

利酒魔人

18-12-09

概要

日本酒の利き酒をしても当てられない。これは、自分の中で評価指標がなく、異なる酒が同じエリアにマッピングされているためだと考えられる。

そこで、利き酒のやり方に則った評価指標を詰め込んだ Android アプリを開発する。自由記述式（感想など）の既存アプリでは解決できない「どう評価したらよいかわからない」を解決する。

※日本酒全般の知識は文献[1]を参考にする。

※利酒魔人はアイコンのキャラの名前。前作が割勘魔人だったので、同じ方針を引き継ぐことにした。

使い方

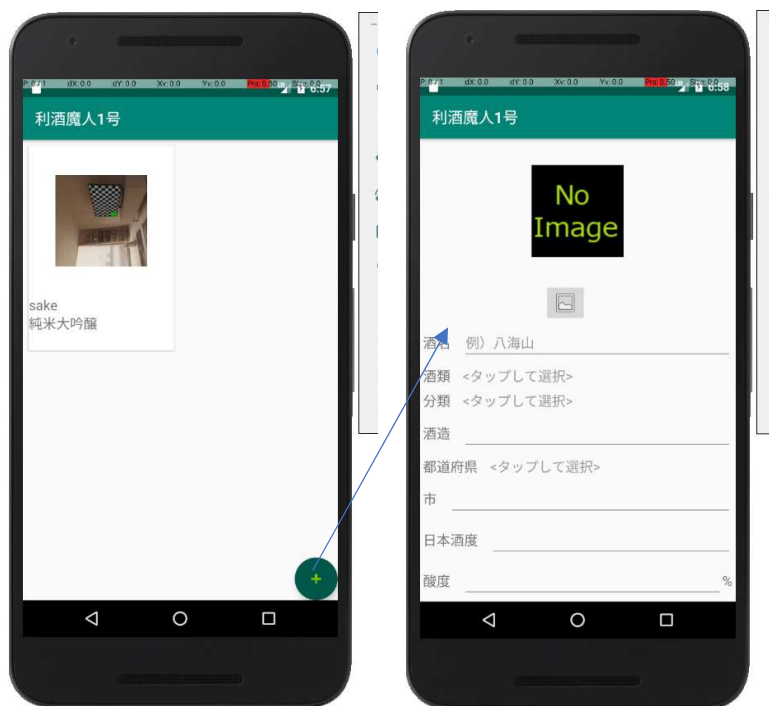
19-12-22

権限の追加

カメラとストレージの使用権限を追加する。



