ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

———————*———————

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

XÂY DỰNG HỆ THỐNG TƯ VẤN TÀI LIỆU HỌC TIẾNG ANH TRÊN ANDROID

Sinh viên thực hiện: **Hoàng Đức Việt**
Lớp AS - K57

Giáo viên hướng dẫn: TS.
Phạm Văn Hải

HÀ NỘI 05-2017

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Hoàng Đức Việt

Điện thoại liên lạc: 0902239958 Email: cooro1994@gmail.com

Lớp: Việt Nhật AS Hệ đào tạo: Chính quy

Đồ án tốt nghiệp được thực hiện tại: Đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 01/03/2017 đến 30/05/2017

2. Mục đích nội dung của ĐATN

- Nghiên cứu tìm hiểu về hệ thống tư vấn và các kỹ thuật cơ bản.
- Nghiên cứu các đặc tính của Kansei ảnh hưởng đến tâm lý và hành vi lựa chọn của con người.
- Đề xuất xây dựng một hệ thống tư vấn tài liệu học tiếng Anh E-learning dành cho người dùng điện thoại Android.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu về hệ thống tư vấn, hệ thống gợi ý và các kỹ thuật cơ bản.
- Tìm hiểu về giải thuật Context-matching và ứng dụng trong việc tư vấn dựa trên nguyện vọng và năng lực của người dùng.
- Tìm hiểu về Kansei Engineering và ứng dụng của nó trong việc cá nhân hoá kết quả tư vấn.
- Phân tích, thiết kế và cài đặt hệ thống tư vấn tài liệu học tiếng Anh E-learning dành cho người dùng điện thoại Android.

4. Lời cam đoan của sinh viên:
Tôi - Hoàng Đức Việt - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của TS. Phạm Văn Hải.
Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 31 tháng 05 năm 2017
Tác giả ĐATN

Hoàng Đức Việt

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ

Hà Nội, ngày 31 tháng 05 năm 2017
Giáo viên hướng dẫn

TS. Phạm Văn Hải

LỜI CÁM ƠN

Để có thể hoàn thiện được đồ án, tôi đã nhận được sự giúp đỡ của rất nhiều người.

Trước hết, em xin gửi lời cảm ơn tới thầy giáo, TS. **Phạm Văn Hải** đã tận tình hướng dẫn, chỉ bảo và giúp đỡ trong suốt thời gian qua. Sự nhiệt huyết và những kiến thức mà thầy truyền đạt là vô giá đối với em, đã giúp em vượt qua những khó khăn trong quá trình nghiên cứu và hoàn thiện đồ án. Một lần nữa, em xin chân thành cảm ơn thầy.

Tiếp đó, em xin cảm ơn tập thể các thành viên team **sPhoton**, mọi người đã luôn ở bên cạnh em chỉ bảo tận tình dù em còn mắc nhiều sai sót. Đặc biệt, em xin gửi lời cảm ơn chân thành nhất tới các anh **Nguyễn Phi Hiệp** đã cho em cơ hội được tiếp xúc công việc trong môi trường thực tế, đồng thời học hỏi được nhiều điều bổ ích về cả chuyên môn lẫn tác phong nghiệp vụ.

Xin gửi lời cảm ơn đến bạn bè của tôi - tập thể lớp Việt Nhật K57, những người anh em đã luôn bên cạnh tôi trong 5 năm vừa qua. Cảm ơn các bạn vì đã sát cánh cùng tôi trong những phút giây vui vẻ, hạnh phúc và cả những khi thất bại. Nhờ đó mà tôi có thể trưởng thành được như ngày hôm nay.

Cuối cùng, con xin cảm ơn gia đình, bố, mẹ, ông, bà đã luôn là chỗ dựa tinh thần vững chắc và là nguồn động lực vô hạn giúp con vượt qua khó khăn không chỉ trong quá trình thực hiện đồ án mà còn trên toàn bộ con quá trình học tập, rèn luyện tại môi trường Đại học Bách khoa Hà Nội.

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Sự phát triển mạnh mẽ của khoa học công nghệ đi kèm với sự phát triển của Internet và điện thoại thông minh. Với lợi thế nhỏ gọn, tiện dụng và thông minh, smartphone đã trở thành một phần không thể tách biệt trong cuộc sống, hỗ trợ rất nhiều cho con người trong các hoạt động hằng ngày. Một trong những công dụng hữu ích phải kể đến đó là việc học tập trên điện thoại. Tuy nhiên, với lượng thông tin ngày càng nhiều và đa dạng như hiện nay, việc tìm kiếm cho được khoá học hay tài liệu để học phù hợp với mục đích và trình độ của bản thân mình là một vấn đề nan giải chưa có hướng giải quyết.

Đồ án sẽ đề xuất hệ thống tư vấn tài liệu học tiếng Anh cho người dùng trên điện thoại di động giúp cho người dùng có thể lựa chọn được tài liệu học phù tiếng Anh phù hợp với bản thân mình. Hệ thống được xây dựng dựa vào việc thu thập thông tin về trình độ và nguyện vọng của người dùng, đồng thời dựa vào đánh giá của người dùng về những kết quả tư vấn. Qua đó xây dựng được hồ sơ người dùng và tư vấn ra những kết quả phù hợp với họ.

Nội dung đồ án sẽ được trình bày dưới các phần sau:

- **Chương 1:** Đặt vấn đề và định hướng giải pháp
- **Chương 2:** Cơ sở lý thuyết
- **Chương 3:** Phân tích thiết kế xây dựng ứng dụng
- **Chương 4:** Kết quả thực hiện
- **Chương 5:** Kết luận và hướng phát triển.

Mục lục

Danh sách hình vẽ

Danh sách bảng

Chương 1

Đặt vấn đề và định hướng giải pháp

1.1 Đặt vấn đề

Sự phát triển mạnh mẽ của khoa học công nghệ đi kèm với sự phát triển của Internet và điện thoại thông minh. Với lợi thế nhỏ gọn, tiện dụng và thông minh, smartphone đã trở thành một phần không thể tách biệt trong cuộc sống, hỗ trợ rất nhiều cho con người trong các hoạt động hằng ngày. Người dùng smartphone ngoài chức năng liên lạc, họ còn sử dụng nhiều loại dịch vụ khác nhau như giải trí, định vị, mua sắm, thanh toán trực tuyến,... Một trong những công dụng hữu ích phải kể đến đó là việc học tập trên điện thoại.

Nhiều chuyên gia nhận định, cùng với sự phát triển của Internet, Giáo dục trực tuyến (E-Learning) đang dần trở nên phổ biến nhờ tính tiện dụng, tương tác cao và nhu cầu rất lớn từ cộng đồng học tập. Với chiếc smartphone trên tay, chỉ bằng một vào thao tác tìm kiếm đơn giản, người dùng đã có thể truy cập đến vô vàn các khoá học, sách tham khảo, bài giảng online khác nhau. Tuy nhiên, với lượng thông tin ngày càng nhiều và đa dạng như hiện nay, không phải khoá học, tài liệu nào cũng phù hợp với mục đích và trình độ học của người dùng. Việc theo học một khoá học không phù hợp sẽ dẫn đến việc người học mất dần hứng thú, động lực học, gây ra tốn kém thời gian mà hiệu quả thu được là không cao.

Để giải quyết bài toán tìm kiếm và lựa chọn thông tin cần thiết phù hợp với nhu cầu người dùng, các hệ thống thông tin thường tích hợp một hệ lọc để đưa ra chỉ những thông tin mà người dùng có thể quan tâm. Hệ thống này được gọi là hệ thông tư vấn, hay hệ gợi ý (Recommender System). Hệ thống tư vấn dựa trên các thông tin thu thập được từ người dùng, phân tích xử lý và đối chiếu với cơ sở tri thức, từ đó đưa ra được những thông tin hữu dụng giúp cho người dùng đạt được mục đích của mình. Hiện nay trên thế giới đã có rất nhiều hệ thống tư vấn được tích hợp ứng dụng trong nhiều lĩnh vực khác nhau như thương mai điện tử, phim ảnh, âm nhạc, sách,... Tuy nhiên rất ít trong số đó dùng cho mục đích tư vấn tài liệu học, và hầu hết các hệ thống giáo dục trực tuyến không được tích hợp chức năng tư vấn.

Để giải quyết vấn đề trên, đồ án này đề xuất một hệ thống tư vấn tài liệu học tiếng Anh cho người dùng trên điện thoại di động giúp cho người dùng có thể lựa chọn được tài liệu học phù tiếng Anh phù hợp với bản thân mình. Hệ thống được xây dựng dựa vào việc thu thập thông tin về trình độ và nguyện vọng của người dùng, đồng thời dựa vào đánh giá của người dùng về những kết quả tư vấn. Qua đó xây dựng được hồ sơ người dùng và tư vấn ra những kết quả phù hợp với họ.

1.2 Định hướng giải quyết

Hệ thống tư vấn tài liệu học tiếng Anh sẽ được xây dựng theo mô hình client-server. Trong đó client sẽ là ứng dụng điện thoại di động được viết trên nền tảng Android đóng vai trò một giao diện tương tác, thu thập và xử lý thông tin người dùng và trả về kết quả tư vấn. Server lưu trữ trên nền tảng Firebase của Google sẽ đóng vai trò là hệ cơ sở tri thức, lưu giữ thông tin hồ sơ người dùng, tài liệu tiếng Anh và mạng lưới từ khoá (keyword network) sử dụng thực hiện tư vấn.

Trên client, người dùng sẽ được yêu cầu cung cấp về thông tin trình độ cũng như mong muốn học của họ, cụ thể là :

Trình độ:

- Thời điểm bắt đầu học
- Trình độ đọc hiểu
- Vốn từ vựng
- Vốn ngữ pháp

Nguyện vọng:

- Mục đích học của người dùng

Từ tập câu hỏi trên và xây dựng thành profile người dùng. Dựa trên profile này, hệ thống tiến hành context - match giữa hồ sơ người dùng và thuộc tính của từng tài liệu và chấm điểm độ phù hợp, sau đó hiện kết quả cho người dùng về khoá học, tài liệu tương ứng với trình độ và nguyện vọng của họ. Người dùng sau đó sẽ tiến hành đánh giá xem các kết quả tư vấn trả về có phù hợp với họ hay không. Hồ sơ người dùng sẽ có sự thay đổi dựa trên các đánh giá đó. Cứ tiếp tục như vậy, các kết quả tư vấn tiếp theo sẽ càng ngày chính xác đúng với nhu cầu cá nhân của người dùng hơn.

Chương 2

Cơ sở lý thuyết

Trong nội dung của chương này, đồ án sẽ trình bày về các kiến thức cơ bản cũng như như các thuật toán được sử dụng trong hệ thống tư vấn.

2.1 Khảo sát các hệ tư vấn đã có

Hệ thống tư vấn đã trở thành một đề tài khá phổ biến trong khoảng thời gian gần đây. Trong lĩnh vực xây dựng các hệ thống tư vấn trong quá khứ, người ta đã làm việc và nghiên cứu khá nhiều và ứng dụng rộng rãi trên các lĩnh vực khác nhau. Hầu hết công việc chủ yếu tập trung phát triển những phương pháp gợi ý những những đối tượng ưa thích đến cho người dùng. Ví dụ như những trang web gợi ý những bộ phim, gợi ý những quyển sách mà người dùng có thể yêu thích. Hệ thống tư vấn thông thường sẽ tiếp cận và giải quyết vấn đề theo 1 trong 2 hướng: Lọc dựa trên nội dung (*content-based filtering*) hoặc lọc cộng tác (*collaborative filtering*). Tuy nhiên trên thực tế, việc áp dụng cả 2 hướng để giải quyết vấn đề cũng thường được cân nhắc trong việc giải quyết bài toán trong thực tế. (*hybrid recommender system*)

2.1.1 Lọc cộng tác

là phương pháp được xây dựng dựa trên lý thuyết : "Những người có cùng hứng thú/mong muốn/sở thích về một vấn đề gì đó trong quá khứ thì có thể họ sẽ cũng có cùng hứng thú/mong muốn/sở thích trong tương lai". Ví dụ: hai người dùng A và B có chung sở thích ăn uống, họ đã mua các đồ ăn giống nhau. Nếu B còn thích thêm cả Cocacola nữa thì rất có thể A cũng thích, nên ta có thể gợi ý cho A mua thêm Cocacola.

Phương pháp này thực hiện việc thu thập và đánh giá một lượng lớn thông tin về hành vi, sở thích của người dùng để tiên đoán sở thích của họ dựa trên sự giống nhau về thông tin giữa các người dùng. Ưu điểm của phương pháp này là nó không phải phụ thuộc vào việc nhận định, đánh giá để “hiểu được” nội dung của đối tượng tư vấn mà vẫn có thể đưa ra được kết quả thoả mãn mong muốn của người dùng. Tuy nhiên mặt

hạn chế của phương pháp trên là nó cần một lượng lớn dữ liệu người dùng đa dạng để có thể hoạt động chính xác được. Và việc tính toán hành vi của từng người dùng cũng tiêu hao một lượng lớn tài nguyên máy tính.

2.1.2 Lọc dựa trên nội dụng

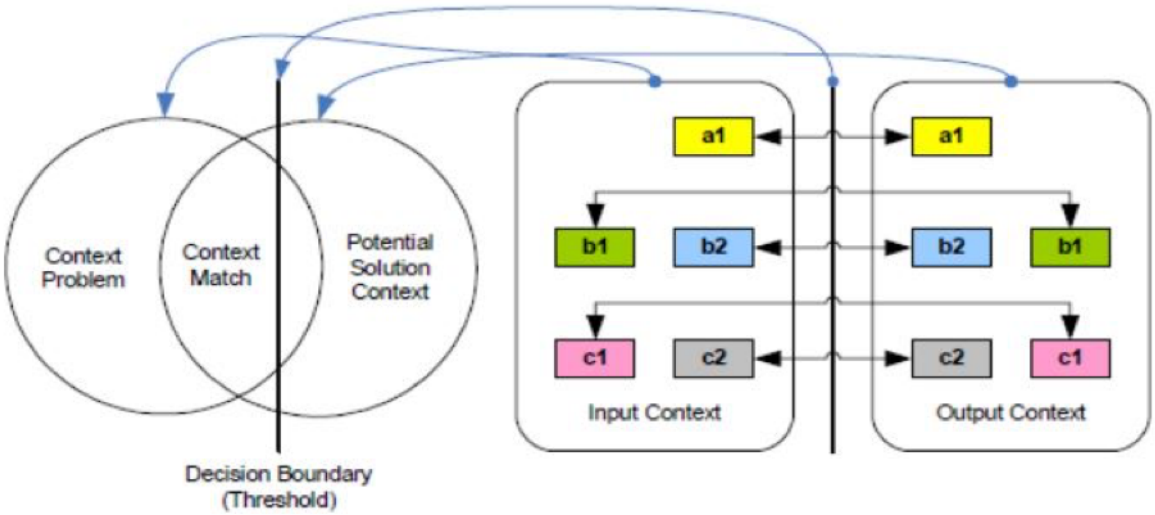
là phương pháp mà người ta quân tâm đến 2 thực thể chính là người dùng và đối tượng được khuyến nghị đến cho người dùng. Quá trình lọc thực hiện bắt đầu từ việc xây dựng một hồ sơ người dùng (user profile) là tập các ưu tiên về sở thích của người dùng về đối tượng, và hồ sơ miêu tả đối tượng. Sau đó tiến hành lọc kết quả dựa vào phương pháp context-matching. Đây là phương pháp bắt nguồn từ lĩnh vực nghiên cứu triết xuất thông tin và lọc thông tin.

Hệ thống tư vấn trong tài liệu này áp dụng phương pháp lọc dựa trên nội dung, và cân nhắc thêm lọc cộng tác khi hệ thống đã phát triển sau này.

2.2 Thuật toán Context-matching

Thuật toán context-matching[5] sử dụng để giải quyết bài toán cần Context-matching với Partial-matching (khi mà khả năng đạt được "perfect match" là thấp). Xây dựng các input context, output context và hàm đánh giá giữa các properties của chúng. Mục tiêu cuối cùng là một giá trị Boolean thể hiện sự phù hợp, hay không phù hợp của thực thể được tính đến.

Mô hình thuật toán [6] được biểu diễn ở hình sau :



Hình 2.1: Mô hình thuật toán Context Matching

2.2.1 Tổng quan

Cấu trúc cơ bản của thuật toán là $ON < event > IF < condition > THEN < action >$. Trong đó $< event >$ có thể là context data cần tính (event khởi động), hoặc là kết quả trả về của một lần so sánh ở phía trên, $< condition >$ là cách so sánh giữa một input context property với một output context property sẽ được báo cáo dưới đây, $< action >$ là kết quả của việc đánh giá trên $< condition >$ và nó có thể là việc tiếp tục so sánh input và output tiếp theo hoặc là giá trị Boolean quyết định độ phù hợp của context (action cuối cùng).

2.2.2 Input context và output context

Input context là nguyên mẫu để so sánh, các kết quả phù hợp là những kết quả phải có độ match với input context lớn nhất định. Output context là giải pháp tiềm năng cho vấn đề cần giải quyết trong bài toán, là một số lượng những thực thể đem so sánh với input context, được trích xuất từ cấu trúc dữ liệu. Từ input và output context, cần xây dựng một bộ các context properties là những thuộc tính bên trong quyết định context. Việc so sánh sẽ là so sánh giữa các context property.

2.2.3 Các giá trị sử dụng

Giả sử áp dụng thuật toán cho bộ context properties $[a1, b1, b2, c1, c2]$. Ta có :

- w: giá trị trong khoảng [0.10...1.00] phản ánh mức độ ưu tiên của thuộc tính
- e: giá trị đánh giá xem thuộc tính ở input và output có match nhau không, e được biểu diễn trong khoảng [0...1]
- $av = (e * w)$: giá trị thực tế đánh giá độ match của thuộc tính, nằm trong khoảng [0.00...1.00]
- $sav = av(a_1) + av(b_1) + av(b_2) + av(c_1) + av(c_2)$: tổng giá trị av
- mpv: giá trị sav cao nhất có thể đạt được. Nó thể hiện trường hợp mà tất cả các context property đều match nhau.
- $rv = \dfrac{sav}{mpv}$: giá trị trả về sau so sánh. Nó đánh giá tổng quan xem kết quả context match đến mức độ nào, nằm trong khoảng [0.00...1.00].
- t: giá trị cho ngưỡng đạt [0.10...1.00]. Context match phù hợp khi so sánh ta có giá trị $rv > t$.

