | tvStudyVoca | TextView | Hiện từ vựng đang học |
| 2 | btnStudyOk | Button | Chọn câu tiếp theo |
| 3 | rbCauA | RadioButton | Câu A |
| 4 | rbCauB | RadioButton | Câu B |
| 5 | rbCauC | RadioButton | Câu C |
| 6 | rbCauD | RadioButton | Câu D |
| 7 | button3 | Button | Tùy chình |
| 8 | cbtnPhatam | CheckBox | Chọn để phát âm |
| 9 | btnTNend | Button | Kết thúc học |

* + 1. **Màn hình Kiểm Tra Phát Âm**



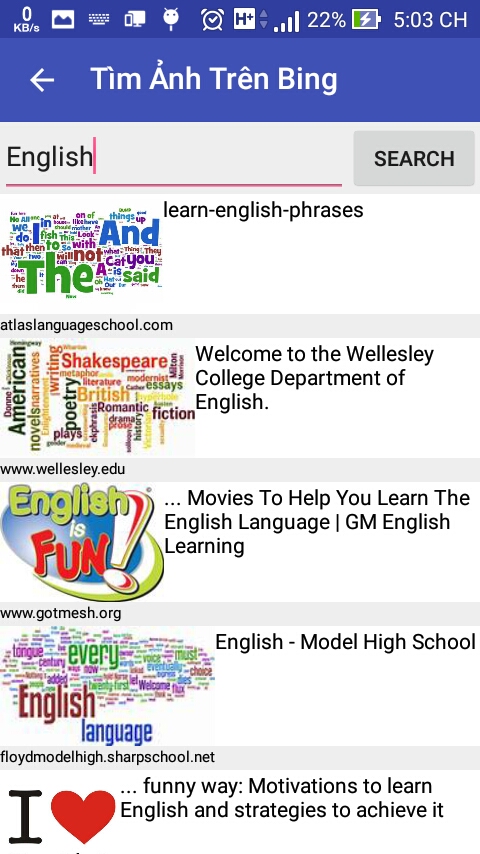
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | tvStudyVoca | TextView | Hiện từ vựng đang học |
| 2 | btnKiemTra | Button | Kiểm tra phát âm |

* + 1. **Màn hình Hỏi Đáp**



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | \_frEdCauHoi | EditText | Gõ câu hỏi cần hỏi |
| 2 | \_frImage | ImageView | Hình cần hỏi |
| 3 | \_frbtnTimThe | Button | Chọn ảnh từ thẻ nhớ |
| 4 | \_frbtnTimIn | Button | Chọn ảnh từ Internet |
| 5 | \_frbtnNew | Button | Đăng lên newfeed |
| 6 | btnGoQues | Button | Câu hỏi của cộng đồng |

* + 1. **Màn hình Tìm Ảnh Trên Ping**



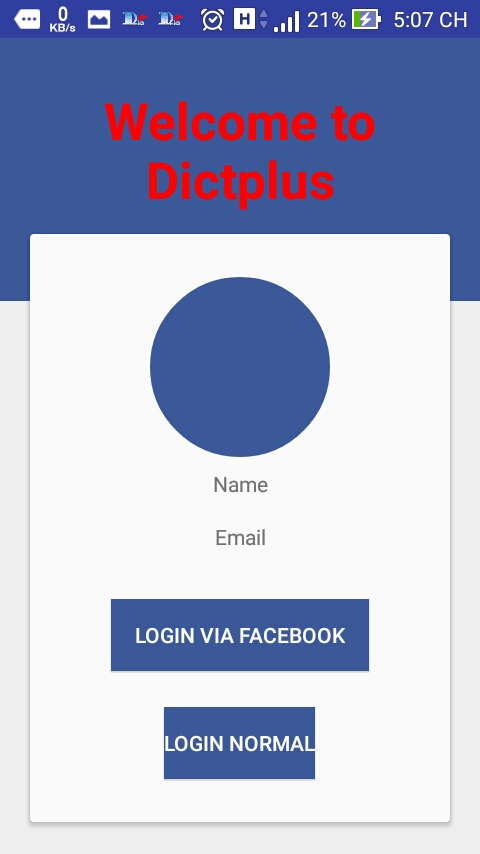
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | editText | EditText | Gõ nội dung cần tìm |
| 2 | lvImage | ListView | Hình cần tìm |
| 3 | btnSearch | Button | Tìm hình ảnh |

* + 1. **Màn hình Danh Sách Các Câu Hỏi**



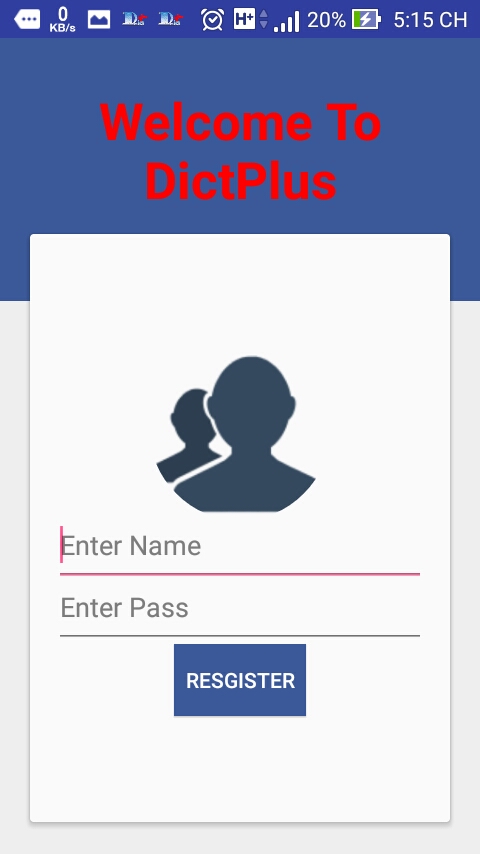
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | lvImage | ListView | Danh sách các câu hỏi comment và hình ảnh |

