**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**---🙢🕮🙠---**

****

**Dự án**

**Ứng dụng trắc nghiệm QuizzHappy**

**Giảng viên hướng dẫn: Mai Cường Thọ**

**Lớp: 63. CNTT-2**

**Người thực hiện: Nguyễn Thiết Duy Đan**

**Khánh Hòa – năm 2024**

MỤC LỤC

[MỤC LỤC 1](#_Toc168241391)

[DANH MỤC HÌNH VẼ 3](#_Toc168241392)

[Chương 1. GIỚI THIỆU VỀ ỨNG DỤNG TRẮC NGHIỆM 5](#_Toc168241393)

[1.1 Giới thiệu ứng dụng trắc nghiệm: 5](#_Toc168241394)

[1.2 Khảo sát: 5](#_Toc168241395)

[1.3 Mục tiêu: 6](#_Toc168241396)

[1.4 Tổng quan về QuizzHappy: 6](#_Toc168241397)

[Chương 2. THIẾT KẾ HỆ THỐNG 7](#_Toc168241398)

[2.1 Sơ đồ logic: 7](#_Toc168241399)

[2.2 Giao diện: 7](#_Toc168241400)

[2.2.1 Giao diện Intro: 8](#_Toc168241401)

[2.2.2 Giao diện đăng ký: 8](#_Toc168241402)

[2.2.3 Giao diện đăng nhập: 8](#_Toc168241403)

[2.2.4 Giao diện Home: 9](#_Toc168241404)

[2.2.5 Giao diện User: 9](#_Toc168241405)

[2.2.6 Giao diện Quiz: 10](#_Toc168241406)

[2.2.7 Giao diện Result: 10](#_Toc168241407)

[Chương 3. CÀI ĐẶT HỆ THỐNG 12](#_Toc168241408)

[3.1 Màn hình Intro: 12](#_Toc168241409)

[3.2 Chức năng đăng nhập: 13](#_Toc168241410)

[3.3 Chức năng đăng ký: 15](#_Toc168241411)

[3.4 Màn hình chính: 17](#_Toc168241412)

[3.5 Chức năng thông tin người dùng: 21](#_Toc168241413)

[3.6 Chức năng trả lời câu hỏi: 22](#_Toc168241414)

[3.7 Chức năng hiện kết quả: 25](#_Toc168241415)

[Chương 4. ĐÁNH GIÁ KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 30](#_Toc168241416)

[4.1 Công nghệ sử dụng 30](#_Toc168241417)

[4.2 Kết quả đạt được 30](#_Toc168241418)

[4.3 Hướng phát triển 30](#_Toc168241419)

[Tài liệu tham khảo 31](#_Toc168241420)

DANH MỤC HÌNH VẼ

[Hình 2.1 Sơ đồ logic của ứng dụng 7](#_Toc168241421)

[Hình 2.2 Màn hình Intro 8](#_Toc168241422)

[Hình 2.3 Mà hình Đăng Ký 8](#_Toc168241423)

[Hình 2.4 Màn hình Đăng Nhập 9](#_Toc168241424)

[Hình 2.5 Màn hình Home 9](#_Toc168241425)

[Hình 2.6 Màn hình User 10](#_Toc168241426)

[Hình 2.7 Màn hình Quiz 10](#_Toc168241427)

[Hình 2.8 Màn hình Result 11](#_Toc168241428)

[Hình 3.1 Màn hình Intro 12](#_Toc168241429)

[Hình 3.2 Sử dụng thư viện Handler 12](#_Toc168241430)

[Hình 3.3 Tổ chức giao diện đăng nhập 13](#_Toc168241431)

[Hình 3.4 Xử lý Fragment 14](#_Toc168241432)

[Hình 3.5 Cách lấy giá trị TextView 14](#_Toc168241433)

[Hình 3.6 Phương thức kiểm tra tên người dùng và mật khẩu 14](#_Toc168241434)

[Hình 3.7 Phương thức hiển thị mật khẩu 15](#_Toc168241435)

[Hình 3.8 Tổ chức giao diện đăng ký 16](#_Toc168241436)

[Hình 3.9 Kiểm tra không được để trống 16](#_Toc168241437)

[Hình 3.10 Kiểm tra đăng ký 17](#_Toc168241438)

[Hình 3.11 Phương thức nhập dữ liệu vào database 17](#_Toc168241439)

[Hình 3.12 Tổ chức giao diện Home 18](#_Toc168241440)

[Hình 3.13 Thiết kế kiểu hình round 18](#_Toc168241441)

[Hình 3.14 Gọi kiểu hình ra 18](#_Toc168241442)

[Hình 3.15 Cài đặt Adapter 19](#_Toc168241443)

[Hình 3.16 Thiết lập RecyclerView 19](#_Toc168241444)

[Hình 3.17 Tạo Bundle lấy userID 19](#_Toc168241445)

[Hình 3.18 Lấy userID từ Bundle 20](#_Toc168241446)

[Hình 3.19 Truy vấn để lấy username 20](#_Toc168241447)

[Hình 3.20 Thêm thư viện Glide 20](#_Toc168241448)

[Hình 3.21 Sử dụng Glide để load hình ảnh 21](#_Toc168241449)

[Hình 3.22 Tổ chức giao diện User 21](#_Toc168241450)

[Hình 3.23 Tổ chức giao diện Quiz 22](#_Toc168241451)

[Hình 3.24 Lấy tổng số câu và câu hiện tại 23](#_Toc168241452)

[Hình 3.25 Phương thức đếm thời gian 23](#_Toc168241453)

[Hình 3.26 Truy vấn lấy dữ liệu bỏ vào mảng 24](#_Toc168241454)

[Hình 3.27 Hiển thị màu sắc sai và đúng 25](#_Toc168241455)

[Hình 3.28 Tổ chức giao diện Result 26](#_Toc168241456)

[Hình 3.29 Sử dụng Glide để dùng ảnh Gif 26](#_Toc168241457)

[Hình 3.30 Thiết kế vòng xoay circularProgressBar 27](#_Toc168241458)

[Hình 3.31 Thiết kế nền của circularProgressBar 28](#_Toc168241459)

[Hình 3.32 Gọi các drawable vừa thiết kế ra 28](#_Toc168241460)

[Hình 3.33 Làm animation cho circularProgressBar 29](#_Toc168241461)

1. GIỚI THIỆU VỀ ỨNG DỤNG TRẮC NGHIỆM
   1. Giới thiệu ứng dụng trắc nghiệm:

Ứng dụng trắc nghiệm là một ứng dụng hỗ trợ giúp người dùng thực hiện các câu trắc nghiệm để ôn lại các kiến thức mà người dùng mong muốn. Ứng dụng được xây dựng giúp cho người dùng có thể tự học dù bất cứ nơi đâu với đa dạng các chủ đề sẽ giúp cho người không cảm thấy nhàm chán.