2.2.4 Các bước thực hiện thuật toán

- Bước 1: Đánh giá context match cho từng context property -> rút ra giá trị e .
- Bước 2: Thiết lập giá trị w đã được định sẵn cho từng context property
- Bước 3: Tính giá trị av của thuộc tính $av = (e * w)$
- Bước 4: Tính tổng các giá trị av của cả quá trình đánh giá $sav = av(a_1)+av(b_1)+av(b_2)+av(c_1)+av(c_2)$
- Bước 5: Tính giá trị sav cao nhất có thể đạt $mpv = w(a_1)+w(b_1)+w(b_2)+w(c_1)+w(c_2)$
- Bước 6: Tính giá trị kết quả trả về $rv = \frac{sav}{mpv}$
- Bước 7: Sử dụng giá trị ngưỡng đã có sẵn để xác định xem output context có match với input context hay không . IF $(rv) >= (t)$ THEN $context - match = true$ [1] or IF $(rv) < (t)$ THEN $context - match = false$ [0]

2.3 Kỹ thuật Kansei Engineering

2.3.1 Tổng quan

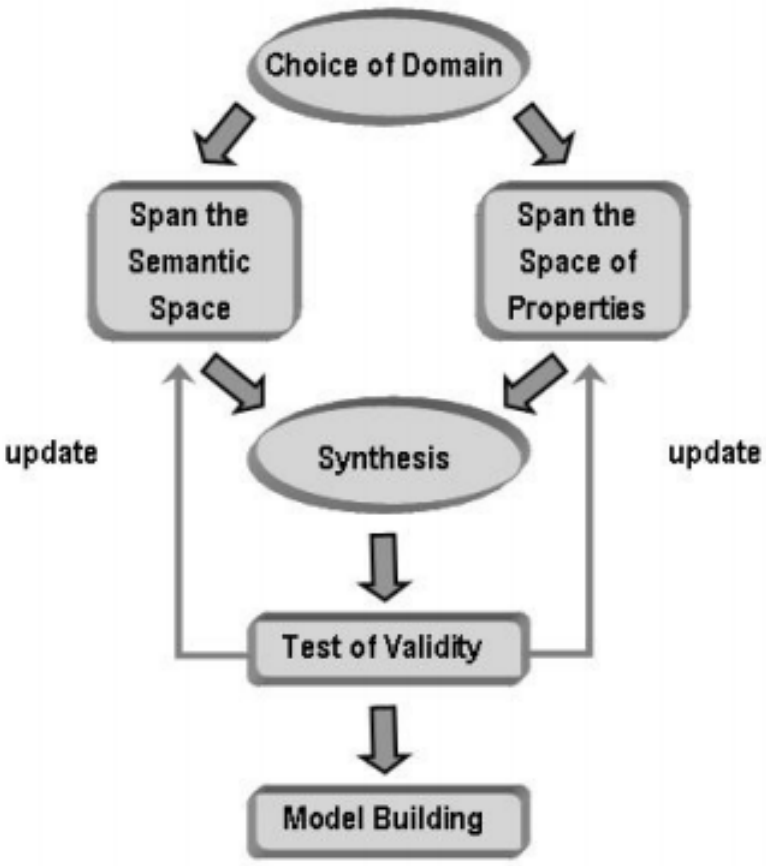
Kansei Engineering là kỹ thuật tích hợp khía cạnh cảm xúc của con người vào trong quá trình xây dựng sản phẩm, nhằm mục đích tạo ra được sản phẩm phù hợp với yêu cầu và mong muốn của người dùng. Nó là việc mang lại sự hài lòng, thoả mãn cho người dùng một cách có khoa học. Để đạt được điều đó, trải nghiệm của người dùng dựa trên các sản phẩm tương tự được thu thập và phân tích, từ đó thiết lập mô hình dự đoán mối quan hệ giữa cảm xúc của con người và các đặc tính vật lý của sản phẩm.

3 điểm chính được chú trọng trong Kansei Engineering đó là :

- Làm thế nào để hiểu được cảm xúc nội tâm của người dùng
- Làm thế nào để phản ánh hiểu biết đó vào trong việc phát triển sản phẩm
- Làm thế nào để xây dựng một hệ thống có tổ chức theo định hướng Kansei Engineering

2.3.2 Mô hình

Mặc dù nhiều mô hình Kansei Engineering khác nhau phục vụ cho các bài toán cụ thể khác nhau. Nhưng về cơ bản, tất cả đều tuân theo mô hình tổng quát [8] sau đây:



Hình 2.2: Mô hình tổng quát Kansei Engineering

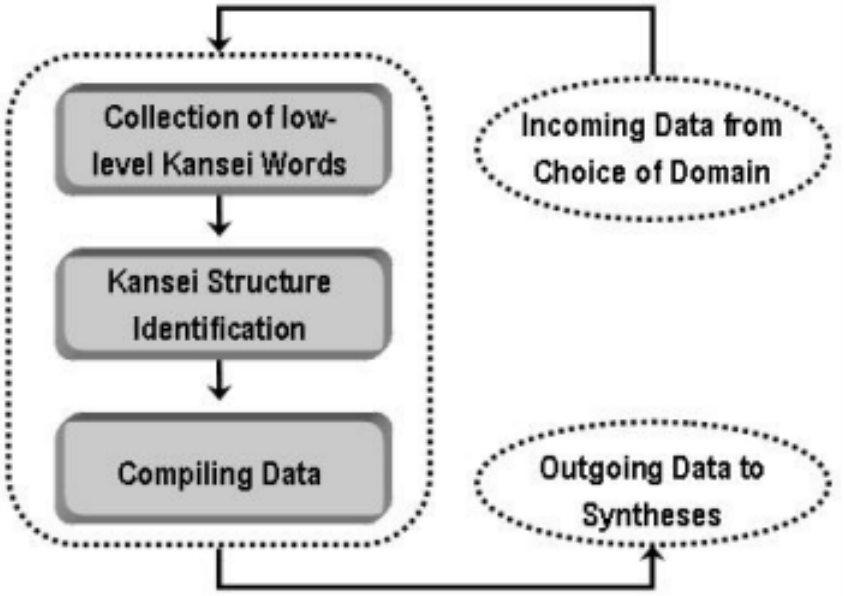
Chọn miền chủ đề

Chủ đề mang ý nghĩa là ý tưởng tổng quát đằng sau sản phẩm. việc lựa chọn miền chủ đề trong đó bao gồm lựa chọn loại sản phẩm sẽ xét đến, đối tượng người dùng sử dụng và các đặc tả cụ thể khác. Tiếp theo đó, miền không gian Kansei và miền không gian thuộc tính sản phẩm sẽ được xác định. Chúng sẽ được phân tích trong bước tổng hợp để tìm ra mối quan hệ giữa thuộc tính của sản phẩm và cảm xúc của người dùng. Từ đó xác định được thuộc tính sản phẩm sẽ ảnh hưởng đến người dùng như thế nào.

Mở rộng miền không gian Kansei

Osgood et al[4] đề xuất rằng mọi vật đều có thể được miêu tả bằng miền không gian vector cảm xúc. Từ miền chủ đề đã xác định, các từ khoá Kansei Word sẽ được thu thập. Kansei Word là các danh từ, tính từ cảm xúc mà người dùng có thể sử dụng để miêu tả về sản phẩm. Số lượng từ khoá thu thập được sẽ đa dạng tuỳ theo từng loại chủ đề khác nhau, giao động từ 100 đến 1000 từ khoá khác nhau. Tuy nhiên, do yếu tố chủ quan của con người, một số từ khoá có thể mang ý nghĩa tương đồng hoặc gần giống nhau. Do vậy tập từ khoá này sau đó sẽ được nhóm lại với nhau bằng phương

pháp thủ công hoặc toán học. cuối cùng, chúng phân tích để chọn ra được những từ khoá đại diện mang ý nghĩa tổng quát độc lập với nhau.



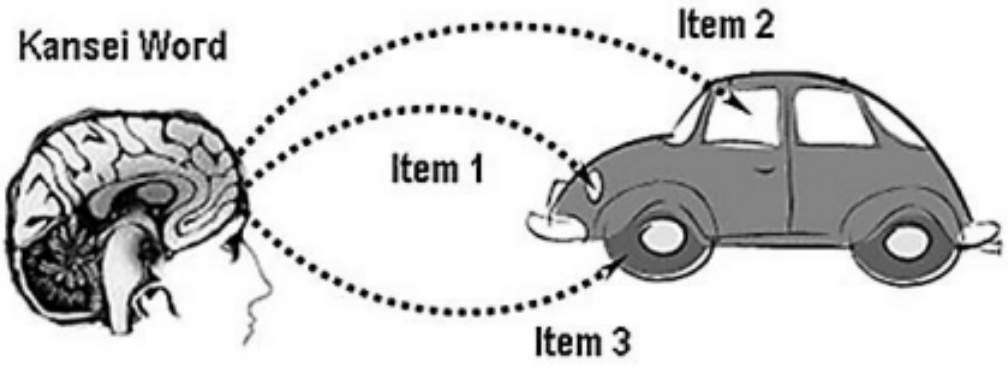
Hình 2.3: Quy trình mở rộng miền không gian Kansei

Mở rộng miền không gian thuộc tính sản phẩm

Tương tự như bước Mở rộng miền không gian Kansei, ở bước này, các thuộc tính của sản phẩm như hình thức, màu sắc, thể loại, nội dung..v..v... được thu thập từ các sản phẩm khác. Các sản phẩm được chọn để khai thác thuộc tính có thể là các sản phẩm đang lưu hành trên thị trường, đề xuất của khách hàng, hoặc thậm chí là ý tưởng thiết kế mới. Tương tự, do có thể có sự tương đồng giữa các thuộc tính gây ảnh hưởng đến độ chính xác nên các thuộc tính được chọn ra là các thuộc tính tiêu biểu nhất của sản phẩm.

Tổng hợp

Trong bước tổng hợp, 2 miền không gian từ khoá Kansei và thuộc tính sản phẩm sẽ được móc nối vào với nhau. Mỗi từ khoá Kansei Word sẽ tương ứng với một hoặc nhiều thuộc tính của sản phẩm, ảnh hưởng trực tiếp đến các thuộc tính đó. Như trong nghiên cứu về thiết kế lon bia của Ishihara et al. (1998)[3] cho thấy rằng, cảm giác "đắng" của người uống chịu ảnh hưởng bởi màu sắc và hình dạng logo lon bia, với màu đen và logo vuông cho người uống cảm giác "rất đắng", trong khi màu trắng và logo hình bầu dục tạo cảm giác ngược lại.



Hình 2.4: Pha tổng hợp

Có nhiều phương pháp tổng hợp, trong đó phổ biến là:

- Category Identification
- Regression Analysis/Quantification Theory Type I
- Rough Sets Theory
- Genetic Algorithm
- Fuzzy Sets Theory

Kiểm tra độ chính xác

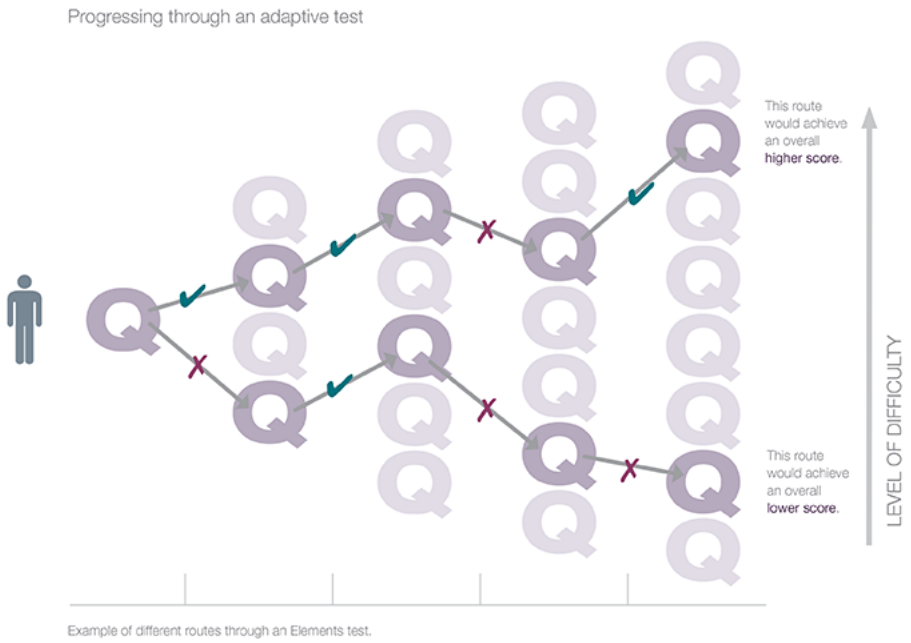
Trước khi mô hình đã xây dựng có thể đem vào sử dụng. Nó cần được phải được kiểm tra độ chính xác, đánh giá xem nó có đủ độ tin cậy và phù hợp với thực tế hay không. Trong trường hợp mô hình không đạt yêu cầu, cần có sự thay đổi trong các bước trên rồi tiếp tục quy trình đánh giá lại cho đến khi ra được kết quả đạt yêu cầu cuối cùng.

2.4 Computerized Adaptive Testing

2.4.1 Tổng quan

Computerized Adaptive Testing - Bài thi tương tác tuỳ biến qua máy tính là hình thức làm bài thi trên máy tính. Trong đó, nội dung và độ khó của câu hỏi sẽ được tuỳ chỉnh sao cho phù hợp với năng lực của thí sinh dự thi. Sau mỗi câu hỏi, trình độ của thí sinh được cập nhập, quyết định độ khó của câu hỏi tiếp theo. Nếu thí sinh trả lời tốt các câu hỏi trước đó, hệ thống sẽ đưa ra các câu hỏi có độ khó cao hơn. Ngược lại,

nếu thí sinh trả lời sai nhiều, độ khó của câu hỏi tiếp theo sẽ được giảm xuống. Và khi đến một ngưỡng nhất định, đã xác định chắc chắn được trình độ của thí sinh thì bài kiểm tra sẽ được hoàn tất.



Hình 2.5: Computerized Adaptive Test[1]

2.4.2 Mô hình

Mô hình CAT cơ bản gồm 4 bước sau đây:

- Chọn ra câu hỏi từ tập câu hỏi dựa vào trình độ thí sinh
- Câu hỏi được chọn sẽ được thí sinh trả lời, đúng hoặc sai
- Trình độ của thí sinh sẽ được cập nhập, dựa vào kết quả của tất cả những câu hỏi đã trả lời
- Lặp lại quá trình trên cho đến khi một ngưỡng quy định trước đạt được.

2.4.3 Ưu điểm

Với phương thức khác hoàn toàn với bài kiểm tra trên giấy thông thường, với CAT mỗi thí sinh được nhận các bài thi hoàn toàn khác nhau. Hơn nữa, số lượng câu hỏi thí sinh cần phải trả lời sẽ được giảm thiểu. Thí sinh sẽ không phải trả lời những câu hỏi quá khó so cũng như quá dễ so với trình độ của họ, do đó thời gian làm bài trung bình sẽ ngắn hơn so với bài kiểm tra thông thường song vẫn cho ra kết quả chính xác tương tự.

2.5 Các cơ sở lý thuyết về công nghệ sử dụng

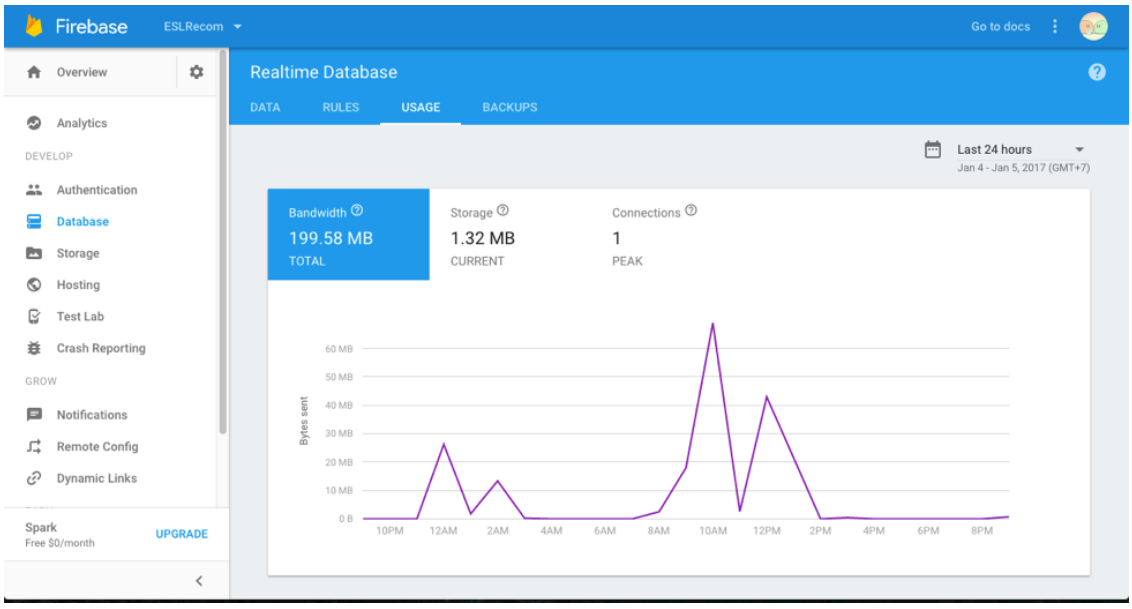
2.5.1 Firebase

Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Firebase hỗ trợ tối đa đối với những ứng dụng Backend, nó bao gồm các tiện ích lưu trữ dữ liệu, xác thực người dùng, static, hosting,... Vì thế giúp cho lập trình viên giảm thiểu công việc, và tập trung nâng cao trải nghiệm người dùng.

Realtime Database – Cơ sở dữ liệu thời gian thực

Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Cụ thể hơn là bạn có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.



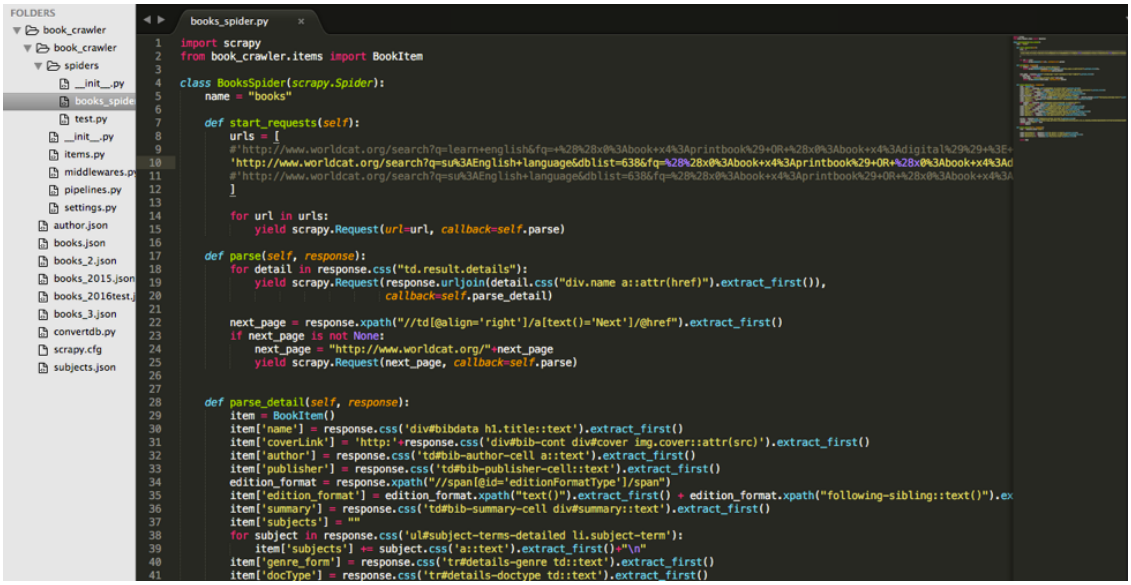
Hình 2.6: Cơ sở dữ liệu thời gian thực

2.5.2 Scrapy

Scrapy là một framework được viết bằng Python, nó cấp sẵn 1 cấu trúc tương đối hoàn chỉnh để thực hiện việc crawl và extract data từ website một cách nhanh chóng và dễ dàng. Bạn muốn lấy dữ liệu từ các website nhưng dữ liệu đó quá lớn để copy rồi paste vào database của bạn, scrapy hỗ trợ bạn làm điều đó. Việc lấy dữ liệu website

Hình 2.7: Lưu trữ cơ sở tri thức chia sẻ giữa các thiết bị

hoàn toàn tự động nhanh chóng và việc sử dụng scrapy cũng rất đơn giản giúp bạn tiếp kiệm được nhiều thời gian và công sức.