* + 1. **Màn hình Đăng Nhập**



|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | txt\_name | TextView | Tên người dùng |
| 2 | txt\_email | TextView | Email người dùng |
| 3 | buttonLognor | Button | Login bình thường |
| 4 | btn\_fb\_login | Button | Login bằng FaceBook |

* + 1. **Màn hình Đăng Ký**



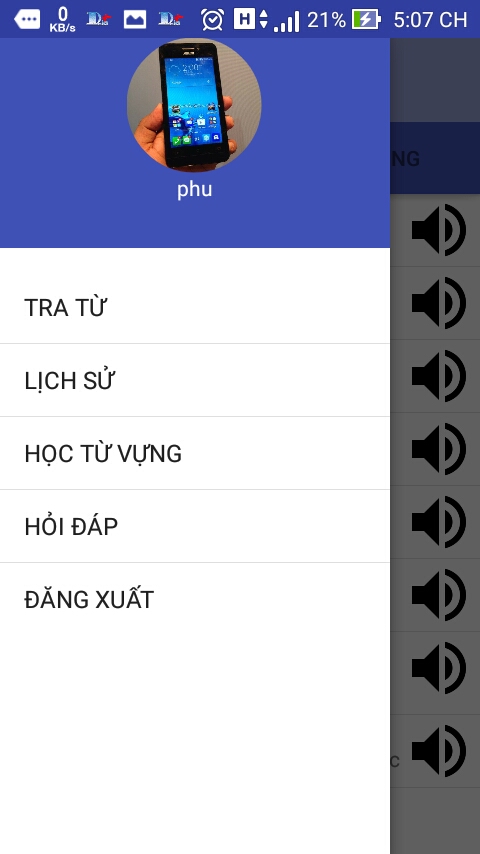
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu | Chức năng |
| 1 | editText2 | TextView | Gõ User name |
| 2 | editText3 | TextView | Gõ Password |
| 3 | btn\_fb\_dangki | Button | Đăng ký |
| 4 | btn\_fb\_login | Button | Login |

* + 1. **Màn hình Chờ**



Sử dụng AyncTask

* + 1. **Màn hình Menu**



# Cài đặt và thử nghiệm:

Cần cài android 4.0 trở lên.

# Hướng phát triển:

* Bộ gõ tiếng việt là một đề tài hay mang nhiều tính ứng dụng trong thực tế. Ứng dụng hiện tại chỉ hỗ trợ những chức năng cơ bản cho người dùng.
* Tuy nhiên, có thể phát triển thêm những chức năng mới: tự nhận diện lỗi chính tả, gõ chữ không dấu đoán chữ có dấu, thêm một số bảng mã và kiểu gõ khác.
* Tự động gợi ý từ đã gõ, lấy data của từ điển để gợi ý từ và nhận diện lỗi chính tả tôt hơn
* Nhận dang giọng nói để người dùng có thể nhập văn bảng bằng giọng nói một cách nhanh chóng

1. **Đánh giá - Kết luận:**

* Lời đầu tiên, nhóm chúng em xin chân thành cám ơn thầy cô đã cung cấp cho chúng em những kiến thức hết sức bổ ích về môn học này, cũng như những kỹ năng cần thiết trong lập trình qua những đề tài mang tính thực tế cao và gần gũi với đời sống. Qua đó, chúng em đã tích lũy nhiều kỹ thuật cũng như là kinh nghiệm quý báu.
* Sau quá trình cài đặt và phát triển ứng dụng PHKey đã hoàn thành. Tuy còn nhiều thiếu sót nhưng cũng đã cơ bản hoàn thiện được những chức năng cần có, mong thầy cô có thể góp ý để giúp chúng em phát triển hoàn thiện ứng dụng.

# Mục tiêu - Giới hạn:

* 1. **Mục tiêu:**
* Phần mềm giúp nguời sử dụng có thể gõ tiếng việt một cách nhanh chóng trong môi trường Windows.
* Phần mềm sử dụng bảng mã unicode thông dụng và những cách gõ thông dụng hiện nay là Telex, Vni.
* Cài đặt thêm một số chức năng gõ tắt, bỏ dấu, bật hộp thoại khi khởi động, khởi động cùng Windows,…
  1. **Giới hạn:**
* Phần mềm chưa thể tự định nghĩa kiểu gõ mà người dùng mong muốn.
* Phần mềm vẫn còn đơn giản chưa có nhiều chức năng quá mới so với các phần mềm gõ tiếng việt hiện tại.

# Tài liệu tham khảo:

* EbookLamBoGoTiengViet của tác giả Bùi đức Tiến.
* Thư viện Hook của <http://www.codeproject.com/>.
* <http://www.stackoverflow.com>
* <https://www.wikipedia.org>
* <http://diendan.congdongcviet.com/>

**NHẬN XÉT**

**(Của giảng viên)**