Ngoài ra, ứng dụng còn có thể cập nhật các câu hỏi để cho người dùng học hỏi thêm nhiều điều hơn và giúp cho họ có thêm cải thiện thêm kiến thức.

* 1. Khảo sát:

**Các Ứng Dụng Quiz (Kahoot!, Quizlet, Socrative)**

Các ứng dụng quiz cung cấp công cụ tạo và tham gia các câu hỏi trắc nghiệm, giúp người dùng ôn tập và kiểm tra kiến thức.

**Kahoot!**

* **Mô tả:** Nền tảng học tập dựa trên trò chơi.
* **Tính năng:** Tạo câu hỏi trắc nghiệm, tích hợp Google Classroom, báo cáo kết quả.
* **Ưu điểm:** Dễ sử dụng, thúc đẩy tương tác.
* **Nhược điểm:** Phiên bản miễn phí hạn chế tính năng.

**Quizlet**

* **Mô tả:** Ứng dụng học tập với thẻ ghi nhớ và quiz.
* **Tính năng:** Tạo thẻ ghi nhớ, chế độ học đa dạng, cộng đồng chia sẻ tài nguyên.
* **Ưu điểm:** Linh hoạt, hỗ trợ nhiều chế độ học.
* **Nhược điểm:** Một số tính năng yêu cầu trả phí.

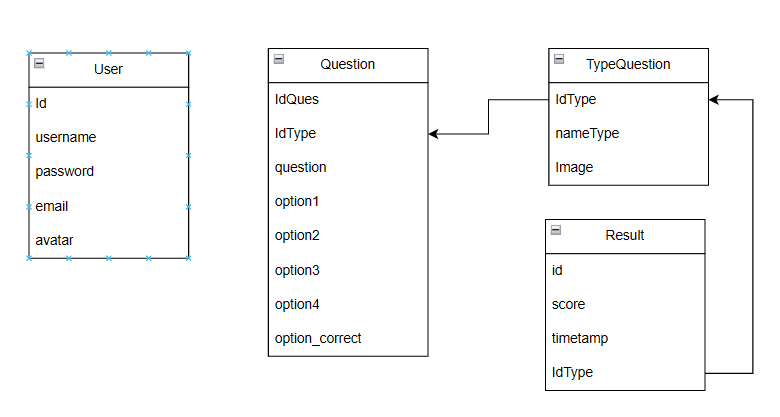
**Socrative**

* **Mô tả:** Ứng dụng kiểm tra và đánh giá học tập trực tuyến.
* **Tính năng:** Tạo câu hỏi trắc nghiệm, câu hỏi mở, báo cáo kết quả tức thời.
* **Ưu điểm:** Tương tác thời gian thực, báo cáo chi tiết.
* **Nhược điểm:** Cần kết nối internet ổn định.
  1. Mục tiêu:
* Mục tiêu tạo ra ứng dụng này là nhằm mục đích hỗ trợ cho người dùng học tập.
* Cải thiện được kiến thức cho người dùng.
* Giúp người dùng có một môi trường kiến thức mới.
  1. Tổng quan về QuizzHappy:

QuizHappy là một ứng dụng tạo và tham gia quiz trực tuyến, được thiết kế để phục vụ nhu cầu giáo dục và giải trí của người dùng. Ứng dụng cung cấp báo cáo chi tiết về kết quả quiz, bao gồm số câu trả lời đúng, thời gian hoàn thành và điểm số, QuizHappy có một thư viện quiz phong phú, được tạo bởi cộng đồng người dùng, bao gồm nhiều lĩnh vực khác nhau như giáo dục, khoa học, văn hóa và giải trí.

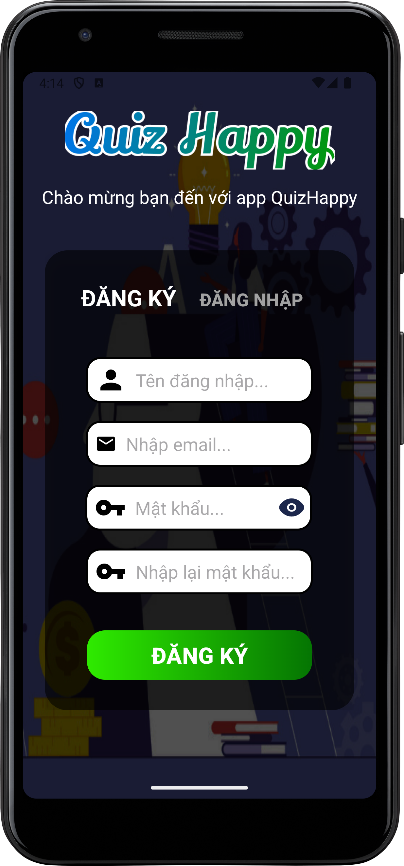
Với giao diện thân thiện và dễ sử dụng, QuizHappy phù hợp với mọi đối tượng người dùng, từ học sinh, giáo viên đến người dùng phổ thông.

1. THIẾT KẾ HỆ THỐNG
   1. Sơ đồ logic:



Sơ đồ logic của ứng dụng

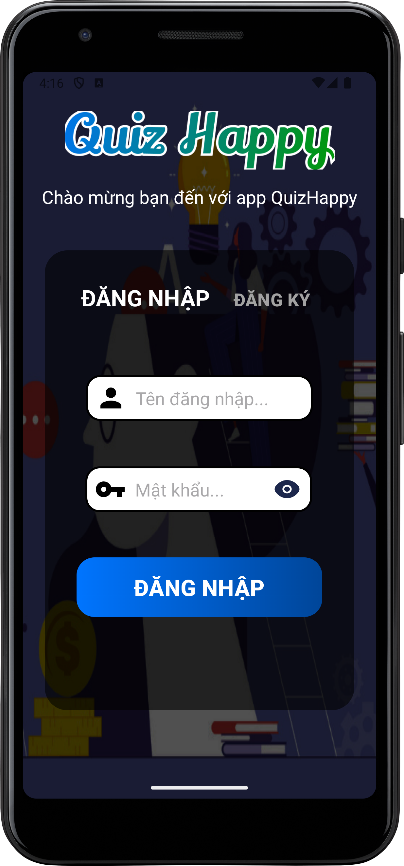
* Database: SQLite
* Hỗ trợ hệ thống: android 14.0.
* Ứng dụng:
  + Giao diện thân thiện, màu sắc hài hòa và dễ dùng.
  + Hệ thống sẽ có hệ thống đăng nhập và đăng ký.
  + Hệ thống có giao diện để chọn các chủ đề.
  + Chỉnh được các thông tin của người dùng(upload hình ảnh).
  1. Giao diện:
     1. Giao diện Intro:

Màn hình Intro

* + 1. Giao diện đăng ký:

Mà hình Đăng Ký

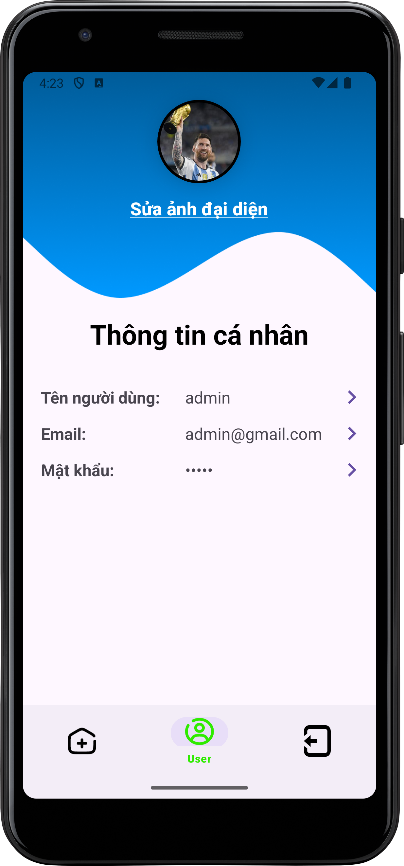
* + 1. Giao diện đăng nhập:

Màn hình Đăng Nhập

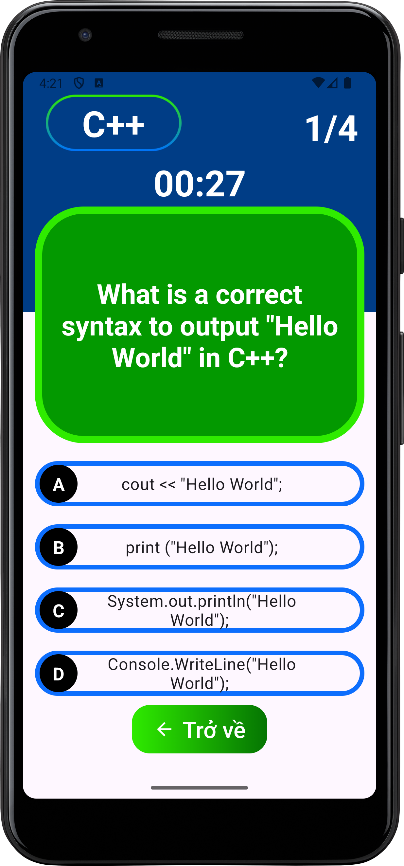
* + 1. Giao diện Home:

Màn hình Home

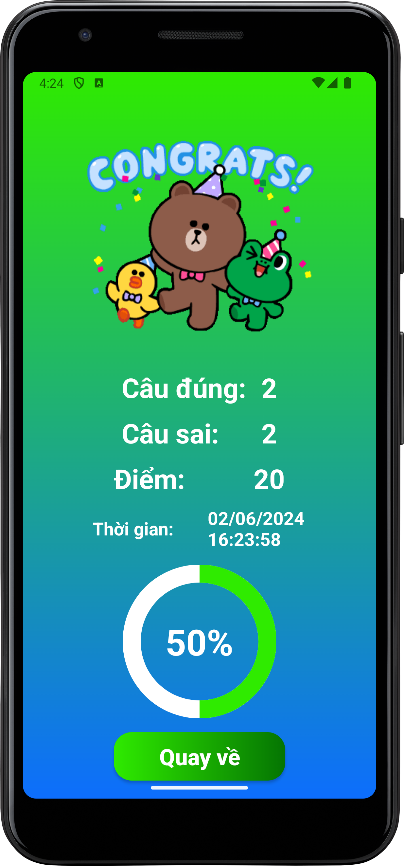
* + 1. Giao diện User:

Màn hình User

* + 1. Giao diện Quiz:

Màn hình Quiz

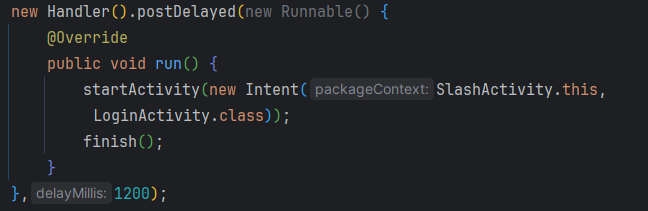
* + 1. Giao diện Result:

Màn hình Result

1. CÀI ĐẶT HỆ THỐNG
   1. Màn hình Intro:

Để thực hiện chức năng này, giao diện được tổ chức như sau:

Màn hình Intro

Chức năng này dùng để làm Intro đầu vào của app. Để chức năng được thực hiện thì tôi dùng thư viện Handler. Điều này nhằm mục đích là nó sẽ xuất hiện trong khoảng 1,2s và được thực hiện bởi hàm sau:

Sử dụng thư viện Handler

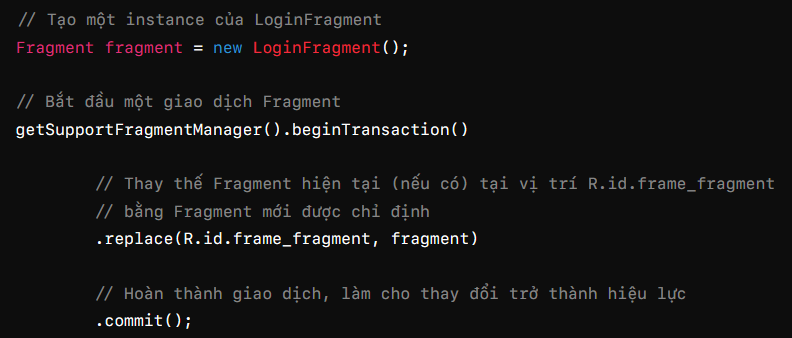
* **new Handler()** là để khai báo một Handler mới.
* **postDelayed** là một phương thức trong lớp ‘Handler’ của Android được sử dụng để lên lịch cho việc thực thi một đoạn mã sau một khoảng thời gian nhất định(delay Millis).
* Sau 1,2s thì phương thức Intern được dùng để chuyển sang một Activity khác, cụ thể là loginActivity.
  1. Chức năng đăng nhập:

Để thực hiện chức năng này, giao diện được tổ chức như sau:



Tổ chức giao diện đăng nhập

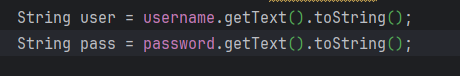
Trong chức năng này, tôi dùng frame\_fragment để thực hiện việc chuyển qua lại giữa đăng nhập và đăng ký. Về phần code xử lý như sau:



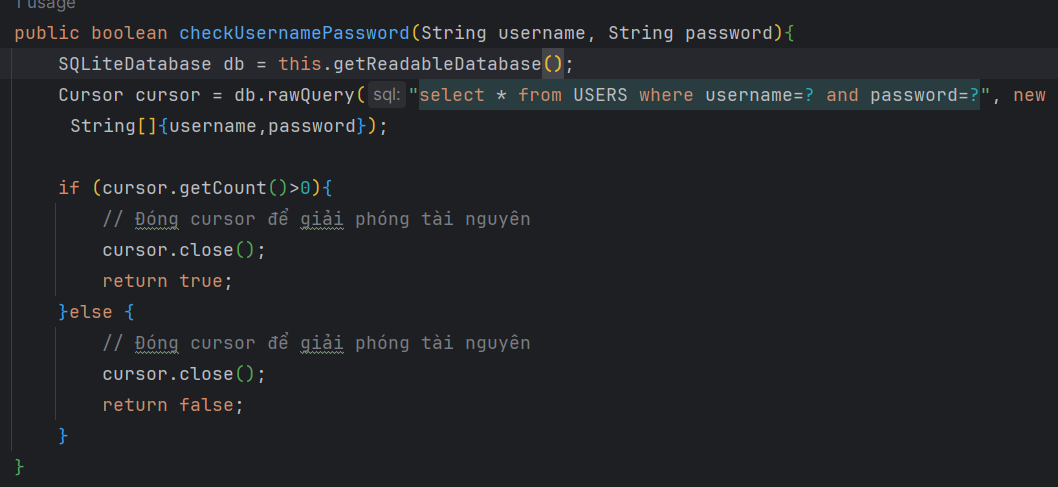
Xử lý Fragment

Ở form đăng nhập, về xử lý như sau:

* Dữ liệu tên đăng nhập, mật khẩu được lưu trong SQLite và để đăng nhập được thì ta phải đọc nó từ SQLite rồi đem đi so sánh.
* Thuật toán như sau:

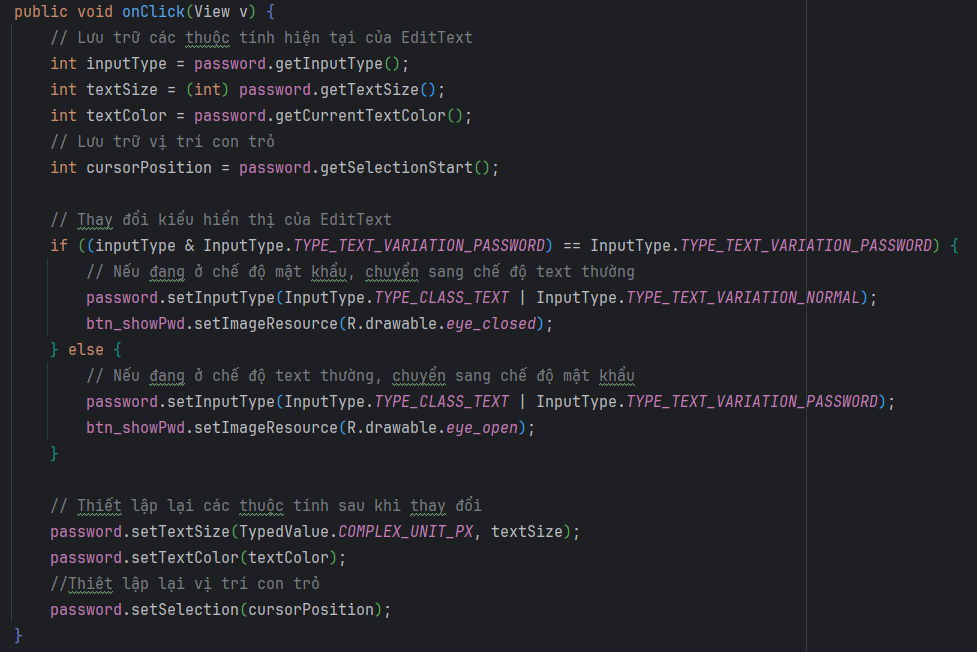
Cách lấy giá trị TextView

* Đây là câu lệnh để lấy chuỗi đã nhập trong ô input sau đó thì đem đi so sánh với dữ liệu có sẵn trong database.



Phương thức kiểm tra tên người dùng và mật khẩu

* Để thực hiện chức năng hiển thị mật khẩu thì ta thực hiện lần lượt các câu lệnh sau:



Phương thức hiển thị mật khẩu

* Đầu tiên thì ta kiểm tra kiểu hiển thị của input là gì bằng cách gọi **TYPE\_TEXT\_VARIATION\_PASSWORD** để kiểm tra xem input có phải hiển thị kiểu password hay không. Nếu đúng thì chuyển sang chế độ thường **TYPE\_TEXT\_VARIATION\_NORMAL,**sau đó thì ngược lại.
* Ngoài ra, tôi còn lưu lại các kiểu thuộc tính được dùng cho input đó để sau khi thay đổi thì không bị mất kiểu thuộc tính đó.
  1. Chức năng đăng ký:

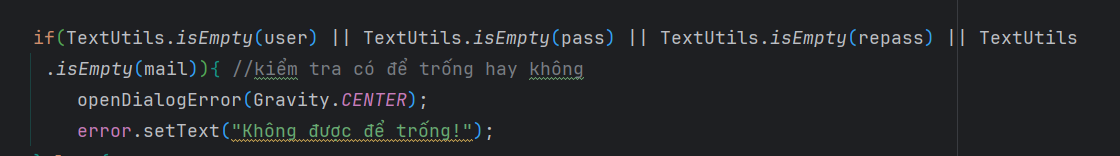
Để thực hiện chức năng này, giao diện được tổ chức như sau:



Tổ chức giao diện đăng ký

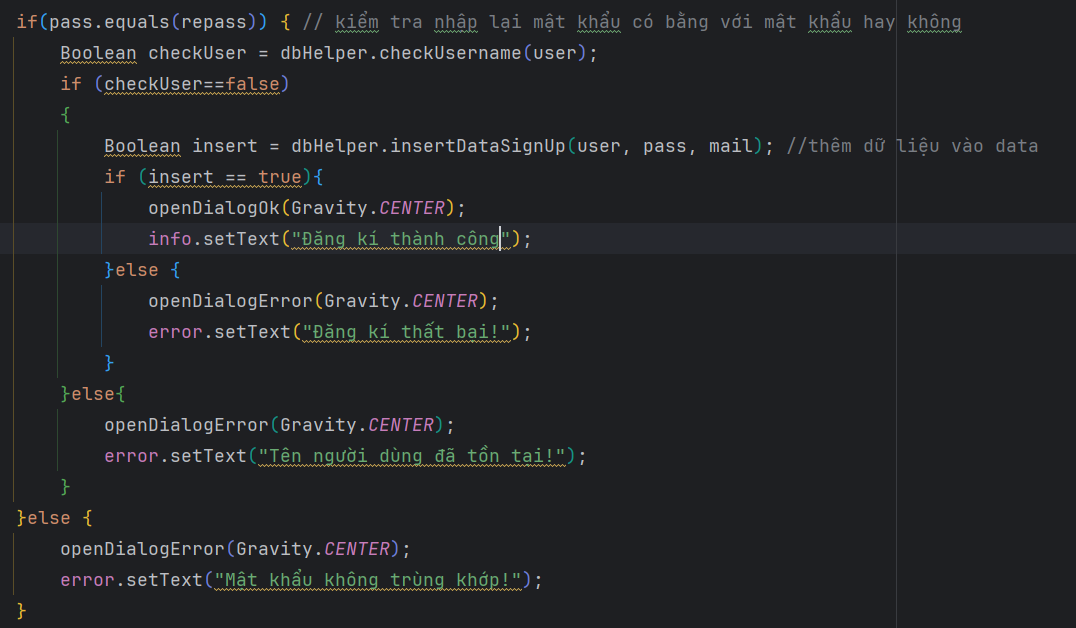
Tương tự, thì chức năng này cũng là một fragment, về phần code xử lý như sau:

* Đầu tiên thì ta kiểm tra xem các input có bị để trống hay không nếu có thì sẽ thông báo ra lỗi.



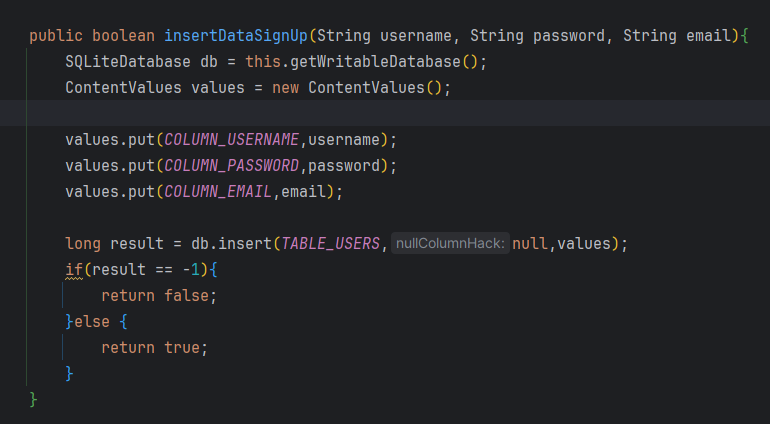
Kiểm tra không được để trống

* Các phương thức như isEmpty() là để kiểm tra các đầu vào của các giá trị đó có bị rỗng hay không.
* Sau đó thì ta đi kiểm tra lần lượt là mật khẩu nhập lại có trùng khớp hay không và tên người dùng đã tồn tại hay chưa. Code xử lý như sau:



Kiểm tra đăng ký

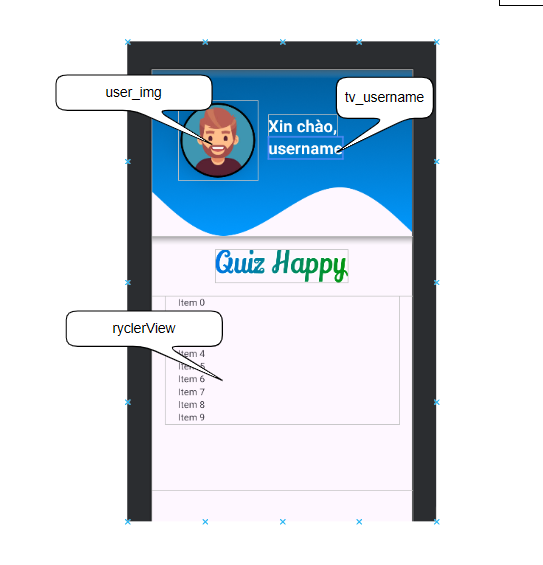
* Nếu đã đạt đủ điều kiện thì lưu lại vào trong database.



Phương thức nhập dữ liệu vào database

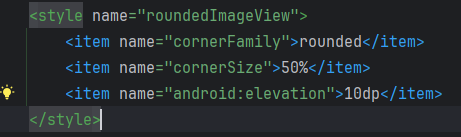
* 1. Màn hình chính:

Để thực hiện chức năng này, giao diện được tổ chức như sau:



Tổ chức giao diện Home

Ở giao diện này, dùng để hiển thị ra các loại câu hỏi. Phần code xử lý như sau:

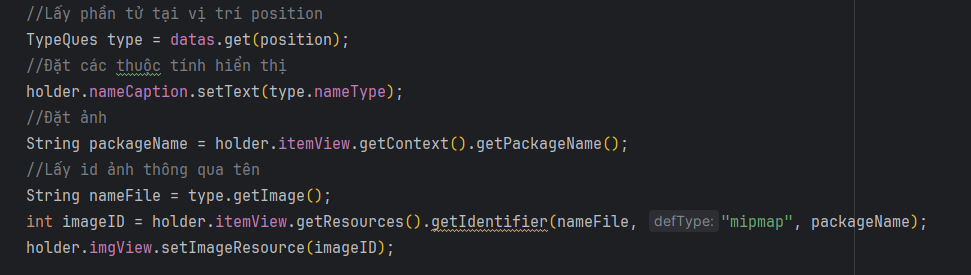
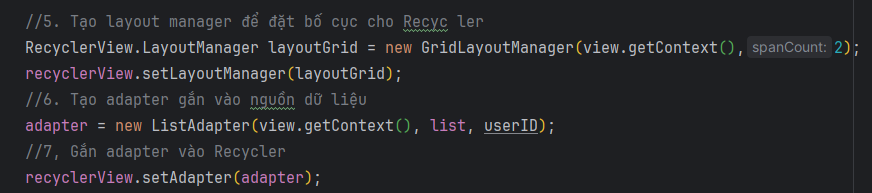
* user\_img: để hiển thị ảnh đại diện của người dùng. Để nó có thể tròn như vậy thì thực hiện các cậu lệnh như sau:

Thiết kế kiểu hình round

* Ta thiết kế kiểu cho nó cornerFamily: rounded là cho nó là dạng hình chọn, cornerSize: 50% độ tròn của nó và elevation là độ đổ bóng.
* Sau đó thì gọi nó ra.