Hình 2.8: Crawl tài liệu từ thư viện điện tử WorldCat.org

2.5.3 Thuật toán $tf - idf$

Viết tắt của thuật ngữ tiếng Anh term frequency – inverse document frequency,$tf - idf$ là thuật toán dùng để tìm trọng số của một từ trong văn bản thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.[2]

Nguyên lý cơ bản của $tf - idf$ là "độ quan trọng của một từ sẽ tăng lên cùng với số lần xuất hiện của nó trong văn bản và sẽ giảm xuống nếu như từ đó xuất hiện trong nhiều văn bản khác". Do đó trọng số của một từ t trong tài liệu f sẽ được tính bằng $tf * idf$, với tf là độ phổ biến của từ t trong tài liệu f và idf là nghịch đảo độ phổ biến của từ t trong các tài liệu còn lại của tập tài liệu. Cụ thể theo công thức sau đây:

$$tf(t,d) * idf(t,D) = \frac{f_d(t)}{\max_{w \in d} f_d(w)} * \log \frac{|D|}{|d \in D : t \in d|} \tag{2.1}$$

Lấy ví dụ một văn bản chứa 100 từ, trong đó từ "cat" xuất hiện 3 lần. Giá trị tf của "cat" sẽ là $\frac{3}{100} = 0.03$. Tiếp theo, giả dụ có 10 triệu văn bản và từ "cat" xuất hiện ở 1000 văn bản trong đó. Giá trị idf sẽ là $\log \frac{10,000,000}{1,000} = 4$. Vậy trọng số $tf - idf$ của từ khoá "cat" sẽ là $0.03 * 4 = 0.12$.

Chương 3

Phân tích thiết kế hệ thống

3.1 Phân tích hệ thống

Các yêu cầu của hệ thống:

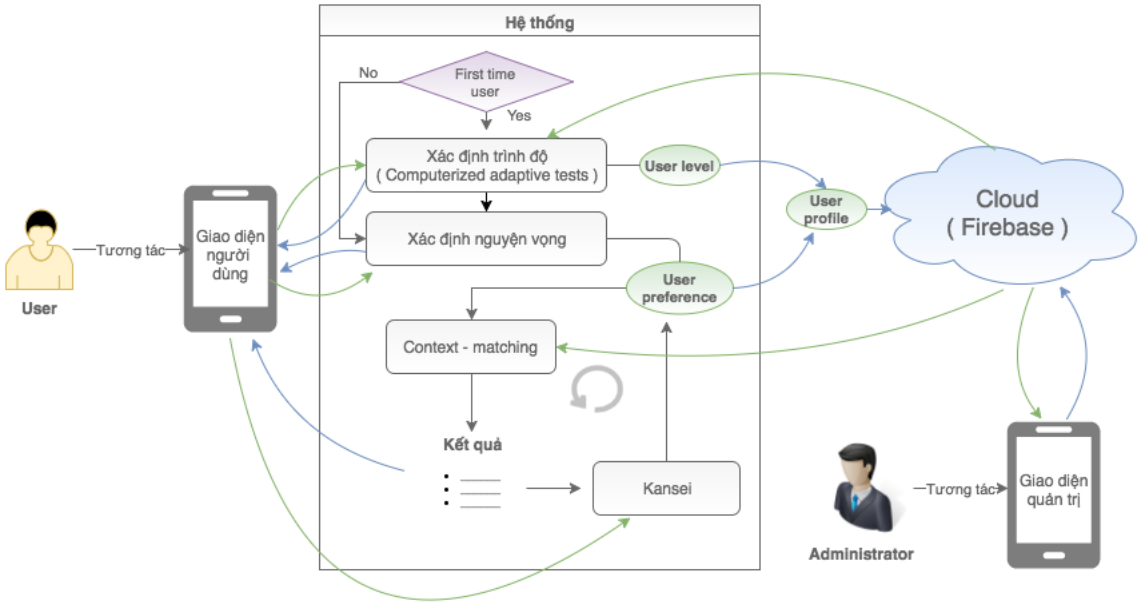
- Cung cấp các phương pháp để xác định được trình độ, nguyện vọng và các yếu tố ưu tiên khác trong việc học tiếng Anh của người dùng thông qua giao diện đơn giản, dễ tương tác. Qua đó, đưa ra được các kết quả tư vấn là tài liệu, sách, video, bài giảng Tiếng Anh..v..v... tương ứng.
- Cung cấp giao diện quản lý cho quản trị viên dễ dàng thực hiện các thao tác quản lý thông tin người dùng, quản lý hệ cơ sở tri thức gồm tập câu hỏi kiểm tra và tài liệu tiếng Anh.

Qua việc khảo sát các hệ thống tư vấn đã và đang được triển khai, đồng thời để giải quyết các yêu cầu đặt ra ở trên, hệ thống đề xuất xây dựng trong đề tài này sẽ có cấu trúc gồm các thành phần sau:

- **Module xác định trình độ:** có nhiệm vụ xác định trình độ của người sử dụng, thông qua việc thực hiện bài kiểm tra General English Test . Sử dụng kĩ thuật Computerized Adaptive Testing, các câu hỏi đưa ra cho người dùng sẽ được tuỳ biến sao cho độ khó phù hợp với năng lực của người dùng. Nhờ vậy, số lượng câu hỏi mà người dùng cần trả lời để xác định được trình độ của họ là ít hơn bài kiểm tra truyền thống, song vẫn cho ra kết quả chính xác như tương tự. Kết quả của bước này sẽ cho ra User level bao gồm trình độ đọc hiểu, vốn từ vựng và vốn ngữ pháp.
- **Module xác định nguyện vọng:** có nhiệm vụ nhận input về nguyện vọng từ người dùng, cụ thể là chủ đề mà người dùng muốn học. Có thể đưa ra các gợi ý cho người dùng về các chủ đề phổ biến. Kết quả bước này sẽ cho ra User preference là các chủ đề người dùng muốn học dưới dạng keyword.

- **Context-matching:** thực hiện nhận thông tin User level và User preference tổng hợp thành User profile. Sau đó sử dụng thuật toán Context-matching tiến hành matching với profile tài liệu và trả về những kết quả có độ khớp cao nhất.
- **Kansei:** kết quả sau khi Context Matching sẽ được trả về cho người dùng đánh giá trên thang cảm xúc từ "rất thích" cho đến "rất ghét". Dựa vào đánh giá, những thuộc tính trong profile tài liệu ứng với "thích" sẽ được cập nhập vào User preference và lấy nó làm cơ sở context matching các kết quả tiếp theo.
- **Hệ cơ sở tri thức:** sử dụng cơ sở dữ liệu online của Firebase làm cơ sở tri thức cho hệ thống. Nhiệm vụ của nó là trao đổi thông tin với client, cập nhập thông tin mới đảm bảo tính đồng bộ cho toàn hệ thống.
  Dữ liệu được lưu trữ bao gồm:
  - Dữ liệu câu hỏi và đáp án General English Test
  - Thông tin người dùng : id, loại người dùng, trình độ, nguyện vọng.
  - Dữ liệu tài liệu học Tiếng Anh: tên, loại tài liệu, tác giả, miêu tả, nội dung ..v..v... và profile của tài liệu dưới dạng một tập keyword.

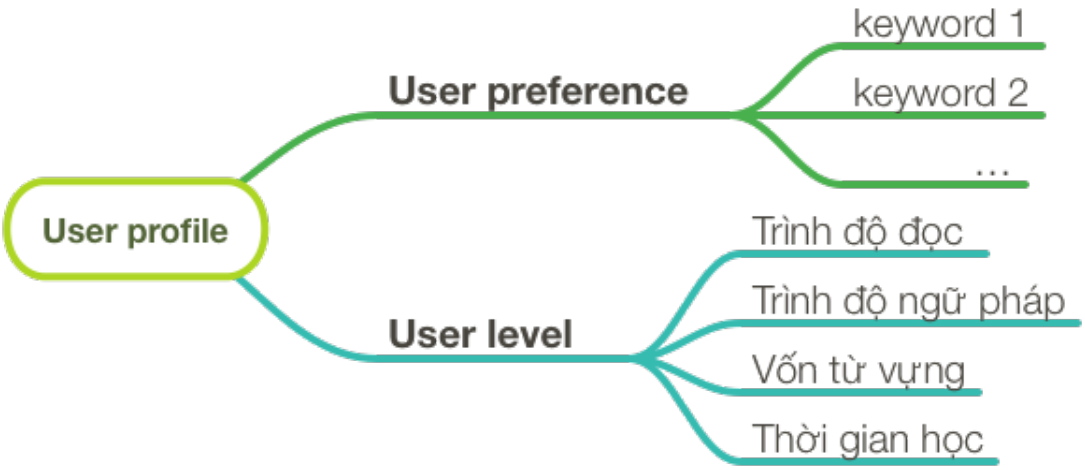
Sau đây là mô hình kiến trúc của ứng dụng:



Hình 3.1: Mô hình kiến trúc ứng dụng

3.2 Xây dựng hồ sơ người dùng

User profile bao gồm trình độ và nguyện vọng của người dùng. Những thông tin này sẽ được dùng trong quá trình matching với các tập dữ liệu bằng thuật toán Context-matching .



Hình 3.2: User profile

3.2.1 Xác định trình độ người dùng

Để xác định trình độ Tiếng Anh của người dùng, tất cả các khía cạnh sau đây cần được khai thác:

- Thời điểm bắt đầu học
- Trình độ đọc hiểu
- Trình độ ngữ pháp
- Vốn từ vựng

Bài kiểm tra tương tác CAT sẽ được sử dụng để đánh giá trình độ.Phương thức chọn câu hỏi và đánh giá bài thi CAT trong đề tài được xây dựng dựa trên mô hình Thử tỉ lệ xác suất nối tiếp [7] (Sequential probability ratio test).

Nguyên lý căn bản nằm bên trong mô hình này là xác suất hợp lý rời rạc. Giả sử từ quan sát thực tế cho thấy thí sinh có trình độ tiếng Anh xuất sắc đạt trung bình 85/100 điểm trong bài thi, trong khi thí sinh có trình độ thấp chỉ đạt 35/100 điểm. Dưới góc nhìn hệ thống có thể coi nó như tập luật $if....else$ sau đây:

1. Thí sinh có trình độ xuất sắc, đã nắm vững kiến thức và hiểu rõ câu hỏi cũng như phương pháp giải quyết, do vậy khả năng mà họ trả lời đúng câu hỏi sẽ là 85 %.

   Prob(Correct|Master) = .85 (P_m)

   Prob(Incorrect|Master) = .15

2. Ngược lại, thí sinh có trình độ thấp, trả lời phần lớn dựa trên may rủi, giác quan thứ 6 của bản thân, do vậy khả năng mà họ trả lời đúng câu hỏi sẽ là 35 %.

   Prob(Correct|Nonmaster) = .35 (P_n)

   Prob(Incorrect|Nonmaster) = .75

Trong bài kiểm tra CAT, với câu hỏi bất kì phù hợp trình độ được chọn ra trong tập câu hỏi đưa cho thí sinh. Quan sát trả lời của thí sinh, xác suất khả năng trình độ sẽ được tính bằng:

$$PR = \frac{P_m^t(1-P_m)^f}{P_m^t(1-P_m)^f} \tag{3.1}$$

trong đó P_m = khả năng thí sinh trình độ xuất sắc trả lời đúng câu hỏi
P_n = khả năng thí sinh trình độ thấp trả lời đúng câu hỏi
t = tổng số câu hỏi thí sinh trả lời đúng
f = tổng số câu hỏi thí sinh trả lời sai
Giá trị PR sau đó sẽ được đem so sánh với tập luật:

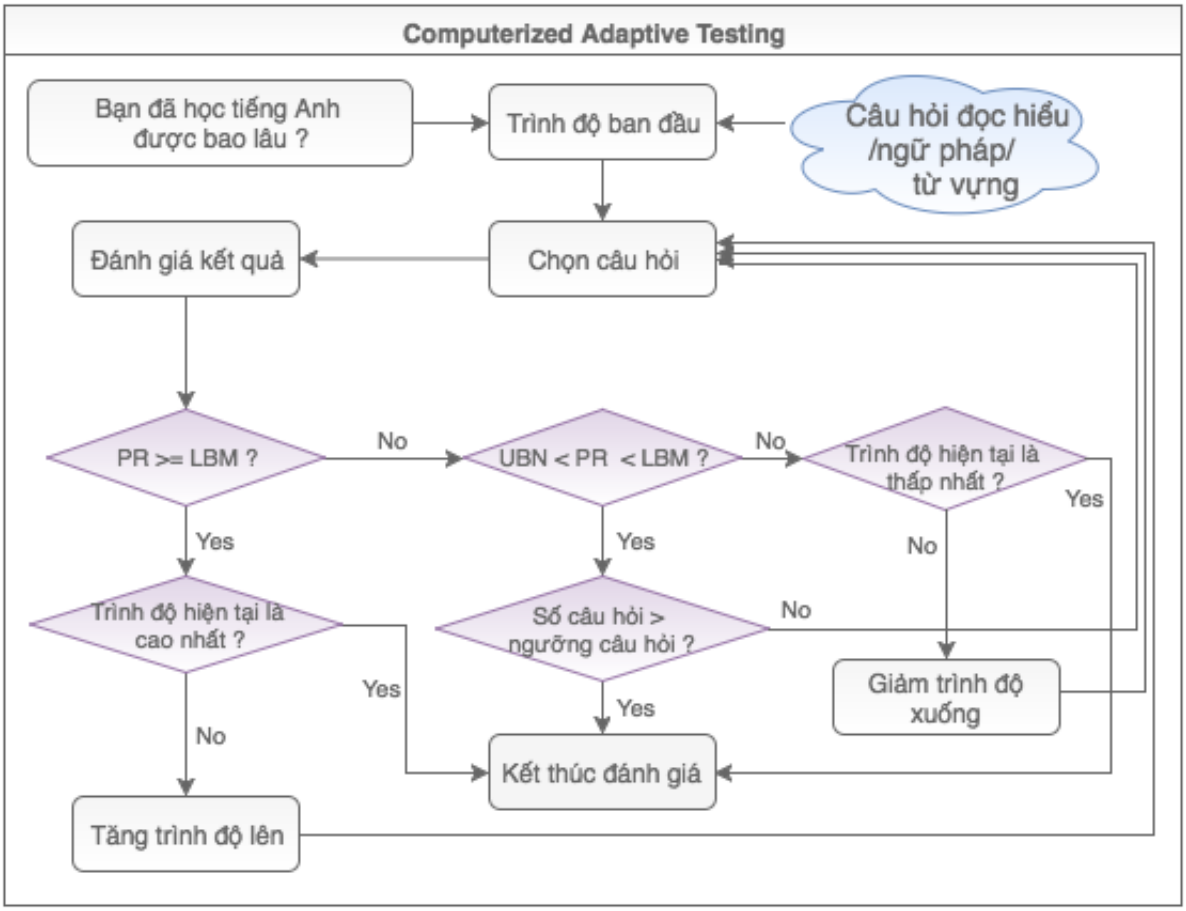
- Nếu $PR > UBN$ (Upper Bound Nonmastery: giá trị ngưỡng trên của độ không thuần thục) -> trình độ người dùng thấp hơn câu hỏi hiện tại, chọn câu hỏi tiếp theo ở trình độ thấp hơn để tiếp tục đánh giá .
- Nếu $PR < LBM$ (Lower Bound Mastery: giá trị ngưỡng dưới của độ thuần thục) -> trình độ người dùng nằm trên câu hỏi hiện tại, chọn câu hỏi tiếp theo ở trình độ cao hơn để tiếp tục đánh giá.
- Nếu $UBN < PR < LBM$, kết quả hiện tại chưa đủ để đánh giá trình độ người dùng, chọn câu hỏi tiếp theo ở cùng trình độ để tiếp tục đánh giá.

Trình độ của thí sinh được xác định khi PR lớn hơn giá trị UBN hoặc nhỏ hơn giá trị LBM. Tuy nhiên, xét trong trường hợp thực tế có khả năng xảy ra việc số lượng câu hỏi thí sinh trả lời sai bằng với số lượng câu hỏi thí sinh trả lời đúng. Điều này dẫn đến tình trạng thí sinh trả lời một số lượng lớn câu hỏi nhưng vẫn không xác định được trình độ. Điều kiện dừng sẽ được đưa vào để giải quyết trường hợp đó.

Hệ thống sẽ dừng bài kiểm tra đánh giá nếu như:

- Nếu $PR > UBN$ và *"độ khó hiện tại là thấp nhất"*
- Nếu $PR < LBM$ và *"độ khó hiện tại là cao nhất"*
- Nếu $UBN < PR < LBM$ và số lượng câu hỏi hiện tại vượt *ngưỡng câu hỏi*

Sau dây là mô hình đánh giá trình độ người dùng sử dụng trong đề tài:



Hình 3.3: Mô hình bài kiểm tra tương tác

Trước tiên, trình độ ban đầu của người dùng sẽ được xác định qua câu hỏi "Bạn đã học tiếng Anh được bao lâu rồi". Trình độ của một người nào đó thường tỉ lệ thuận với thời gian họ bỏ ra để học. do vậy ta có thể phần nào phán đoán được thông qua thông tin thời gian học. Đây là bước tiền đề trước khi đi vào thực hiện bài thi.

Tập câu hỏi đánh giá trình độ ban đầu, gồm hơn 70 câu hỏi lấy từ trang http://www.englishjet.com/. Hình thức của câu hỏi ở dạng trắc nghiệm với 4 đáp án. Chúng được phân loại vào từng tập câu hỏi khác nhau theo các *chủ đề* {*đọc hiểu, từ vựng, ngữ pháp*} và *trình độ* {*nhập môn, cơ bản, trung bình, khá, cao cấp*}. Dựa trên trình độ hiện tại của người dùng, hệ thống sẽ chọn bất kì một câu hỏi trong tập câu hỏi cùng trình độ ra để kiểm tra. Sau khi kết thúc một chủ đề, trình độ hiện tại của

người dùng sẽ được sử dụng làm trình độ ban đầu trong đánh giá chủ đề tiếp theo.

You can exchange the gift .....

A. so long that

B. while

C. as long as

D. meanwhile

E. whether

..... it is returned within seven days.

Việc đánh giá kết quả được thực hiện theo tập luật đã đề cập ở phía trên. Dựa vào quan sát thử nghiệm trong thực tế, hệ thống đề xuất trong đề tài sử dụng các giá trị $UBN = 0.02$, $LBM = 7$ và *ngưỡng câu hỏi* $= 5$.

Ví dụ, một thí sinh học tiếng Anh được 4 năm làm bài kiểm tra trình độ. Hệ thống sẽ dự đoán trình độ của anh ta nằm ở mức trung bình và đưa ra câu hỏi ở mức trình độ đó. Bắt đầu với chủ đề *từ vựng*, thí sinh này trả lời đúng liên tiếp 3 câu hỏi. PR của anh ta lúc đó sẽ là $\frac{0.85^3(1-0.85)^0}{0.35^3(1-0.35)^0} \approx 14.3236 > LBM = 7$. Điều kiện tăng trình độ thoả mãn, trình độ thí sinh được nâng lên mức khá. Tiếp tục quá trình đánh giá, lần này với câu hỏi trình độ khá, thí sinh lần lượt đạt kết quả Đúng-Sai-Đúng-Đúng-Sai, tương ứng với $PR = \frac{0.85^3(1-0.85)^2}{0.35^3(1-0.35)^2} \approx 3.305$. Do $UBN < PR < LBM$ và số lượng câu hỏi đã trả lời đạt ngưỡng, việc đánh giá trình độ từ vựng của thí sinh kết thúc. Kết quả là thí sinh đạt trình độ ***khá*** về *từ vựng*. Lấy trình độ *khá* làm trình độ ban đầu, hệ thống tiếp tục đánh giá chủ đề tiếp theo. Kết quả thu được là ***trung bình*** về *ngữ pháp* và ***cao cấp*** về *đọc hiểu*. Lấy trung bình 3 kết quả thu được, hệ thống kết luận trình độ tiếng Anh của thí sinh là ***khá***.

3.2.2 Xác định nguyện vọng người dùng

Nguyện vọng người dùng được xác định một cách đơn giản bằng việc người dùng trực tiếp nhập nội dung mình muốn học. Hệ thống sẽ cung cấp cho người dùng một giao diện nhập dữ liệu đơn giản. Người dùng sử dụng bàn phím sẽ nhập nguyện vọng của mình vào dưới dạng các keyword, phân tách nhau bởi dấu cách.

Để hỗ trợ xác định nguyện vọng, hệ thống sẽ đưa ra các keyword gợi ý dựa trên nội dung người dùng nhập vào. Các keyword gợi ý này được tổng hợp bằng việc phân

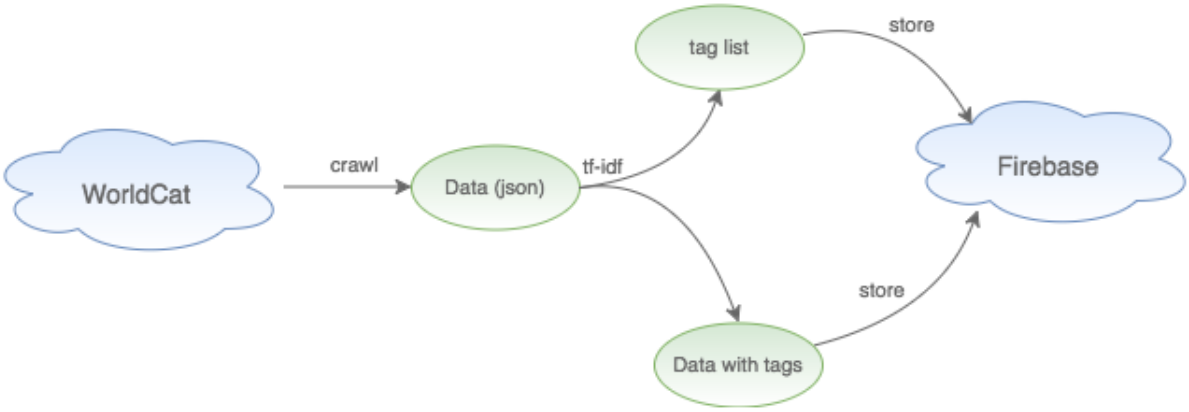
tích keyword các tài liệu Tiếng Anh sử dụng thuật toán $tf - idf$. Nội dung của công đoạn phân tích keyword sẽ được trình bày cụ thể ở phần sau.

Sau hai bước trên, một user profile như ví dụ sau được xây dựng.

```
'userProfile': {
        'id': 1,
        'vocabProfi': 'upper-intermediate',
        'grammarProfi': 'intermediate',
        'readingProfi': 'advanced',
        'overallProfi': 'upper-intermediate',
        'preference': ['ietls', 'grammar', 'advanced']
    }
```

3.3 Thu thập và xử lý tài liệu học tiếng Anh

Đây là bước tiền xử lý trước khi đi vào xây dựng hệ thống.



Hình 3.4: Mô hình thu thập và xử lý tài liệu học tiếng Anh

3.3.1 Thu thập dữ liệu

Tài liệu học tiếng Anh hệ thống sử dụng được lấy từ www.worldcat.org . WorldCat được biết đến như là một CSDL liên hợp toàn cầu. Là một thư viện điện tử chứa đựng dữ liệu từ 72,000 thư viện ở hơn 170 quốc gia và vùng lãnh thổ, WorldCat chứa một lượng dữ liệu khổng lồ gồm hơn 330 triệu bản ghi, với số lượng ngôn ngữ cực kỳ đa dạng, gồm gần 500 ngôn ngữ trên toàn thế giới, trong đó tiếng Anh chiếm khoảng 38%, tiếng Đức khoảng 13%, tiếng Pháp khoảng 9%, ngoài ra là các ngôn ngữ khác như tiếng Tây Ban Nha, tiếng Trung, tiếng Nhật, tiếng Hàn, và bao gồm cả tiếng Việt (cho dù chỉ là một tỷ lệ nhỏ). Nhiều chuyên gia đánh giá rằng WorldCat có thể bao

gồm tới trên 70% lượng tài liệu có trên toàn cầu, và là bộ CSDL thư mục toàn diện nhất thế giới từ trước tới giờ.

Sử dụng framework ***Scrapy***, function sau đây được viết để trích xuất dữ liệu từ WorldCat.

```
import scrapy
from book_crawler.items import BookItem

class BooksSpider(scrapy.Spider):
  name = "books"
  page = 0

  def start_requests(self):
        url = 'http://www.worldcat.org/search?q=kw%3Aenglish&fq
           =yr%3A2014..2017+%3E+%3E+-mt%3Afic+%3E+ln%3Aeng&qt=
           advanced&dblist=638'
        yield scrapy.Request(url=url, callback=self.parse)

  def parse(self, response):
        for detail in response.css("td.result.details"):
              yield scrapy.Request(response.urljoin(detail.
                 css("div.name a::attr(href)").extract_first()
                 ), callback=self.parse_detail)
        next_page = response.xpath("//td[@align='right']/a[text
           ()='Next']/@href").extract_first()
        BooksSpider.page += 1
        if next_page is not None and BooksSpider.page <= 500:
              next_page = "http://www.worldcat.org/"+
                 next_page
              yield scrapy.Request(next_page, callback=self.
                 parse)

  def parse_detail(self, response):
        item = BookItem()
        item['name'] = response.css('div#bibdata h1.title::text
           ').extract_first()
        item['coverLink'] = 'http:'+response.css('div#bib-cont
           div#cover img.cover::attr(src)').extract_first()
        item['author'] = response.css('td#bib-author-cell a::
           text').extract_first()
```

```
        item['publisher'] = response.css('td#bib-publisher-cell
            ::text').extract_first()
        edition_format = response.xpath("//span[@id='
            editionFormatType']/span")
        item['edition_format'] = edition_format.xpath("text()")
            .extract_first() + edition_format.xpath("following-
            sibling::text()").extract_first()
        item['summary'] = response.css('td#bib-summary-cell div
            #summary::text').extract_first()
        item['subjects'] = ""
        for subject in response.css('ul#subject-terms-detailed
            li.subject-term'):
                item['subjects'] += subject.css('a::text').
                    extract_first()+"\n"
        item['genre_form'] = response.css('tr#details-genre td
            ::text').extract_first()
        item['docType'] = response.css('tr#details-doctype td::
            text').extract_first()
        item['note'] = response.css('tr#details-notes td::text'
            ).extract_first()
        item['description'] = response.css('tr#details-
            description td::text').extract_first()
        item['content'] = response.css('tr#details-contents td
            ::text').extract_first()
        item['abstract'] = response.css('div.abstracttxt::text'
            ).extract_first()
        item['onlineName'] = response.css('div#links-all856 p a
            ::text').extract_first()
        item['onlineLink'] = response.css('div#links-all856 p a
            ::attr(title)').extract_first()
        oclcno = response.css('tr#details-oclcno td::text').
            extract_first()
        if oclcno is not None:
                request = scrapy.Request('http://www.worldcat.
                    org/wcpa/servlet/org.oclc.lac.ui.buying.
                    AjaxBuyingLinksServlet?serviceCommand=
                    getBuyingLinks&oclcno='+oclcno, callback=self
                    .parse_seller)
                request.meta['item'] = item
                yield request

    def parse_seller(self, response):
```

```
        item = response.meta['item']
        item['sellerName'] = response.css('td.seller a::text').
            extract_first()
        item['sellerLink'] = response.css('td.seller a::attr(
            href)').extract_first()
        item['sellerPrice'] = response.css('td.price::text').
            extract_first()
        yield item
```

Để phục vụ cho quy mô thử nghiệm thuật toán, 5000 tài liệu tiếng Anh phát hành trong thời điểm từ năm 2014 đến 2017 được thu thập. Chúng bao gồm sách, e-book, bản ghi âm, file audio, video..v..v... có nội dung nằm trong các chủ đề đa dạng khác nhau:

- Đọc hiểu
- Nghe hiểu
- Hội thoại, giao tiếp
- Viết văn
- IETLS
- TOEIC
- Chuyên ngành pháp luật
- Chuyên ngành y tế
- Chuyên ngành toán học

.....

Tài liệu trích xuất được lưu trữ dưới định dạng json:

```
{
        "abstract": "Approximately 500 words and their
            definitions...",
        "author": "Steven J Matthiesen",
        "content": "Success on the TOEFL --",
        "coverLink": "http://coverart.oclc.org/ImageWebSvc/oclc
            /...",
        "description": "1 online resource (vii, 344 pages)",
        "docType": "Internet Resource, Computer File",
        "edition_format": "eBook : Document : English : 6th
            edition",
```

```
        "genre_form": "Electronic books",
        "name": "Barron's essential words for the TOEFL : test
           ...",
        "note": "Previous edition: 2011.",
        "onlineLink": "https://www.overdrive.com/search?q=
           D2F0C4CD-...",
        "onlineName": "OverDrive",
        "publisher": "Hauppauge : Barron's, 2014.",
        "sellerLink": "https://www.amazon.com/Essential-Words-
           TOEFL...",
        "sellerName": "Amazon.com",
        "sellerPrice": "$9.59",
        "subjects": "Test of English as a Foreign Language --
           Study..."
}
```


3.3.2 Xử lý dữ liệu

Dữ liệu thu thập được xử lý bằng thuật toán $tf - idf$ để xác định các từ khoá tiêu biểu đại diện cho nội dung của tài liệu. Văn bản đầu vào của thuật toán $tf - idf$ bao gồm nội dung của các trường "*name*", "*abstract*" và "*subject*".

```
import math
import json
import jsonpickle
from textblob import TextBlob as tb

def tf(word, blob):
    return blob.words.count(word) / len(blob.words)

def n_containing(word, bloblist):
    return sum(1 for blob in bloblist if word in blob.words)

def idf(word, bloblist):
    return math.log(len(bloblist) / (1 + n_containing(word,
       bloblist)))

def tfidf(word, blob, bloblist):
    return tf(word, blob) * idf(word, bloblist)
```

Kết quả thu được là tập từ khoá ứng với mỗi tài liệu như sau:

Tài liệu	**Tags**
Dictionary of medical terms	medicine, terms, dictionary, zymotic, surgery, specialisations, pathology, anatomical, euphemistic, diagnosis
Pronouncing and defining dictionary of music	musicians, music, bio-bibliography, pronouncing, defining, dictionary
Teaching reading vocabulary	reading, comprehension, lecture, subjects, vocabulary, english, curriculum
....	....

Bảng 3.1: Ví dụ kết quả phân tích dữ liệu

Để thu được những tag tiêu biểu nhất, hệ thống áp dụng điều kiện ràng buộc {*mỗi từ khoá phải xuất hiện ít nhất trong 5 tài liệu*} để loại đi các từ khoá thiểu số.

Sau bước xử lý dữ liệu, ta xây dựng được profile của tài liều tiếng Anh như ví dụ sau:

```
'documentProfile': {
        'id': 12,
        'author': 'John S Kwan',
        'name' : 'English spelling',
        'abtract': '...',
        'content': '...',
        ....
        'tags': ['spelling', 'punctuation', 'orthography', '
           rules', 'self-study']
    }
```

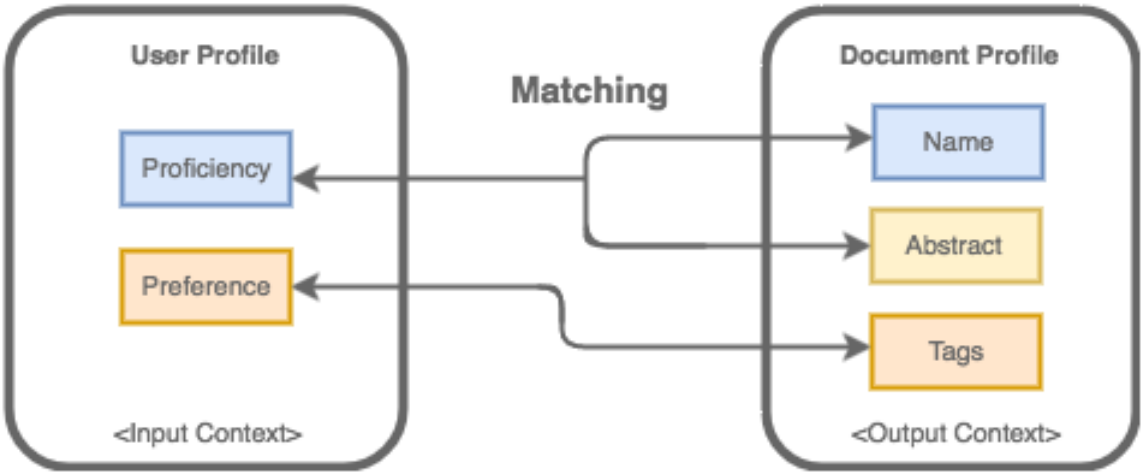
Ngoài ra, một bảng chứa các từ khoá sắp xếp theo thứ tần xuất xuất hiện giảm dần cũng được xây dựng để hỗ trợ người dùng trong bước xác định nguyện vọng như đã đề cập ở bên trên. Chúng sẽ được lưu trữ trong CSDL của Firebase và trả về client mỗi khi có request gửi lên.

```
'tagList': [
        {'name': 'teaching', 'score': 429},
        {'name': 'writing', 'score': 389},
        {'name': 'vocabulary', 'score': 383},
        {'name': 'juvenile', 'score': 351},
        {'name': 'problems', 'score': 313},
        {'name': 'exercises', 'score': 304},
 ...
]
```

3.4 So sánh và đưa ra tư vấn tài liệu học

Sau khi có được *User Profile* người dùng, kết hợp với *Document Profile* đã xử lý trước lấy từ cơ sở tri thức Firebase. Hệ thống sẽ thực hiện so sánh sử dụng thuật toán Context Matching như đã trình bày ở (2.2.1) để đưa ra kết quả tư vấn.

Cụ thể, với bài toán tư vấn tài liệu Tiếng Anh dựa vào trình độ và nguyện vọng của người dùng, *Input Context* và *Output Context* được sử dụng là *User Profile* và *Document Profile*. Các thuộc tính được quan tâm đến tương ứng gồm {*"trình độ"*, *"nguyện vọng"*} và {*"tên tài liệu"*, *"mô tả"*, *"tập từ khoá"*}.



Hình 3.5: So sánh độ tương đồng giữa người dùng và tài liệu

3.4.1 Tính giá trị match e

Do đặc thù bài toán với số lượng thuộc tính của *User Profile* và *Document Profile* không tương đồng, cộng với việc số liên kết được xét đến trong quá trình Context-Matching là ít, việc áp dụng hoàn toàn cách tính của thuật toán được đề xuất sẽ cho ra kết quả có độ chính xác và độ đa dạng tương đối thấp.

Cụ thể là với cách tính giá trị e truyền thống, e chỉ có thể mang một trong hai giá trị 1 - phù hợp, 0 - không phù hợp. Xét một ví dụ có input context là *User profile* {*"Intermediate"*, *"grammar, ielts, video"*} và các output context cần so sánh gồm:

1. ***DP1*** {*"Study English - Intermediate Level. [Series 1], Episode 25"*, *"video interviews with native speakers on topics with relevance to IELTS"*, *"grammar, ielts, video"*}
2. ***DP2*** {*"Funny phonics & silly spelling."*, *"For English learners of intermediate proficiency"*, *"grammar, ielts, video, funny, phonetic, spelling, literature, conversation"*}

3. ***DP3*** {*"6 IELTS Grammar Tests."*, *"For English learners of intermediate proficiency"*, *"grammar, ielts, test"*}

Dễ thấy, nếu áp dụng đúng nguyên mẫu thuật toán.

Ta có $e_{DP1}(1) = e_{DP2}(1) = e_{DP3}(1) = 1$ (do cả 3 tài liệu đều thuộc trình độ Intermediate) và $e_{DP1}(2) = e_{DP2}(2) = 1$, $e_{DP3}(2) = 0$ ($DP1$ và $DP2$ chứa đủ từ khoá nguyện vọng của người dùng, trong khi $DP3$ thiếu mất *"video"*)

Sau khi kết thúc tính toán, ta sẽ có độ khớp của 3 tài liệu trên với người dùng là : $rv_{DP1} = rv_{DP2} > rv_{DP3}$. Kết quả này có 2 nhược điểm:

- Nội dung tài liệu $DP1$ chắc chắn sẽ phù hợp hơn so với $DP2$ do tập từ khoá của $DP1$ trùng khớp hoàn toàn với nguyện vọng người dùng, trong khi của $DP2$ chỉ khớp một phần. Tuy nhiên, giá trị khớp rv của 2 tài liệu này lại là như nhau.
- Tài liệu $DP3$ tương đối phù hợp với nguyện vọng của người dùng, thậm chí có thể còn phù hợp hơn $DP2$ thì lại có rv thấp hơn. Tuy thiếu mất từ khoá *"video"*, nhưng trong tình huống này, nguyện vọng chính của người dùng học là học ngữ pháp IELTS. Dù hình thức học không phải là video đi chăng nữa thì nó vẫn là kết quả phù hợp với nguyện vọng chấp nhận được.

Vì vậy, để giải quyết các nhược điểm trên, đề tài đề xuất một cách tính giá trị e mới phù hợp với điều kiện bài toán hơn như sau:

Khi so sánh proficiency - name/abstract: Giữ nguyên cách tính như cũ. Tìm xem từ khoá trình độ người dùng có trong tên hoặc miêu tả của tài liệu hay không -> $e(1) = 1|0$. Thông thường, giá trị trình độ tổng thể sẽ được dùng để so sánh. Trong trường hợp người dùng muốn tìm kiếm về tài liệu liên quan đến từ vựng, ngữ pháp và đọc hiểu thì sẽ sử dụng giá trị trình độ tương ứng để so sánh.