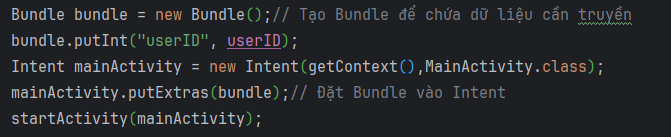


Gọi kiểu hình ra

* ryclerView dùng để hiển thị các loại câu hỏi được xử lý bằng cách dùng Adapter để cung cấp dữ liệu và giao diện cho mỗi mục trong danh sách. Khi bấm vào các item trong đó nó sẽ được chuyển đến màn hình trả lời câu hỏi.

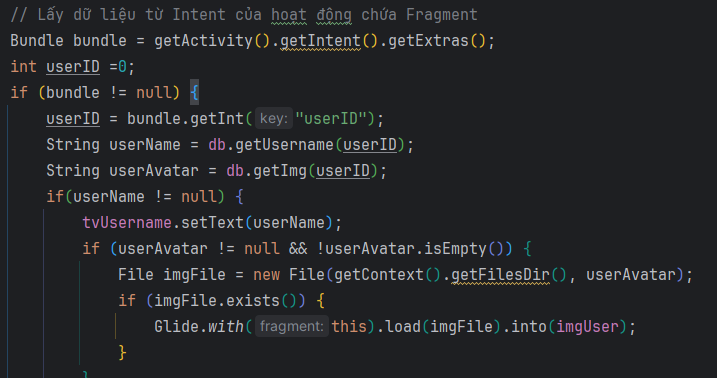
Cài đặt Adapter

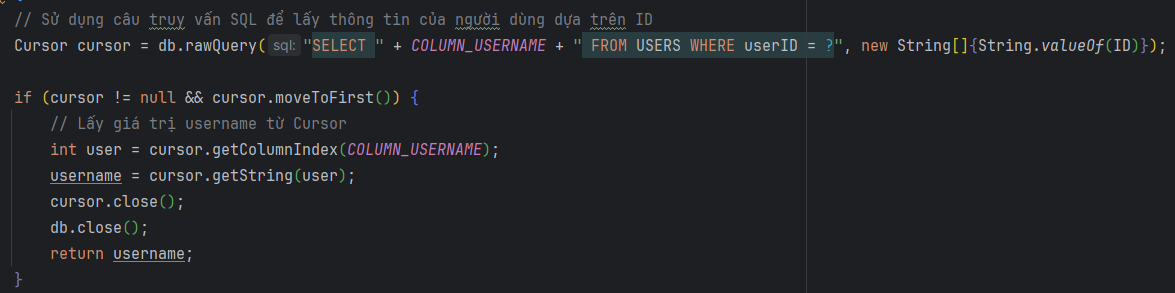
Thiết lập RecyclerView

* Ở đây tôi dùng layout Grid với 2 cột để hiển thị danh sách các loại câu hỏi.
* Để hiển thị được ảnh đại diện và tên người dùng thì đầu tiên ta phải có userID được lấy từ login thông qua Bundle. Dưới đây là cách để lấy userID và lưu nó lại vào trong Bundle thông qua putInt(“tên tag”,dữ liệu) hay còn gọi là key và value.

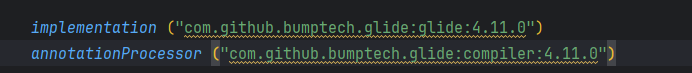
Tạo Bundle lấy userID

* Sau đó thì qua fragment ta đã truyền gọi nó ra.

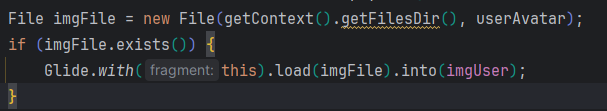
Lấy userID từ Bundle

* Sau khi có được userID thì ta đem nó đi để lấy được username và img đã được lưu thông qua việc truy vấn đến database. Ví dụ như code sau:

Truy vấn để lấy username

* Để hiển thị được ảnh thì ta phải lấy được được userAvatar của ảnh được lưu trong database rồi đem nó đi lấy đường dẫn đến ảnh bằng cách là dùng đối tượng File. Đầu tiên thì tạo một đường dẫn đến tệp ảnh trong bộ nhớ của ứng dụng rồi gắn cái userAvatar vào. Rồi sau đó kiểm tra xem tệp ảnh đó còn tồn tại hay không nếu có thì dùng Glide để hiển thị ảnh. Ví dụ code như sau:
* Thêm thư viện:

Thêm thư viện Glide

Sử dụng Glide để load hình ảnh

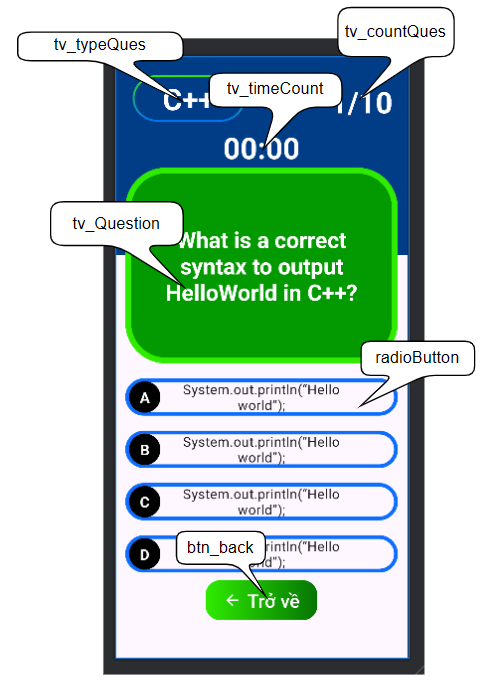
* 1. Chức năng thông tin người dùng:

Để thực hiện chức năng này, giao diện được tổ chức như sau:

Ở chức năng này thì, việc hiển thị user\_img cũng là dùng Glide. Các phần xử lý khác:

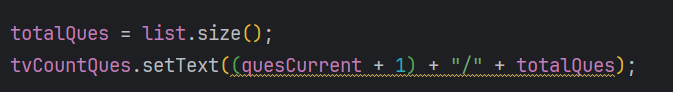
Tổ chức giao diện User

* btn\_edit\_img: dùng để thay đổi ảnh đại diện bằng cách gọi tới intent màn hình thư viện của điện thoại. Trước khi gọi cái đó thì đầu tiên tôi phải vào manifest và thêm đoạn code sau:
* <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />
* Sau đó thì vào fragment và thêm đoạn code sau
* if (ContextCompat.*checkSelfPermission*(getContext(), Manifest.permission.*READ\_EXTERNAL\_STORAGE*)
* != PackageManager.*PERMISSION\_GRANTED*) {
* ActivityCompat.*requestPermissions*(getActivity(),
* new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*}, *REQUEST\_PERMISSION*);
* }
* Rồi sau đó mới gọi đến intent
* Sau đó tôi lấy đường dẫn của ảnh để lấy được hình ảnh và lưu lại trong bộ nhớ trong của điện thoại và cũng lưu vào trong SQLite với cái tên kết bởi các yếu tố sau:
* String imageName = "user\_avatar\_" + userID + "\_" + timeStamp + ".png";
* Các textView thì dựa vào việc lấy được userID đã làm ở HomeFragment rồi truy vấn đến database để lấy dữ liệu hiển thị lên.
* Các btn\_edit thì khi bấm vào sẽ xuất hiện một dialog để giúp chúng ta sửa các dữ liệu.
  1. Chức năng trả lời câu hỏi:

Để thực hiện chức năng này, giao diện được tổ chức như sau:

Tổ chức giao diện Quiz

Phần xử lý:

* tv\_typeQues: là phần textView hiển thị loại câu hỏi được chọn. Để hiển thị được thì ta cũng dùng Bundle để lưu lại typeID ở trong Adapter của listQues. Mà để lấy được typeID để mà lưu vào thì ta lấy vị trí của item được click + 1 thì ta được typeID.
* tv\_countQues: là phần hiển thị câu hỏi hiện tại/tổng số câu hỏi. Phần code xử lý như sau:

Lấy tổng số câu và câu hiện tại

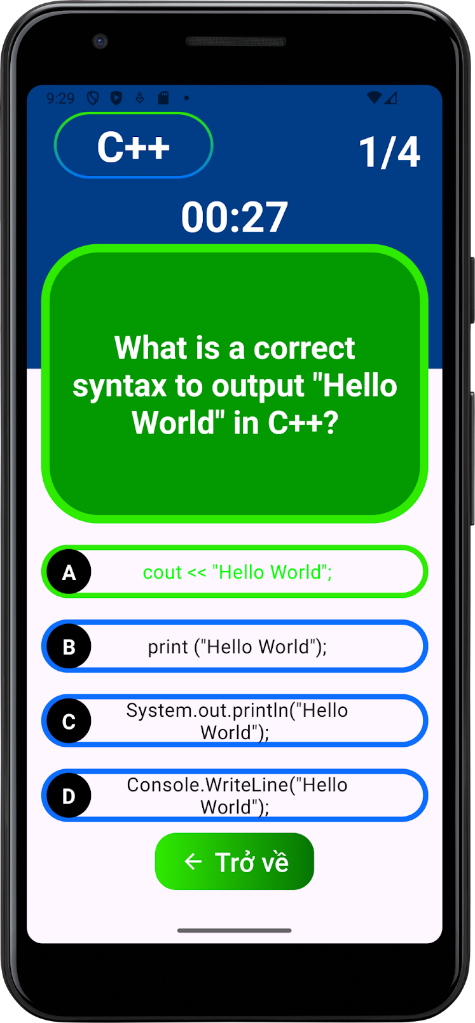
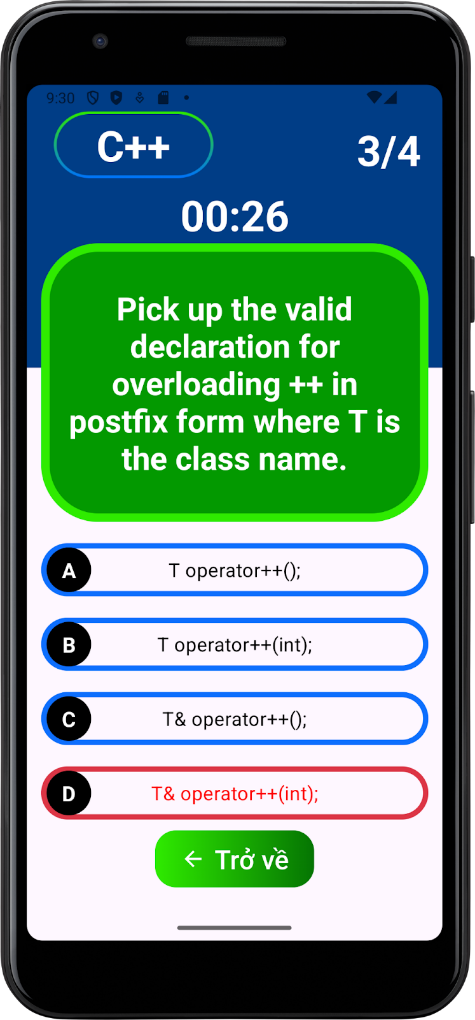
* tv\_timeCount: dùng để đếm thời gian trả lời câu hỏi - thời gian giới hạn là 30s và khi còn 10s thì nó chuyển sang màu đỏ. Các để đếm được thời gian thì cần dùng đến đối tượng CountDownTimer sau đó thì xử lý code như sau:

Phương thức đếm thời gian

* Khi thời gian hết thì nó sẽ tự động chuyển sang câu hỏi khác.
* Để hiển thị được các câu hỏi và các câu trả lời thì ta phải truy vấn đến database dựa vào typeID và đem nó lưu vào Array<Questions>(). Code như sau:

Truy vấn lấy dữ liệu bỏ vào mảng

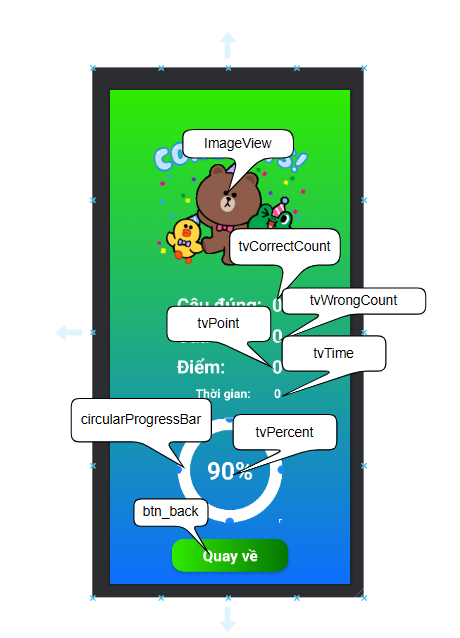
* Sau đó thì ta lấy dữ liệu từ mảng của đối tượng Questions để mà gắn vào TextView và RadioButton tương ứng.
* Lý do mà sử dụng RadioButton mà không sử dụng các button khác là vì cho phép chỉ được chọn một đáp án.
* Và khi chọn được đáp án đúng thì đường viền + chữ nó sẽ hiển thị xanh lá còn sai thì màu đỏ. Không phân biệt đúng hay sai vẫn chuyển câu hỏi khác. Kết quả như sau:



Hiển thị màu sắc sai và đúng

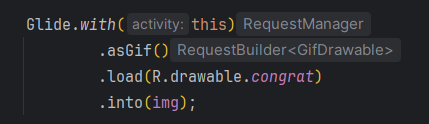
* Để xử lý được chức năng này thì ta thực hiện lần lượt như sau:
* Đầu tiên khi bấm vào một đáp án bất kỳ thì đem vị trí của cái đáp án đó so sánh với vị trí đáp án đúng được lưu trong database.
* Tiếp đó là kiểm tra xong nếu đúng thì gắn cho radioButton đó drawable của đúng còn sai thì gắn drawable của sai.
* Sau khi đến câu cuối là tương ứng với tổng số câu thì sẽ có một dialog hiển thị lên để thông báo hoàn thành và ngoài ra cũng lưu các điểm số, thời gian, loại câu hỏi đã làm vào trong database còn số câu đúng và số câu sai thì được dùng lưu vào Bundle.
* btn\_back thì được dùng để trở về Home bằng cách dùng Intent.
  1. Chức năng hiện kết quả:

Để thực hiện chức năng này, giao diện được tổ chức như sau:

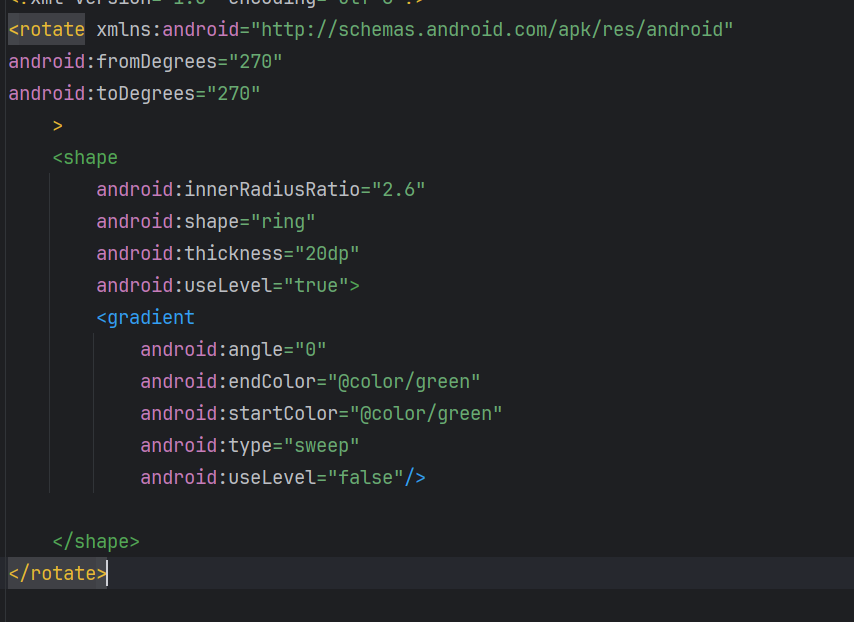


Tổ chức giao diện Result

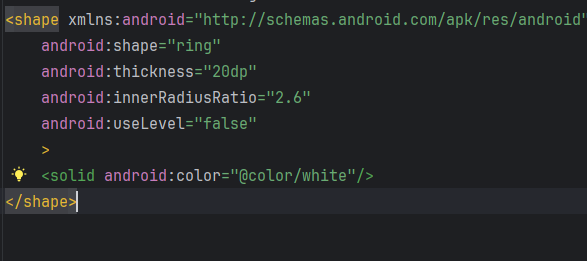
Phần xử lý:

* ImageView: là một ảnh gif vì vậy không dùng src thông thường được do vậy tôi đã sử dụng Glide để hiển thị. Được code như sau:

Sử dụng Glide để dùng ảnh Gif

* Bắt buộc phải có asGif thì mới sử dụng được gif sau đó thì load lên ảnh lên ImageView.
* Các textView được hiển thị lần lượt xử lý như sau:
  + tvCorrectCount: thì lấy được dựa vào Bundle được lưu từ QuestionsActivity. Tương tự thì tvWrongCount cũng vậy.
  + Điểm và thời gian thì được lấy từ database thông qua truy vấn đến bảng RESULT.
  + tvPercent thì được tính bằng công thức: số câu đúng/ tổng số câu \* 100.
* double percentage = (double) anwTrue / totalques \* 100;
* percent.setText(String.*format*("%.0f%%", percentage));
* Để circularProgressBar sử dụng được thì các bước làm không quá phức tạp:
* Đầu tiên thì thiết kế cái đường trong xml để nó chạy theo phần trăm bằng những dòng code sau.

Thiết kế vòng xoay circularProgressBar

Thiết kế nền của circularProgressBar

* + Sau đó thì gọi trong ProgressBar

Gọi các drawable vừa thiết kế ra

* + Cuối cùng là phần xử lý code, ta dùng Handler để làm animation chạy từ nhanh đến chậm cho vòng tròn. Để làm vòng tròn đó chạy thì phải có giá trị và giá trị đó thì phải bé hơn hoặc bằng giá trị percent sau đó thì giá đó ****tăng dần lên.

Làm animation cho circularProgressBar

1. ĐÁNH GIÁ KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN
   1. Công nghệ sử dụng

* Front-end: Figma, XML.
* Back-end: Java, SQLite.
  1. Kết quả đạt được
* App hoạt động ổn định.
* Có thể thực hiện thay đổi ảnh đại diện, trả lời các câu hỏi và xem kết quả.
* Nhiều chức năng khác hoạt động ổn định.
  1. Hướng phát triển
* Tiếp tục bổ sung thêm những chức năng chưa hoàn thiện.
* Phát triển hệ thống trực quan, dễ dàng sửa đổi .
* Phát triển cho nhiều nền tảng khác nhau.

Tài liệu tham khảo

1. PHENIKAA, T., & TIN, K. C. N. T. Báo cáo đồ án cơ sở Đề tài: Ứng dụng cẩm nang du lịch.