Khi so sánh preference - tags: Giá trị e [0.00...1.00] được tính bằng trung bình cộng của (Số nguyện vọng khớp/Tổng số nguyện vọng) và (Giá trị phổ biến của các từ khoá khớp/Tổng giá trị phổ biến của tập từ khoá trong tài liệu):

$$e = \frac{1}{2}\left(\frac{|P \cap T|}{|P|} + \frac{\sum\limits_{P \cap T} ts}{\sum\limits_{T} ts}\right) \tag{3.2}$$

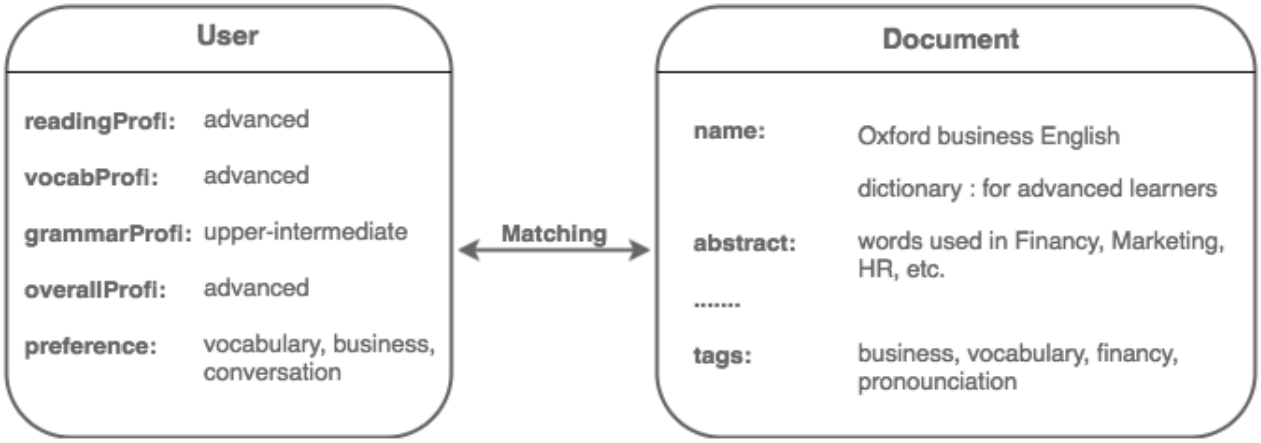
trong đó P = tập nguyện vọng người dùng

T = tập từ khoá của tài liệu

ts = giá trị phổ biến của từ khoá, là số lần xuất hiện của từ khoá đó trên tất cả tài liệu (3.3.2)

3.4.2 Ví dụ với case study

Một người dùng thực hiện bài kiểm tra và đạt kết quả trình độ cao cấp (*advanced*). Anh ta có nhu cầu học từ vựng tiếng Anh văn phòng. Hệ thống sẽ thử matching User Profile này với cuốn sách "*Oxford business English dictionary : for learners of English*".



Hình 3.6: Input context & Output context

Quá trình Context-Matching sẽ diễn ra như sau:

Bước 1: Đánh giá match và xác định e

No	Input	Output	*e*
1	advanced	Oxford business English dictionary : for advanced learners words used in Financy, Marketing, HR, etc.	1
2	vocabulary, business, conversation	business, vocabulary, financy, pronounciation	0.6788

Bảng 3.2: Xác định e

Ta có $e_1 = 1$,

Giá trị phổ biến của "vocabulary", "business", "conversation", "financy", "pronounciation" lần lượt là 383, 288, 158, 52, 248.

$$=> e_2 = \frac{1}{2}\left(\frac{2}{3} + \frac{383+288}{383+288+52+248}\right) = 0.6788$$

Bước 2: Xác định w

Giá trị w được chuyên gia quy định sẵn trong hệ thống, trong ví dụ này, chúng có gía trị $w_1 = 0.37$ & $w_2 = 0.78$

Bước 3: Tính giá trị av

$av_1 = e_1 * w_1 = 0.37$
$av_2 = e_2 * w_2 = 0.5294$

Bước 4: Tổng sav

$sav = av_1 + av_2 = 0.8994$

Bước 5: Tính giá trị match lớn nhất mpv

$mpv = w_1 + w_2 = 1.15$

Bước 6: Tính độ phù hợp rv

$$rv = \frac{sav}{mpv} = \frac{0.8994}{1.15} = 0.782$$

Bước 7: So sánh với giá trị ngưỡng t

Giá trị ngưỡng t được chuyên gia quy định sẵn trong hệ thống, trong ví dụ này, nó có giá trị $t = 0.65$.

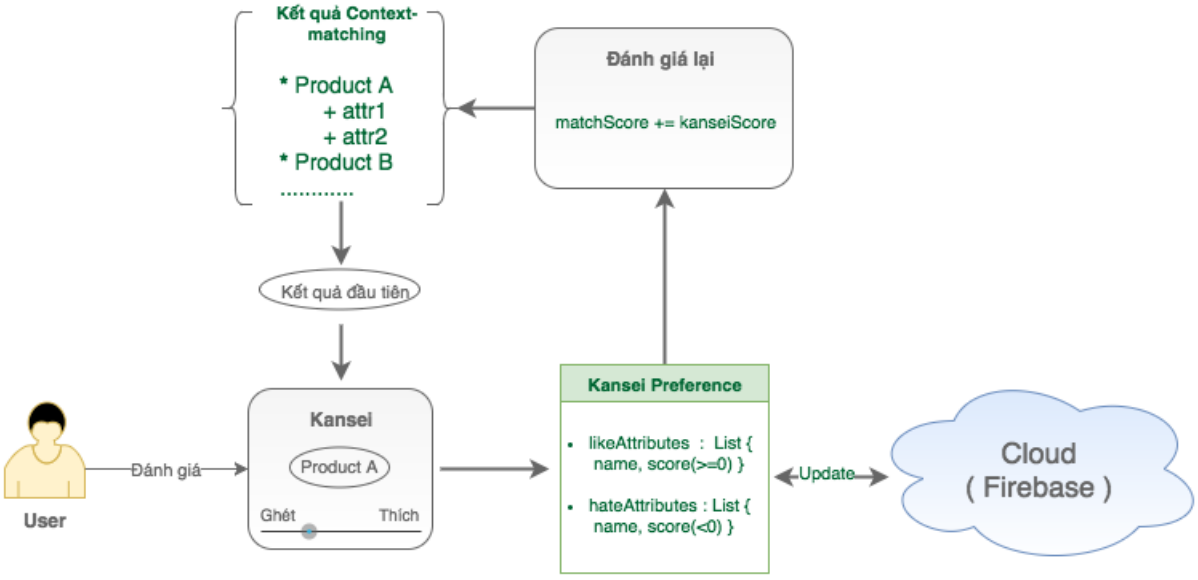
Do $rv = 0.782 > t$ => Kết quả là tài liệu được xét có phù hợp với nguyện vọng và trình độ người dùng.

3.5 Áp dụng Kansei Engineering để cải thiện kết quả tư vấn

Bước Context-matching trả về cho người dùng kết quả tư vấn đã phù hợp với *trình độ* và *nguyện vọng* của họ. Tuy nhiên, ta dễ dàng thấy được với một đối tượng người dùng bất kỳ (cùng *trình độ* và cùng *nguyện vọng*) sẽ luôn cho ra tập kết quả y hệt nhau. Trong khi đó, mục tiêu cuối cùng mà đề tài này hướng đến không phải là các kết quả tư vấn chung cho một đối tượng người dùng mà phải là kết quả tư vấn phù hợp với *cá nhân* từng người dùng khác nhau.

Cụ thể, nó phải phù hợp với thói quen, sở thích cá nhân của người dùng - những yếu tố mà có thể hoặc không thể thể diễn đạt qua từ khoá nguyện vọng. (Vd: Cả 2 người dùng đều muốn học từ vựng tiếng Anh, nhưng người dùng A thích học qua video có ví dụ minh hoạ dễ nhớ, người dùng B thích đọc sách, có giải nghĩa cụ thể chi tiết..v..v...)

Bởi vậy, bước đánh giá Kansei được thực hiện để cá nhân hoá kết quả tư vấn cho người dùng.



Hình 3.7: Mô hình đánh giá Kansei

Mô hình thuật toán thể hiện như hình vẽ bên trên là một chu trình tuần hoàn, bắt đầu từ tập kết quả nhận được từ bước Context-matching. Kết quả đầu tiên - tài liệu có độ phù hợp cao nhất với người dùng, được lấy ra để người dùng đánh giá theo cảm nhận của họ. Người dùng sẽ chọn mức độ ưa thích của mình đối với đối tượng bằng cách điều chỉnh thanh đo *Ghét - Thích*. Tuỳ theo kết quả đánh giá mà ta sẽ có được các thuộc tính người dùng thích (*likeAttributes*), hoặc các thuộc tính mà người dùng ghét (*hateAttributes*) và điểm số tương ứng của chúng. Sử dụng các thuộc tính này để đánh giá lại độ ưu tiên của các kết quả còn lại, rồi lại tiếp tục chọn ra kết quả đầu tiên tiếp theo, và bắt đầu một chu kì mới tương tự.

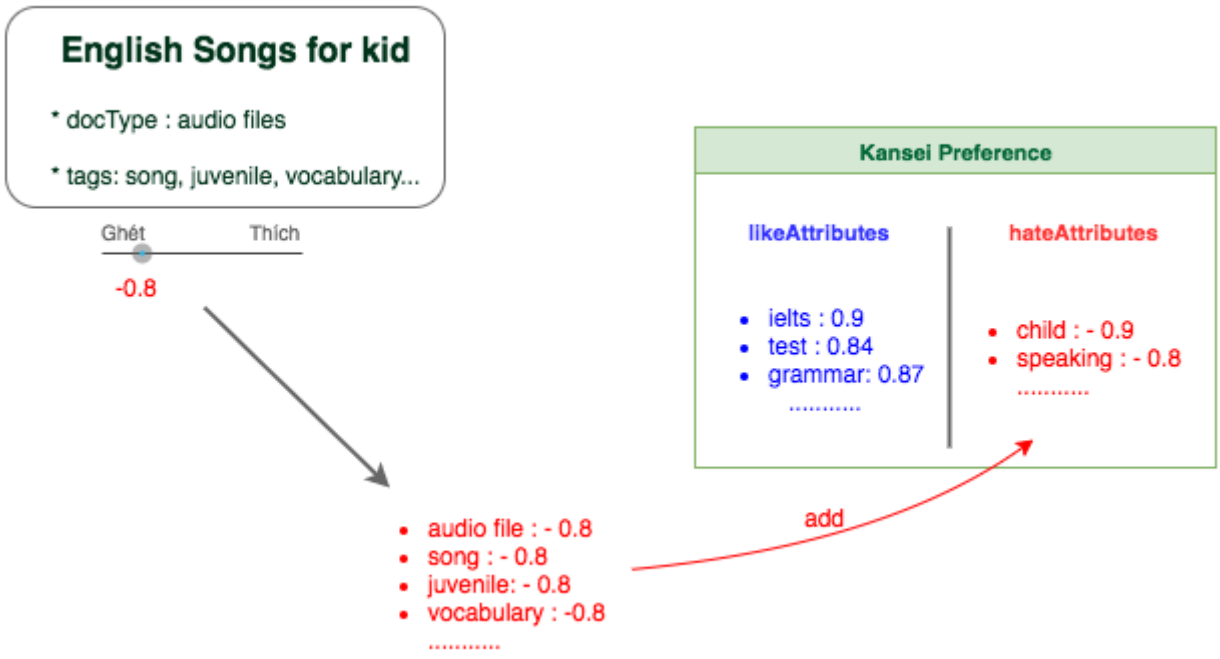
Algorithm 1 Cải thiện kết quả tư vấn với Kansei Engineering

procedure KANSEI($contextMatchingResult$)
    $firstItem \leftarrow contextMatchingResults[0]$ Lấy item đầu tiên
    $contextMatchingResults \leftarrow contextMatchingResults[1, +]$
    Người dùng đánh giá thang cảm xúc Ghét-Thích
    if $score >= 0$ **then**
        $likeAttributes \leftarrow addAll(firstItem.attributes, score)$
        $kanseiPreference \leftarrow add(likeAttributes)$
    else
        $hateAttributes \leftarrow addAll(firstItem.attributes, score)$
        $kanseiPreference \leftarrow add(hateAttributes)$
    $contextMatchingResult \leftarrow reevaluate(contextMatchingResults, kanseiPreference)$
    KANSEI($contextMatchingResults$).

3.5.1 Đánh giá Kansei

Như đã trình bày ở phần Tổng quan Kansei Engineering (2.3.1), Kansei là mô hình thể hiện nguyện vọng bên trong của con người biểu hiện qua cảm xúc. Bằng cách đánh giá Kansei, ta có thể xác định được các yếu tố ưu tiên của sản phẩm tuỳ theo cá nhân từng người dùng khác nhau.

Đối tượng cần đánh giá, tài liệu học tiếng anh có độ phù hợp với *trình độ* và *nguyện vọng* của người dùng cao nhất hiện tại, được hiển thị đầy đủ thông tin cho người dùng. Người dùng sau đó sẽ dựa vào những thông tin được cung cấp và đánh giá *độ thoả mãn* của họ về đối tượng. Việc đánh giá được thực hiện bằng việc điều chỉnh thanh gía trị đến khoảng mà người dùng cảm thấy phù hợp nhất. Giá trị mà người dùng có thể chọn tương ứng với khoảng $[-1, 1]$.



Hình 3.8: Ví dụ đánh giá

Các tiêu chí để đánh giá tài liệu bao gồm có *Độ thú vị*, *Độ khó*, *Độ thoả mãn* và *Giá cả phù hợp*. *Độ thú vị* đánh giá xem tài liệu được tư vấn có thuộc miền chủ đề mà người dùng thích hay không. *Độ khó* đánh giá xem trình độ người dùng có phù hợp để học tài liệu. *Độ thoả mãn* đánh giá xem mức độ hài lòng của người dùng về tài liệu đến đâu. Và cuối cùng, *Giá cả phù hợp* đánh giá giá thành của tài liệu có phù hợp với túi tiền của người dùng hay không.

Ứng với 4 tiêu chí trên, thuộc tính *tag* và *price* của tài liệu được trích xuất sau khi đánh gía để xây dựng Kansei Preference của người dùng. 2 thuộc tính trên được lựa chọn để đánh giá vì qua đánh giá đó là những thuộc tính có ảnh hưởng lớn nhất đến cảm nhận của người dùng. Tuỳ theo giá trị mà người dùng đánh giá, hệ thống sẽ có được các thuộc tính người dùng thích hoặc thuộc tính người dùng ghét.

3.5.2 Kansei Preference Model

Kansei Preference là tập các tiêu chí đánh giá ảnh hưởng tích cực hoặc tiêu cực đến cảm nhận của người dùng về tài liệu mà họ được tư vấn.

Xét trong phạm vi đề tài này, 4 tiêu chí *Độ thú vị*, *Độ khó*, *Độ thoả mãn* và *Giá cả phù hợp* được chọn ra là 4 Kansei Word tiêu biểu đối với bài toán tư vấn tài liệu học tiếng Anh, do vậy, mô hình KanseiPreference có cấu trúc như sau.

```
public class KanseiPreferences {
    private KanseiItem interesting;
    private KanseiItem satisfy;
    private KanseiItem understandable;
    private KanseiItem affordable;
}
```

Mỗi tiêu chí lại ứng với một tập các thuộc tính của tài liệu tiêu biểu cho tiêu chí đó.

```
public class KanseiItem {
    private List<KanseiAttribute> attributes;
    private double weight;
}
```

Do quá trình đánh giá Kansei đánh giá từng sản phẩm một, các thuộc tính có trong sản phẩm này mà người dùng thích đồng thời có thể tồn tại trong một sản phẩm khác mà người dùng đánh giá là ghét. Bởi vậy, để xác định được chính xác thuộc tính ảnh hưởng như thế nào đến người dùng. Lớp thuộc tính được xây dựng có cấu trúc như sau :

```
public class KanseiAttribute {
    private String name;

    private double goodScore = 0;
```

```
    private int totalGoodRated = 0;
    private double badScore = 0;
    private int totalBadRated = 0;

    ...
}
```

Mỗi khi có thuộc tính được đánh giá gửi đến, hệ thống sẽ kiểm tra xem thuộc tính đó có tồn tại trong Kansei Preference hay không. Nếu không tồn tại, thuộc tính đó sẽ được thêm vào, nếu nó tồn tại, giá trị của thuộc tính đó sẽ được thay đổi.

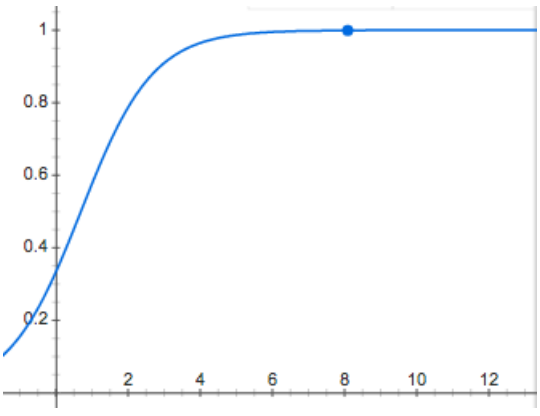
Nếu như người dùng đánh giá gần "*Thích*", biến đếm *totalGoodRated* được cộng thêm 1, giá trị *goodScore* được cộng dồn, ngược lại, *totalBadRated* cũng sẽ được cộng thêm 1 và *badScore* được cộng dồn nếu người dùng đánh giá gần "*Ghét*" hơn.

Tuy nhiên, khi người dùng đánh giá một tiêu chí nào đó của sản phẩm, toàn bộ các thuộc tính tiêu biểu cho tiêu chí đó đều được đánh giá cùng mức điểm mà người dùng cho. Trong khi đó, có thể trong chủ ý người dùng chỉ đang đánh giá một trong các thuộc tính đó thôi. Những trường hợp như vậy sẽ làm cho giá trị thuộc tính đo được không còn chính xác, ảnh hưởng đến kết quả đánh giá. Để tránh những trường hợp trên, giá trị kansei của thuộc tính được đánh giá theo hàm nghịch đảo mũ e.

Giá trị kansei của thuộc tính được tính theo công thức:

$$r_s = score \times \frac{1}{1 + 2e^{-s}} \tag{3.3}$$

trong đó r_s : giá trị kansei của thuộc tính (thích/ghét)
$score$: tổng giá trị người dùng đánh giá ($goodScore/badScore$)
s : biến đếm số lần người dùng đánh giá ($totalGoodRated/totalBadRated$)



Hình 3.9: Giá trị kansei của x thay đổi theo số lần được đánh giá

Thêm vào đó, các giá trị đánh giá tốt và đánh giá xấu của thuộc tính được lưu riêng biệt với nhau. Khi cần xác định giá trị Kansei của thuộc tính, 2 giá trị này sẽ được so sánh với nhau, nếu thuộc tính có r_s thích $> r_s$ ghét, thuộc tính đó là thuộc tính *likeAttribute* và *hateAttribute* trong trường hợp ngược lại và giá trị r_s tương ứng sẽ là giá trị Kansei của thuộc tính, sử dụng để đánh giá lại Context-matching trong bước tiếp theo.

$$r_x = s_{good} * r_{s_{good}} > s_{bad} * r_{s_{bad}} : r_{s_{good}} \ ? \ r_{s_{bad}} \tag{3.4}$$

3.5.3 Đánh giá lại kết quả Context-matching

Sau khi xác định Kansei Preference, hệ thống tiến hành đánh giá lại toàn bộ kết quả context-matching chưa được xét đến.

Kansei Preference bao gồm 4 tiêu chí đánh giá v_i, v_u, v_s, v_a, với mỗi tiêu chí đánh giá lại có một tập các thuộc tính $x_1|x_2|...|x_n$ tiêu biểu cho tiêu chí đó, vậy nên ta có:

$$v_i = \sum_n r_{x_i} \tag{3.5}$$

r_{x_i} : giá trị kansei của thuộc tính i đặc trưng cho tiêu chí v

Điểm đánh giá Kansei của tài liệu được tính bằng:

$$kanseiScore = \frac{v_i * w_i + v_u * w_u + v_s * w_s + v_a * w_a}{mpv + w_i + w_u + w_s + w_a} \tag{3.6}$$

với v là giá trị kansei của các tiêu chí
w là trọng số của tiêu chí, $w = \dfrac{totalRateScore}{totalRateTime}$

Độ phù hợp mới của tài liệu sẽ bằng:

$$matchScore = contextMatchingScore + kanseiScore \tag{3.7}$$

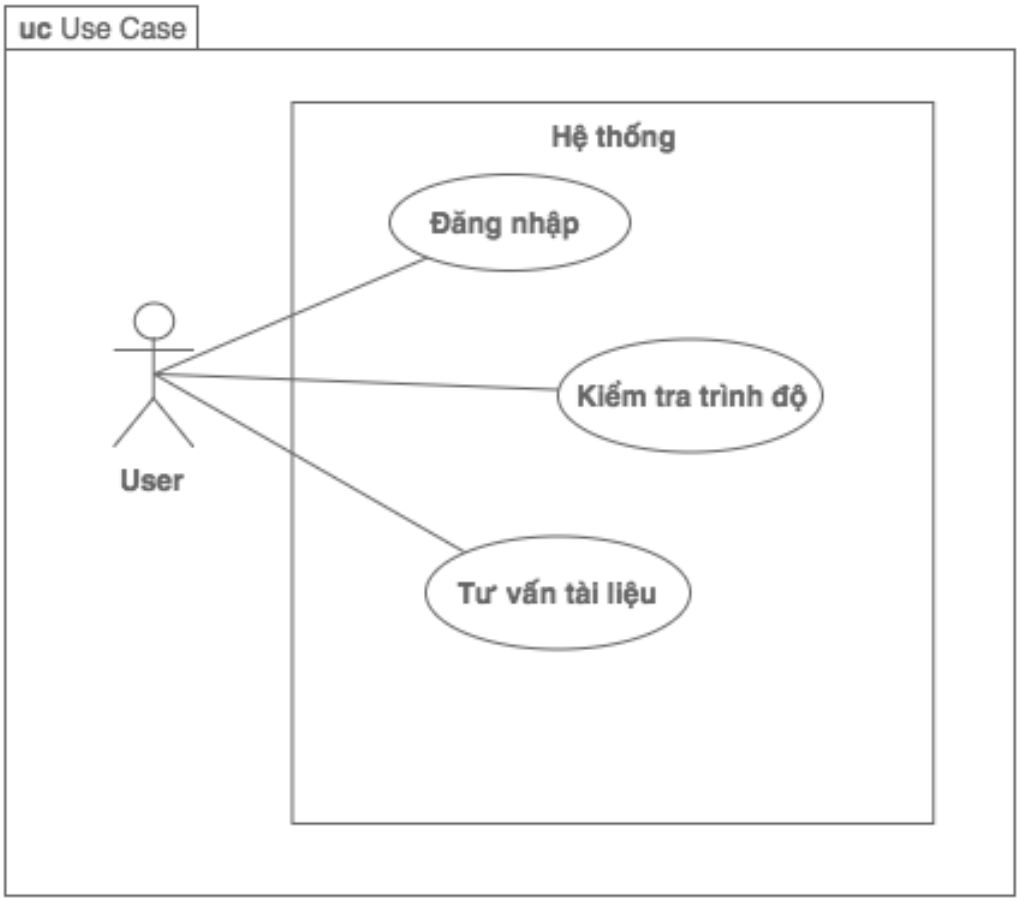
Với tài liệu có chứa thuộc tính mà người dùng ghét, giá trị *matchScore* sẽ giảm đi, ngược lại với tài liệu mà hợp với người dùng , giá trị *matchScore* sẽ tăng lên. Tập kết quả sẽ được sắp xếp lại độ phù hợp mới và lấy ra kết quả phù hợp nhất đầu tiên đưa cho người dùng, bắt đầu một chu kỳ tiếp theo.

Chương 4

Cài đặt hệ thống

4.1 Use case sử dụng

Dựa vào chức năng mà hệ thống sẽ xây dựng sẽ cung cấp, người dùng sẽ có các ca sử dụng như sau:



Hình 4.1: Use case tổng quan của người dùng

UC#001	**Đăng nhập**
Miêu tả	Người dùng kết nối tài khoản Facebook để đăng nhập vào hệ thống. Nếu tài khoản đăng nhập lần đầu tiên, thông tin người dùng sẽ được khởi tạo trên Firebase
Đối tượng	• Người dùng
Diễn biến chính	1. Người dùng tap vào nút đăng nhập bằng Facebook 2. Facebook trả về token đăng nhập, hệ thống gửi token lên server Firebase để kiểm tra 3. Firebase trả về thông tin người dùng 4. Hệ thống chuyển đến màn hình chính
Diễn biến ngoại lệ	3a. Hệ thống không tìm thấy token, xác định người dùng đăng nhập lần đầu tiên, khởi tạo thông tin người dùng và gửi lên Firebase. 3b. Hệ thống thông báo lỗi kết nối Internet.

Bảng 4.2: Đặc tả ca sử dụng: Đăng nhập

UC#002	**Kiểm tra trình độ**
Miêu tả	Người dùng trả lời các câu hỏi hệ thống đưa ra để xác định trình độ hiện tại của mình
Đối tượng	• Người dùng
Diễn biến chính	1. Người dùng chưa thực hiện kiểm tra trình độ sẽ được hệ thống yêu cầu thực hiện bài kiểm tra. 2. Người dùng tap vào chọn khoảng thời gian từ khi bắt đầu học tiếng Anh. 3. Hệ thống tính toán trình độ ban đầu đưa ra câu hỏi với trình độ tương ứng cho người dùng. 4. Người dùng xem câu hỏi và tap vào đáp án mà nghĩ là đúng. 5. Hệ thống nhận câu trả lời và đánh giá lại trình độ của người dùng. 6. Hệ thống kiểm tra xem lĩnh vực hiện tại đã đánh giá được chưa, chuyển sang đánh giá lĩnh vực tiếp theo. 7. Hệ thống kiểm tra xem đã đánh giá toàn bộ các lĩnh vực chưa, hiển thị kết quả đánh giá trình độ từng lĩnh vực và tổng thể cho người dùng
Diễn biến ngoại lệ	6a. Chưa đủ để đánh giá trình độ, quay lại bước 3 7a. Chưa đánh giá hết, chuyển sang đánh giá lĩnh vực tiếp theo, quay lại bước 3

Bảng 4.4: Đặc tả ca sử dụng: Kiểm tra trình độ

UC#003	**Tư vấn tài liệu**
Miêu tả	Người dùng miêu tả nguyện vọng của họ, các tài liệu sẽ được tư vấn dựa trên trình độ và nguyện vọng đó, người dùng sau đó sẽ đánh giá các kết quả tư vấn và hệ thống sẽ phân tích đưa ra các kết quả tiếp theo phù hợp với cá nhân người dùng
Đối tượng	• Người dùng
Diễn biến chính	1. Người dùng tap vào nút bắt đầu tư vấn. 2. Hệ thống chuyển sang màn hình hỏi nguyện vọng người dùng. 3. Người dùng nhập nguyện vọng về tài liệu học tiếng Anh vào ô nhập liệu. 4. Hệ thống tính toán và lần lượt đưa ra từng kết quả một dựa trên trình độ và nguyện vọng của người dùng. 5. Người dùng đánh giá kết quả tư vấn, di chuyển thanh đo ở từng tiêu chí và tap vào nút đánh giá sau khi hoàn thành 6. Hệ thống nhận kết quả đánh giá và cập nhập các tiêu chí ưu tiên đối với người dùng. 7. Kiểm tra xem còn kết quả tư vấn chưa hiển thị hay không, quay trở lại bước 3
Diễn biến ngoại lệ	4a. Hệ thống không tìm thấy tài liệu phù hợp, thông báo ra màn hình "Không tìm được kết quả phù hợp". 7a. Không còn kết quả chưa hiển thị, hệ thống chuyển về màn hình chính

Bảng 4.6: Đặc tả ca sử dụng: Tư vấn tài liệu

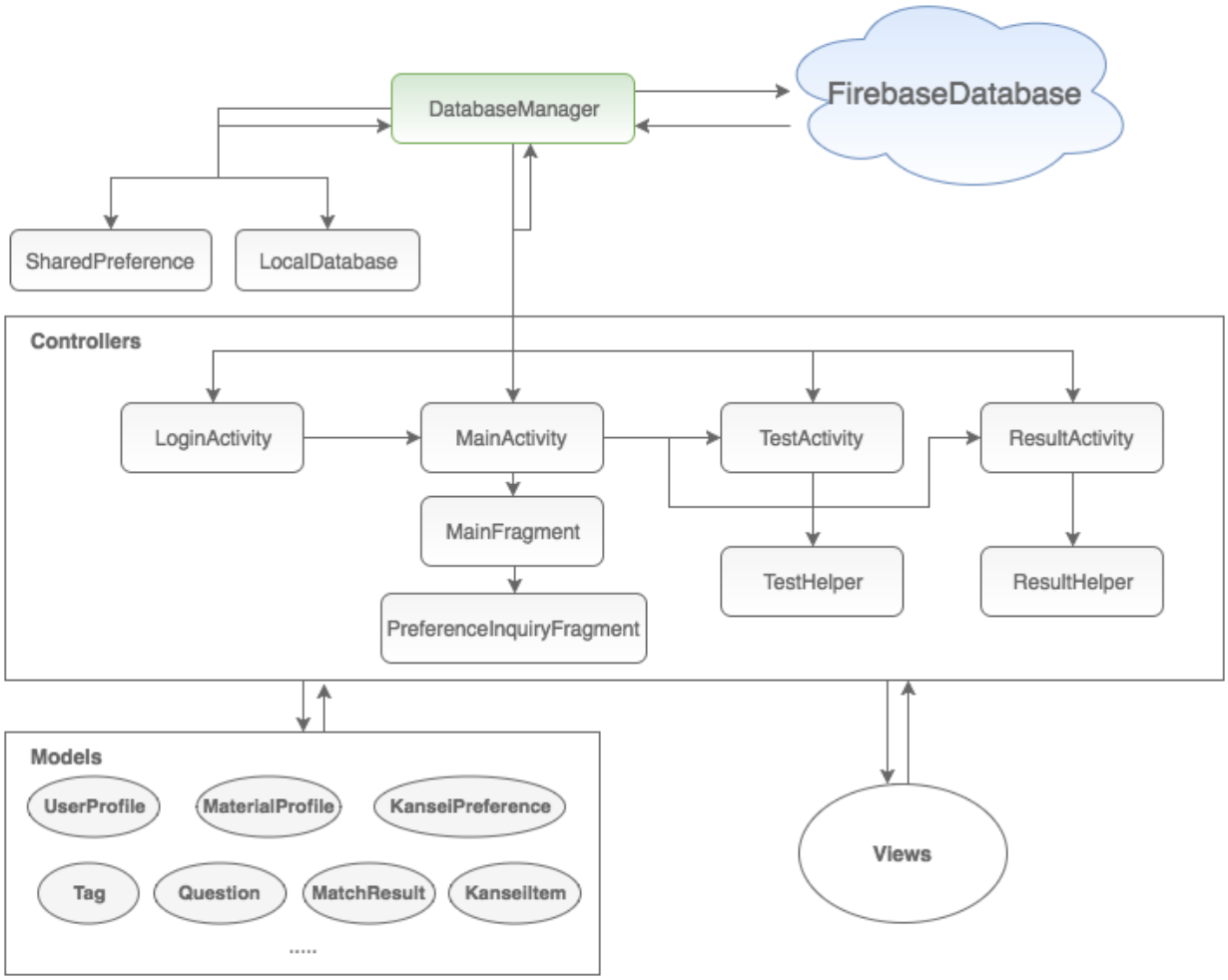
4.2 Cài đặt hệ thống

4.2.1 Môi trường cài đặt hệ thống

- Hệ điều hành: OS X El Capitan.
- Môi trường phát triển: Android Studio
- Cơ sở dữ liệu: Firebase.
- Framework: Android SDK.
- Ngôn ngữ: Java.

4.2.2 Kiến trúc hệ thống cài đặt

Kiến trúc hệ thống cài đặt được mô tả như hình vẽ dưới đây:



Hình 4.2: Mô hình kiến trúc hệ thống

Các thành phần chính của hệ thống:

- **FirebaseDatabase:** Lớp Wrapper của Firebase SDK trên Android, có nhiệm vụ kết nối với cơ sở dữ liệu thời gian thực Firebase Realtime Database và thực hiện các câu lệnh truy vấn đến Database cũng như trả kết quả từ Firebase về client Android.

- **DatabaseManager:** Lớp quản lý truy xuất dữ liệu sử dụng trong hệ thống, đóng vai trò cầu nối giữa Controller và Firebase, đặc tả các câu lệnh truy xuất dữ liệu, gửi đến Server và xử lý kết quả trả về từ Json về các Model đã định nghĩa. Ngoài ra, lớp này còn đóng vai trò giao tiếp với Local Database và Shared Preference để gửi và nhận dữ liệu lưu trữ trên thiết bị.

- **LocalDatabase:** Cơ sở dữ liệu địa phương, copy và lưu trữ dữ liệu ở Firebase sau lần truy xuất lấy dữ liệu đầu tiên. Ở các lần khởi động hệ thống tiếp theo, một truy vấn sẽ được gửi đến Firebase xem bộ dữ liệu có thay đổi gì không, cập nhập bộ dữ liệu địa phương nếu có phát hiện thay đổi, đảm bảo đồng bộ giữa client và server.

- **SharedPreference:** Lưu trữ thông tin đăng nhập của người dùng sau lần đăng nhập đầu tiên. Người dùng sẽ được tự động đăng nhập vào hệ thống vào các lần khởi động ứng dụng tiếp theo.

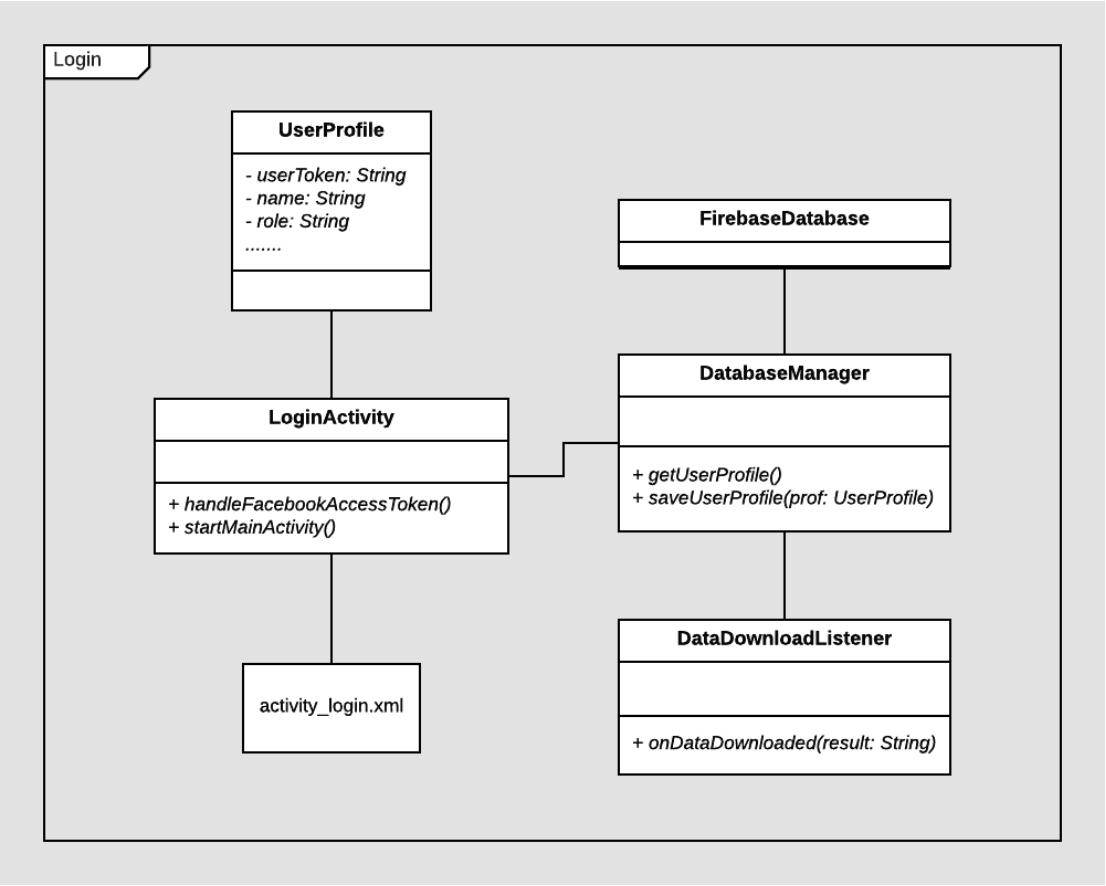
- **Controllers:** Bao gồm các lớp thực hiện điều khiển và thực hiện các tác vụ chính trong hệ thống. Controller bắt các tương tác từ người dùng với giao diện đặc tả ở View và xác định yêu cầu, gọi các phương thức để xử lý chúng sau đó hiển thị kết quả lên giao diện người dùng. Mỗi Activity (và Fragment) đi kèm với chúng đóng vai trò thực hiện một tác vụ nhất định trong hệ thống, nằm giữa giao tiếp với Model và View. Ngoài ra, các lớp Helpers đóng gói các phương thức xử lý logic phức tạp, cài đặt thuật toán thực hiện trong hệ thống.

- **Models:** Bao gồm các lớp chứa dữ liệu được tổ chức có cấu trúc, mỗi lớp đặc trưng cho một đối tượng cụ thể mà Controller hoặc các thành phần khác sẽ gọi đến để sử dụng trong xử lý bài toán. Dữ liệu lưu trữ trong cơ sở dữ liệu đều được lưu dưới dạng các đối tượng của lớp Model.

- **Views:** Bao gồm các file .xml đặc tả cấu trúc, cách bố trí và sự xuất hiện của giao diện ứng dụng, trong đó có layout, drawable, animation, value về màu sắc, style, font chữ ..v..v...Mỗi Activity và Fragment sẽ có một layout tương ứng với chúng.

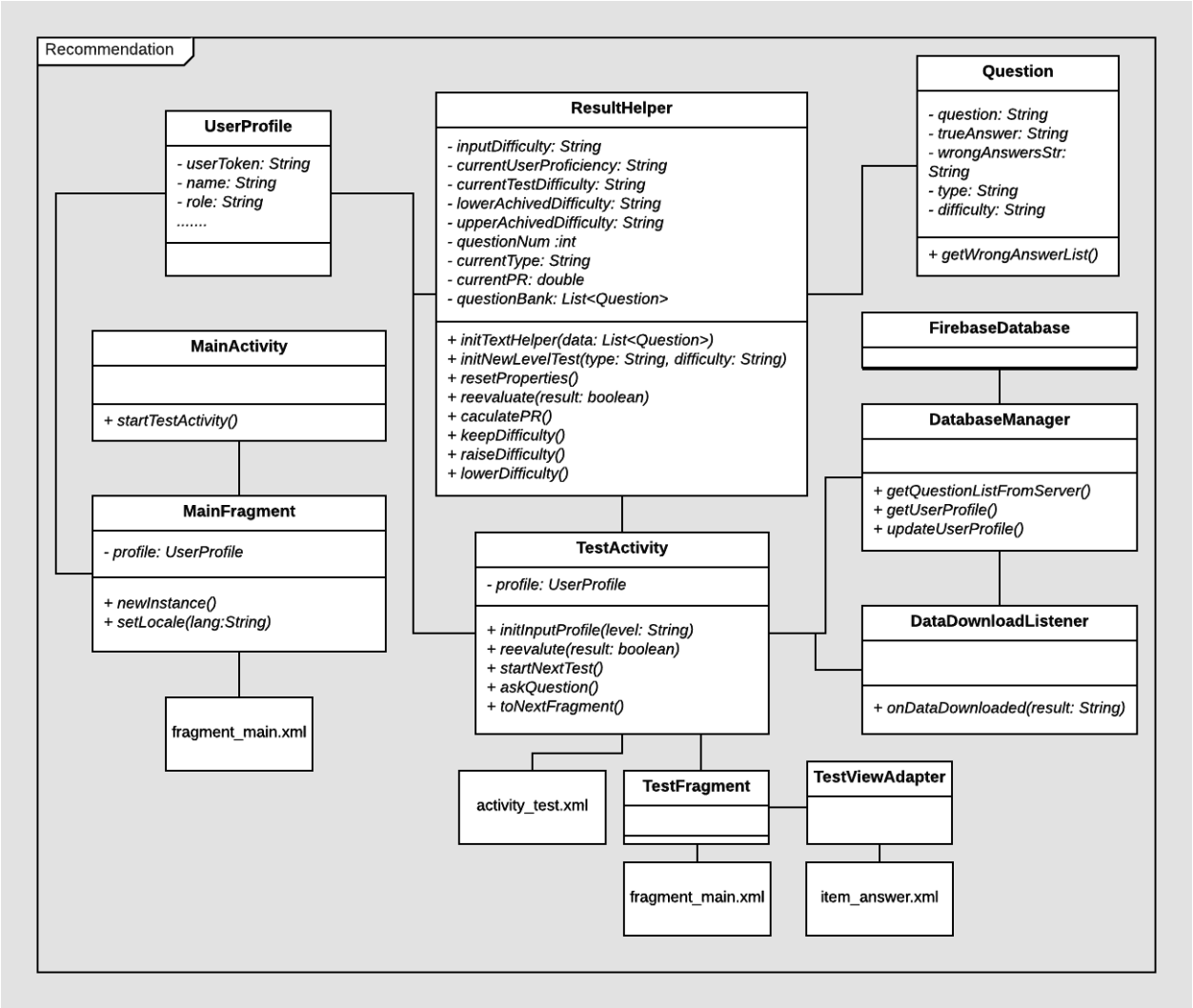
4.2.3 Biểu đồ lớp theo ca sử dụng

- **Đăng nhập:**



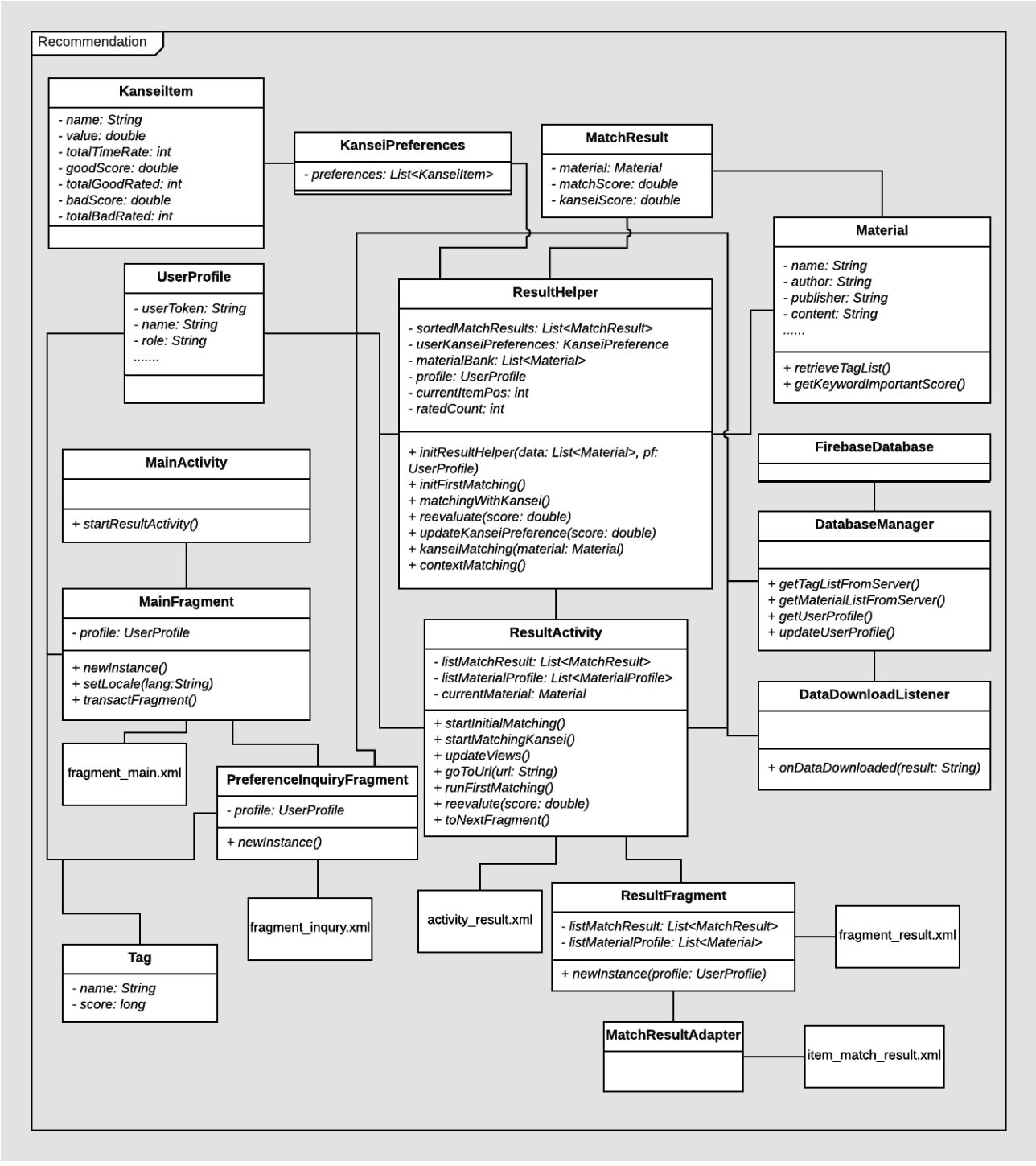
Hình 4.3: Biểu đồ lớp theo ca sử dụng: đăng nhập

- **Kiểm tra trình độ:**



Hình 4.4: Biểu đồ lớp theo ca sử dụng: kiểm tra trình độ

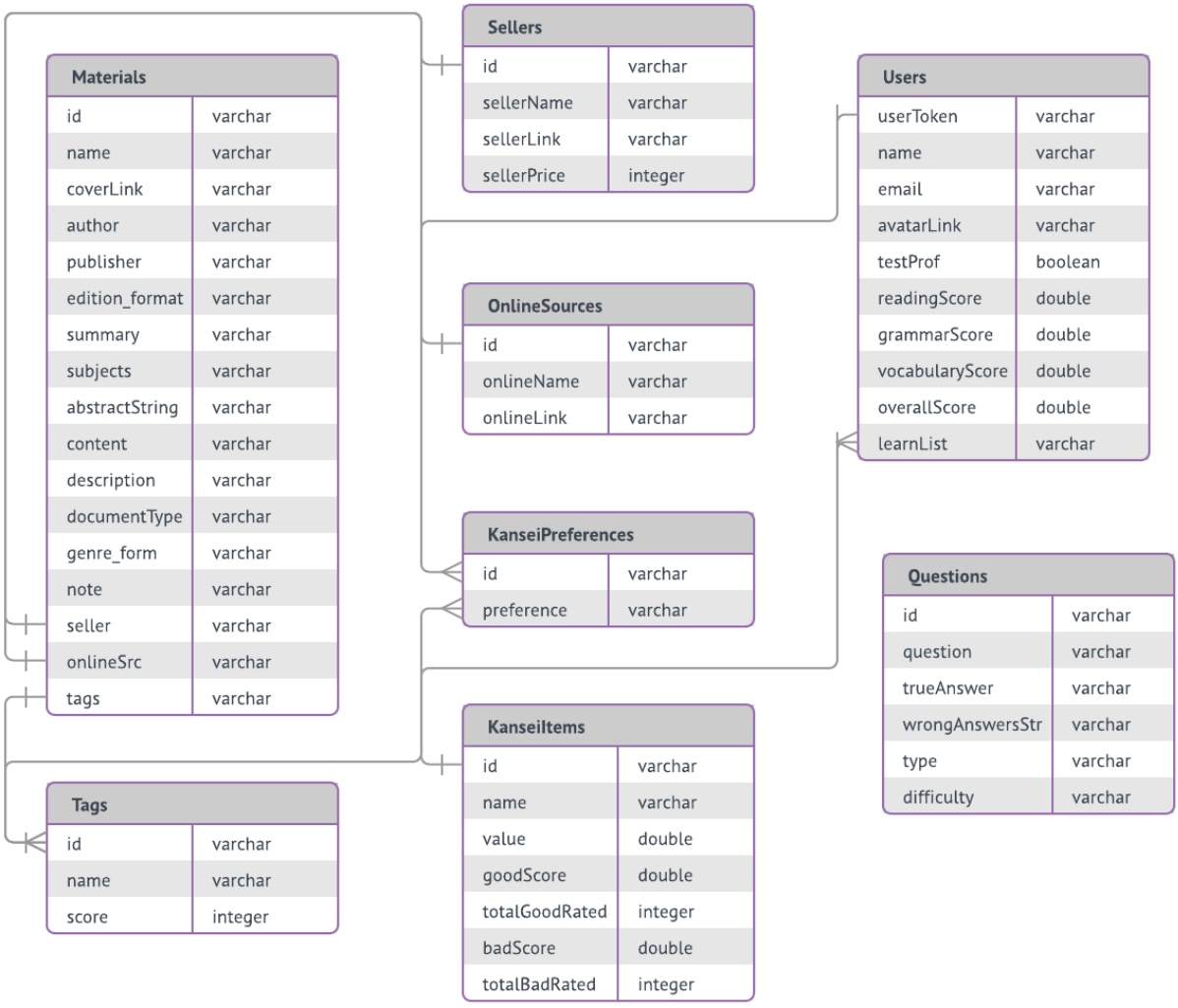
- **Tư vấn tài liệu:**



Hình 4.5: Biểu đồ lớp theo ca sử dụng: tư vấn tài liệu

4.2.4 Mô hình cơ sở dữ liệu

Dữ liệu lưu trữ trên Firebase được lưu trữ dưới dạng NoSQL, khác với các mô hình RDBMS thông thường. Trong NoSQL, dữ liệu được lưu trữ dưới dạng JSON, mô hình dưới dạng mà hệ thống có thể sử dụng trực tiếp, loại bỏ các ràng buộc liên kết giữa bảng với bảng. Các thuộc tính trong một đối tượng được trải đều ra và không khuyến khích phân tầng, truy xuất dựa trên các cặp "key" và "value". Nhờ vậy, NoSQL cho ra hiệu năng cao hơn khi cần truy xuất tập dữ liệu lớn so với các mô hình RDBMS thông thường.



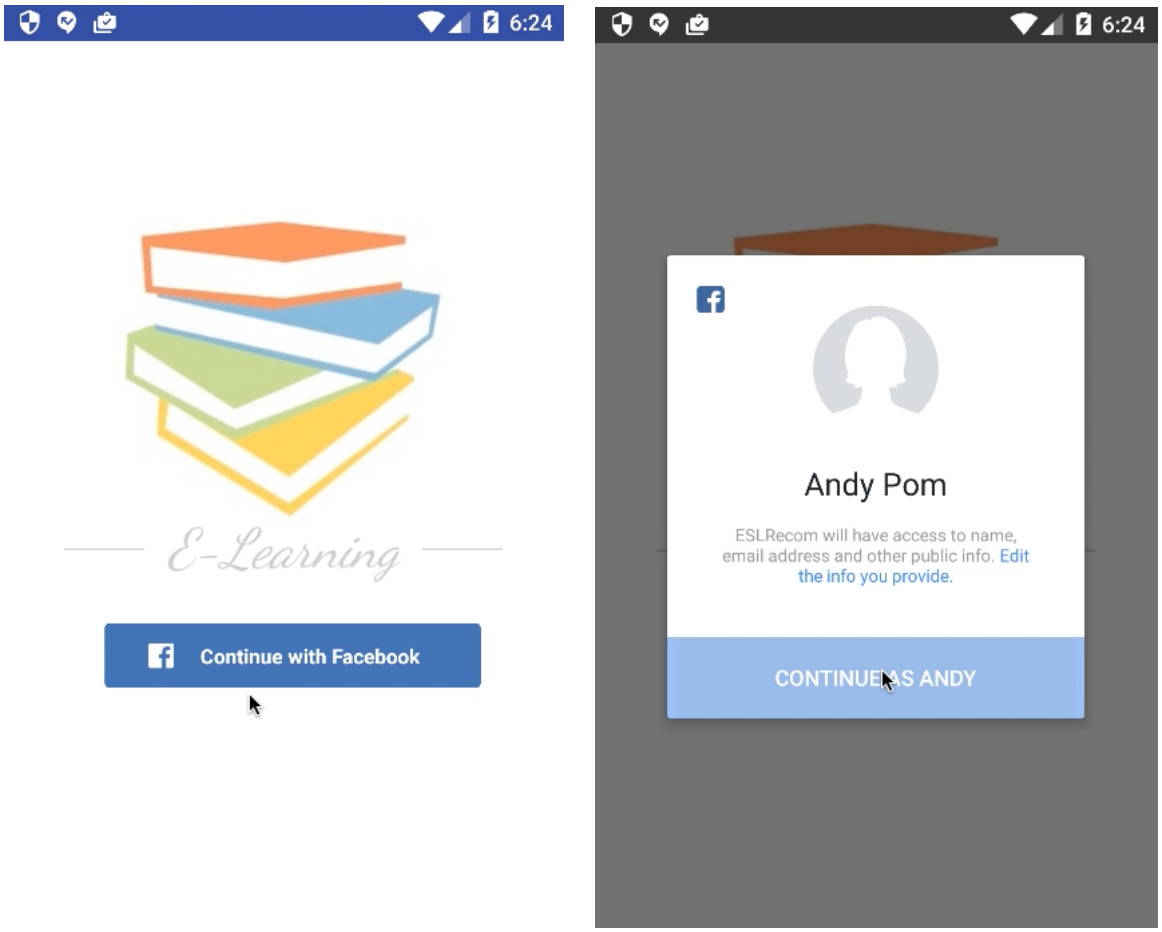
Hình 4.6: Mô hình cơ sở dữ liệu

4.3 Kết quả cài đặt

Sử dụng các kiến thức và phân tích trong đề tài, sau khi cài đặt thử nghiệm, ứng dụng xây dựng được có kết quả như sau:

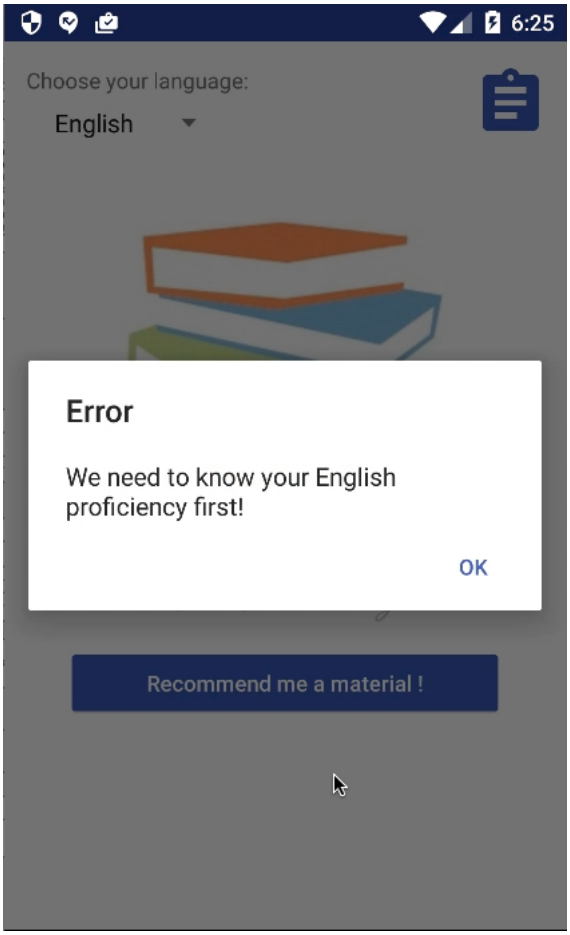
Màn hình đăng nhập:

Màn hình đầu tiên được hiển thị khi bắt đầu sử dụng ứng dụng, người dùng cần đăng nhập (hoặc đăng ký nếu chưa là thành viên) qua tài khoản Facebook trước khi bắt đầu sử dụng. Sau khi đã đăng nhập (hoặc đăng ký) thành công, hệ thống sẽ tự động phát hiện người dùng đã đăng nhập và hiển thị màn hình chính của ứng dụng trong các lần khởi động tiếp theo. Thông tin người dùng trên Firebase sẽ được tải về (hoặc cập nhập) sau khi đăng nhập.



Hình 4.7: Giao diện màn hình đăng nhập

Với người dùng sử dụng ứng dụng lần đầu tiên, hệ thống sẽ yêu cầu người dùng phải thực hiện kiểm tra trình độ trước khi sử dụng chức năng tư vấn.

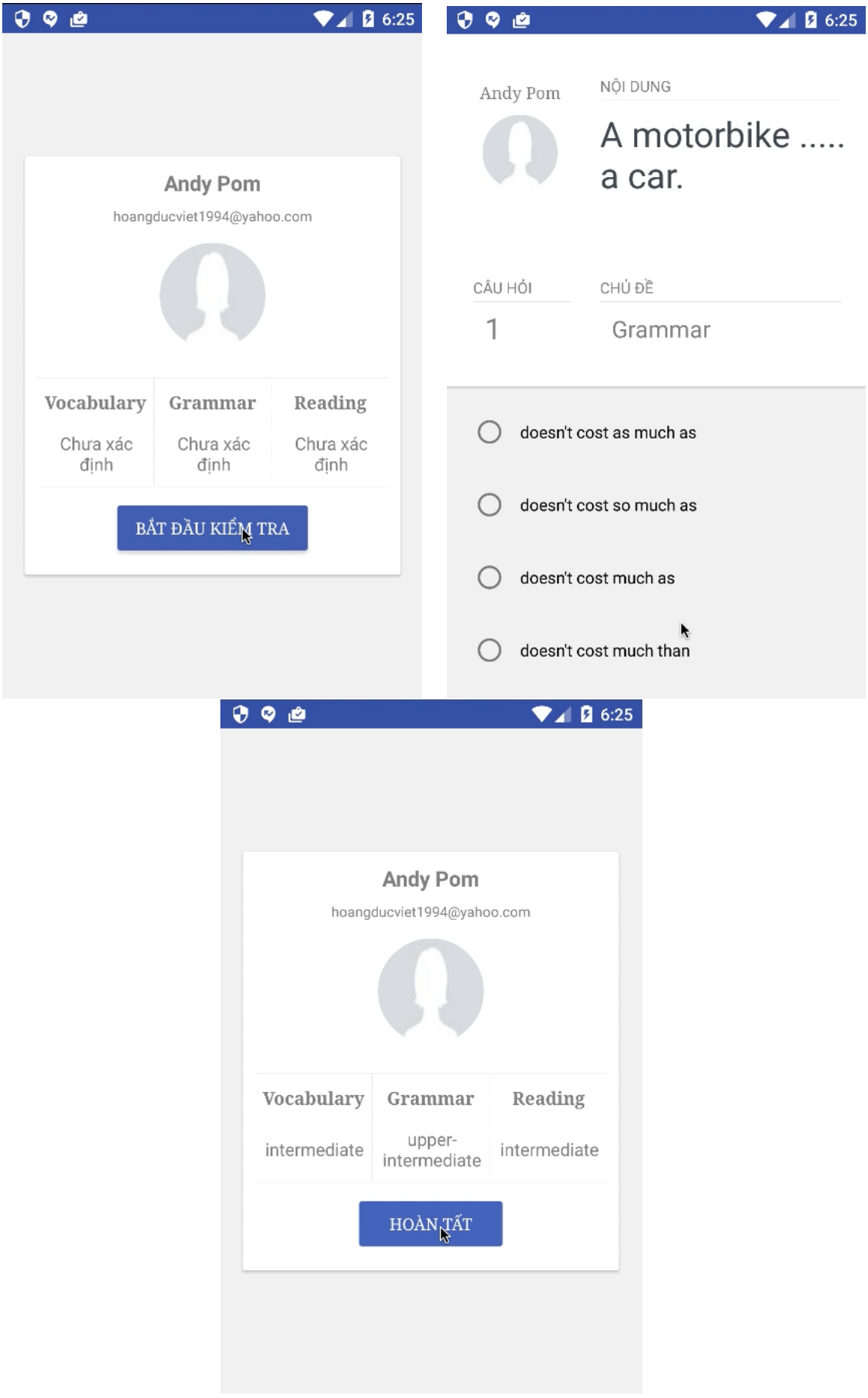


Hình 4.8: Người dùng được yêu cầu làm bài kiểm tra trình độ trong lần sử dụng đầu tiên

Màn hình kiểm tra trình độ:

Sau khi ấn OK. Ứng dụng sẽ chuyển sang màn hình thực hiện kiểm tra trình độ. Người dùng sẽ trải qua lần lượt 3 bài kiểm tra về kiến thức Từ vựng, Ngữ pháp và Đọc hiểu tiếng Anh với các câu hỏi có độ khó thay đổi tuỳ biến vào kết quả trả lời của người dùng. Mỗi bài kiểm tra dài khoảng từ 5 15 câu hỏi tuỳ thuộc vào kết quả trả lời của người dùng.

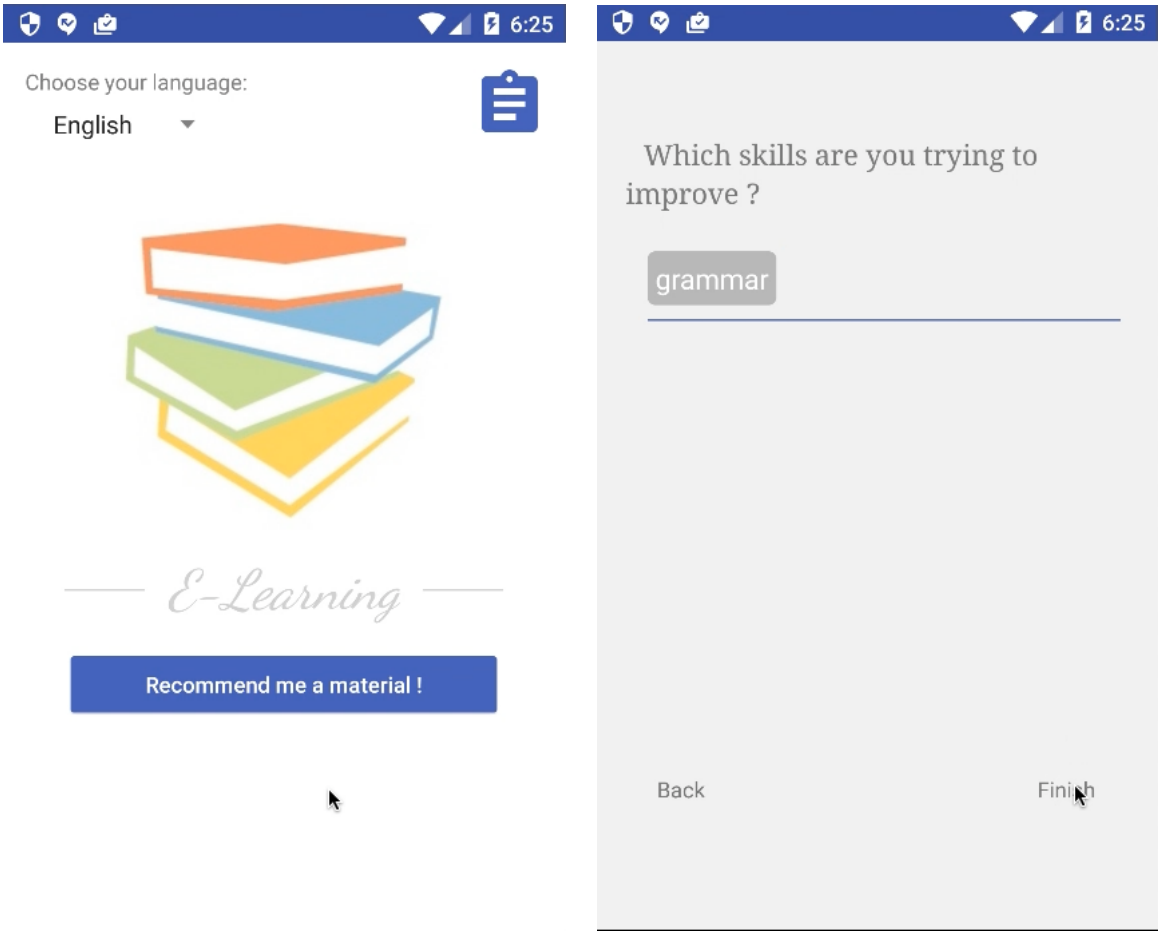
Sau khi đã hoàn tất bài kiểm tra, trình độ của người dùng về từng lĩnh vực được hiển thị lên trên màn hình, người dùng chọn Hoàn Tất để quay trở lại màn hình chính.



Hình 4.9: Giao diện kiểm tra trình độ

Màn hình nhập nguyện vọng học tiếng Anh:

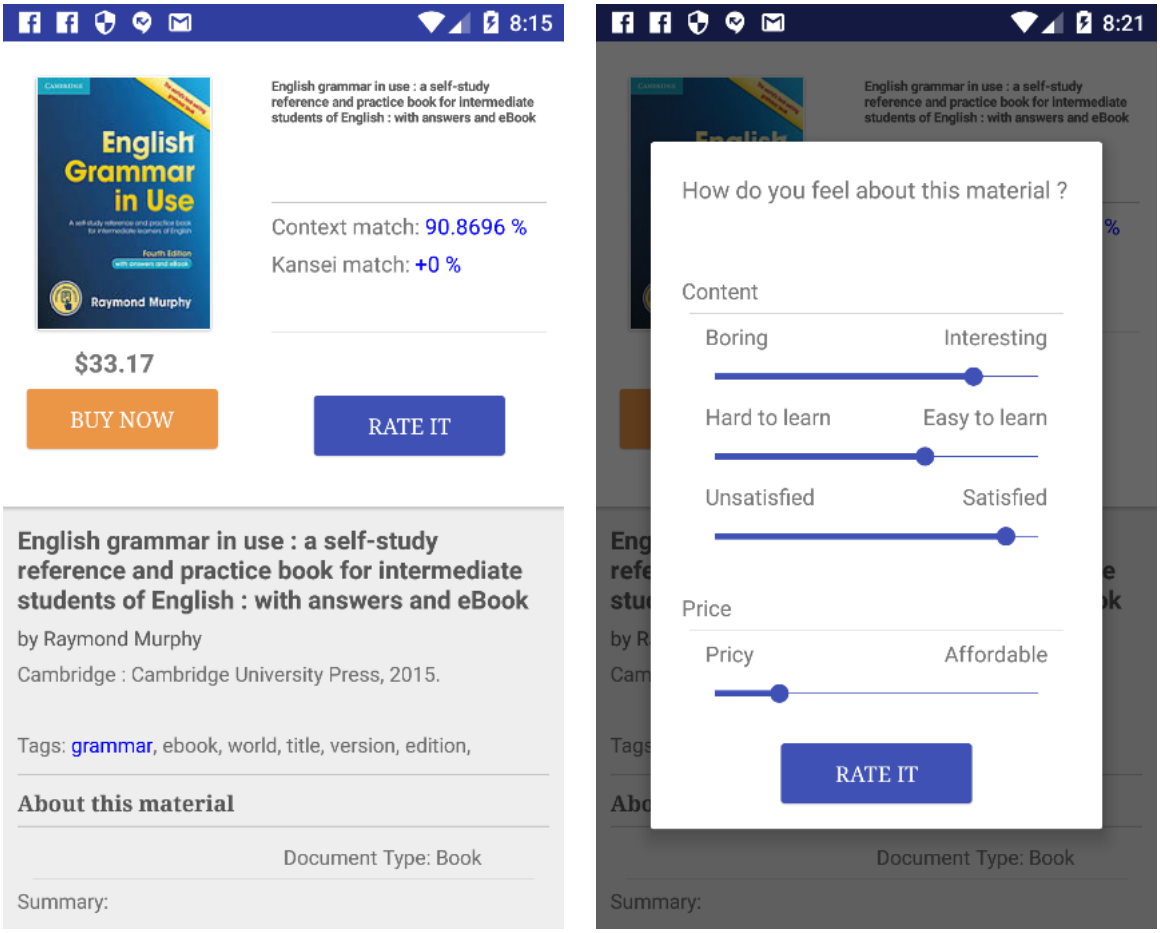
Trong màn hình chính, khi ấn chọn tư vấn, ứng dụng sẽ chuyển sang giao diện nhập nguyện vọng. Người dùng sẽ nhập nguyện vọng về một kĩ năng hoặc lĩnh vực nào đó mà họ cần tư vấn vào ô nhập liệu, các kết quả gợi ý sẽ xuất hiện để gợi ý cho người dùng. Sau khi nhập xong, người dùng ấn "Finish" để bắt đầu thực hiện tư vấn.



Hình 4.10: Giao diện nhập nguyện vọng học của người dùng

Màn hình kết quả và đánh giá kết quả tư vấn:

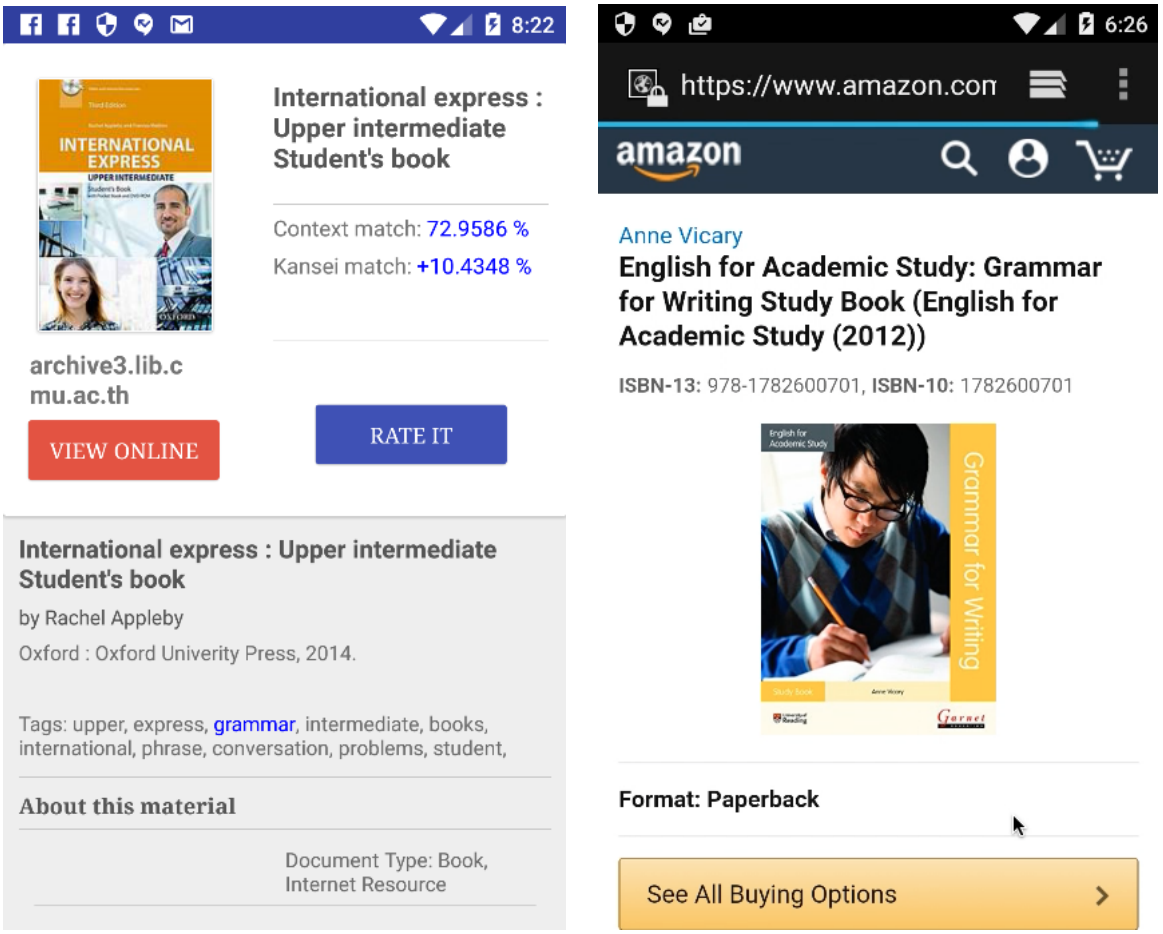
Sau khi kết thúc tính toán, ứng dụng sẽ hiển thị lần lượt các kết quả tư vấn theo giao diện hình bên trái phía dưới. Trang bìa tài liệu, tên tác giả, thể loại, giá cả là các thông tin quan trọng, được bố cục nổi bật lên trên đầu, phía dưới là các thông tin cụ thể khác về tài liệu. Để chuyển sang tài liệu tiếp theo, người dùng ấn vào nút "Rate it" để đánh giá tài liệu. Các tiêu chí đánh giá bao gồm *Độ thú vị*, *Độ khó*, *Độ thoả mãn* và *Giá cả phù hợp* được hệ thống đưa ra nhằm tìm ra các ưu tiên của người dùng.



Hình 4.11: Giao diện kết quả tư vấn và đánh giá Kansei

Sau đánh giá, tiêu chí ưa thích của người dùng được cập nhập. Các kết quả tư vấn chưa hiển thị được đánh giá lại một lần nữa, sau đó kết quả tư vấn mới phù hợp nhất được lấy ra và hiển thị lên màn hình.

Ngoài ra, trên giao diện còn có nút "Buy now" (hoặc "View Online" nếu tài liệu có link xem online), người dùng có thể ấn vào đó và chuyển hướng đến thư viện hoặc trang mua hàng để đặt mua tài liệu.



Hình 4.12: Kết quả sau khi đánh giá Kansei và giao diện mua hàng Amazon

Chương 5

Kết luận và hướng phát triển

5.1 Các kết quả đã đạt được

Đồ án đã đạt được các kết quả sau đây:

- Nắm được kiến thức về thuật toán Context-Matching và áp dụng được vào trong tư vấn tài liệu.
- Nắm được kiến thức về Kansei Engineering và áp dụng được vào trong cải thiện kết quả tư vấn sao cho phù hợp với từng cá nhân người dùng.
- Mô hình bài thi tương tác tuỳ biến cho ra kết quả đo trình độ người dùng với độ chính xác ngang bằng bài kiểm tra thông thường dù chỉ cần sử dụng số lượng câu hỏi ít hơn.
- Xây dựng được ứng dụng thử nghiệm cho ra kết quả tư vấn tài liệu tương đối chính xác.
- Ứng dụng xây dựng được là một công cụ hữu ích hỗ trợ cho việc học E-learning. Có tiềm năng phát triển thành module hỗ trợ việc giảng dạy cho các trung tâm/trường học dạy tiếng Anh qua mạng.

5.2 Những hạn chế còn tồn đọng

Tuy nhiên, đồ án không thể tránh khỏi các thiết sót còn cần giải quyết :

- Logic thuật toán được cài đặt và xử lý trực tiếp trên thiết bị, dẫn đến hiệu năng chưa cao. Các bước xử lý tốn tương đối nhiều thời gian làm ứng dụng chạy chưa được mượt.
- Khối lượng tài liệu tư vấn còn ít, chưa dẫn đến các kết quả tư vấn của mỗi người dùng khác nhau vẫn còn tương đối giống nhau.
- Câu hỏi kiểm tra trình độ tiếng Anh chưa thực sự phản ánh được đúng trình độ người dùng do kiến thức hiểu biết về việc giảng dạy tiếng Anh còn hạn chế.

5.3 Định hướng phát triển trong tương lai

Đề tài "Xây dựng hệ thống tư vấn tài liệu học tiếng Anh E-Learning" trong khuôn khổ đồ án này chỉ dừng lại ở mức độ tìm hiểu, nghiên cứu và bước đầu xây dựng ứng dụng thử nghiệm thuật toán. Để có thể đưa vào triển khai trong môi trường thực tế, ứng dụng cần được tiếp tục hoàn thiện và phát triển. Sau đây là một số hướng đi đề xuất để phát triển ứng dụng trong tương lai:

- Xây dựng hệ thống trên nền tảng Web, chuyển các tác vụ xử lý nặng lên backend của Server.
- Bổ sung các câu hỏi kiểm tra phù hợp hơn, thêm bài kiểm tra các trình độ Nói, Nghe Hiểu, Viết Văn.
- Bố sung tài liệu tiếng Anh có chọn lọc, thêm các tài liệu dưới dạng khoá học online, bài giảng ..v..v...
- Bổ sung các chức năng quản trị hệ thống cho người quản trị như quản lý người dùng, thêm/sửa/xoá tài liệu, bài kiểm tra, log hành vi của người dùng ..v..v...

Tài liệu tham khảo

[1] kornferry: A guide to aspects ability. https://www.trytalentq.com/how-to-complete/aspects-ability/. last visited: March 2017. vii, 10

[2] stevenloria: Finding important words in text using tf-idf. http://stevenloria.com/finding-important-words-in-a-document-using-tf-idf/. last visited: April 2017. 12

[3] Ishihara K. Ishihara, S. and M. Nagamachi. Hierarchical kansei analysis of beer can using neural network. In *Proceedings of Human Factors in Organizational Design and Management - VI*, pages 421–425, 1998. 8

[4] G.J. Suci Osgood, C.E. and P.H. Tannenbaum. The measurement of meaning. 7

[5] Hai V. Pham Philip Moore. Personalization and rule strategies in data intensive intelligent context-aware systems. *The Knowledge Engineering Review, Vol. 00:0,c 2011, Cambridge University Press*, pages 1–24, 2011. 4

[6] Hai V. Pham Philip Moore. Intelligent context with decision support under uncertainty. *2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, 2012. 4

[7] Theodore W Frick R. Edwin Welch. Computerized adaptive testing in instructional settings. *Educational Technology Research and Development*, pages 47–62, 1993. 16

[8] Simon Schütte. Engineering emotional values in product design-kansei engineering in development. *Linköpings Universitet*, pages 55–64, 2005. 6

PHỤ LỤC 1: Quản lý dữ liệu trên Firebase

Giới thiệu về mạng neural nhân tạo

PHỤ LỤC 2 : Tích hợp đăng nhập qua Facebook vào hệ thống