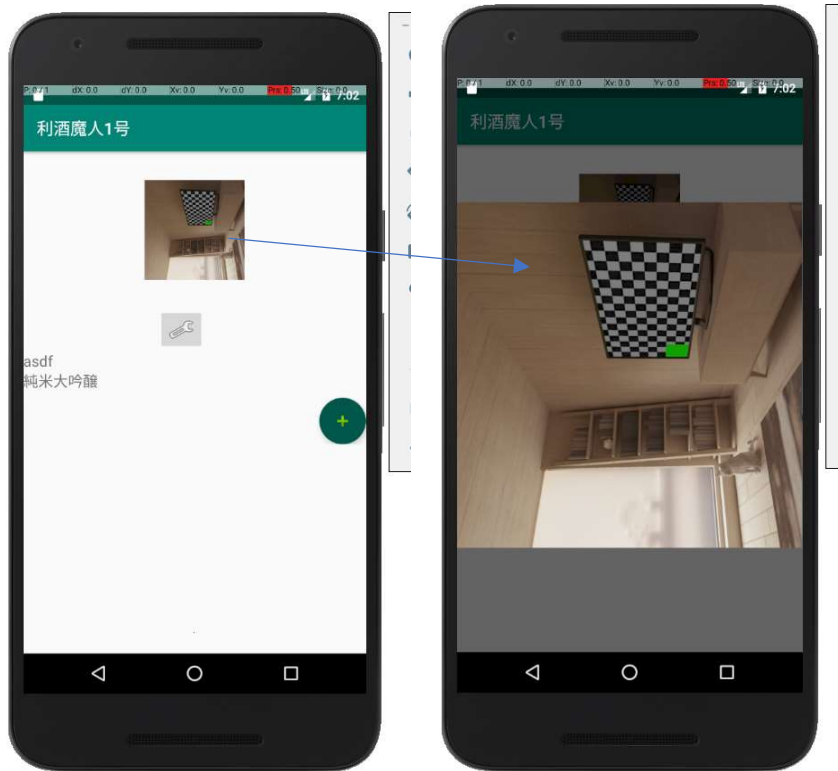
酒情報の登録



メイン画面から、新規の酒情報を登録する。ここでは、static な（何回レビューしても変わらない）情報を入力する。

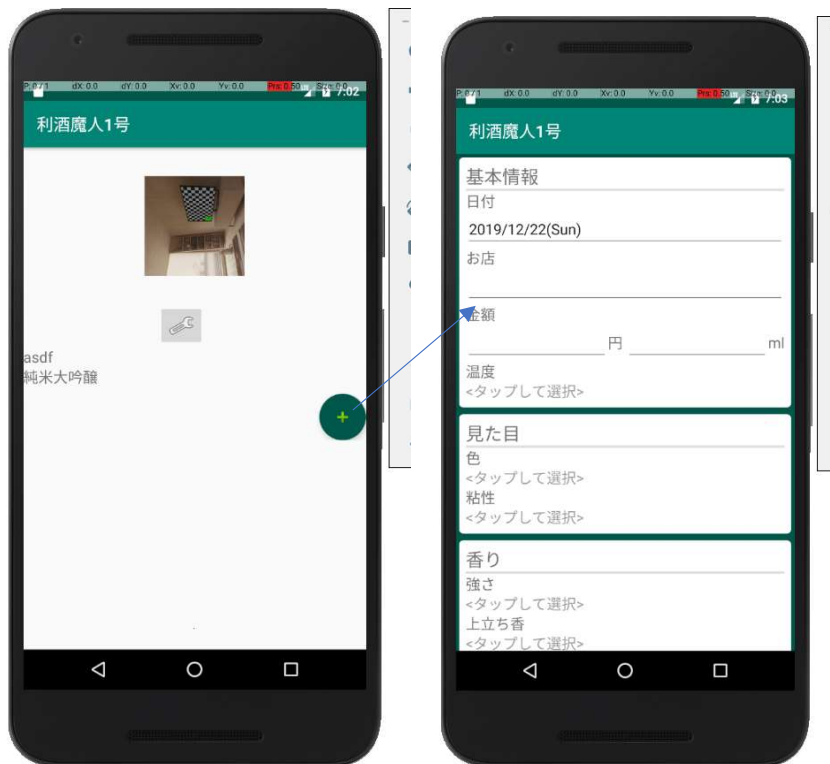


「米をまとめる」機能は、麴米と掛米を区別せずに入力する。実際には、両方に同じデータを入力している。入力が完了したら、「追加」ボタンをタップする。

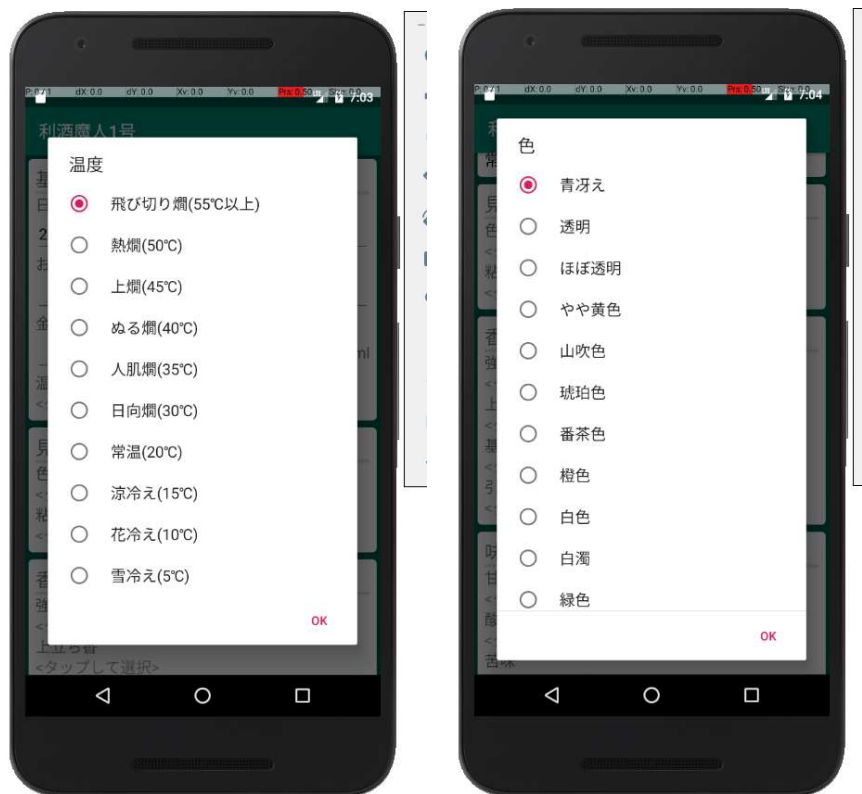


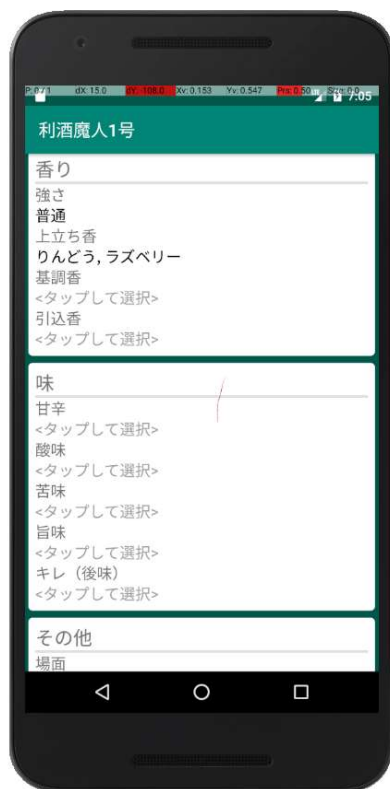
入力した情報が表示される。ここでバックボタンを押すと、メイン画面に戻る。
写真を登録していると、画像タップで大きい画像が表示される。

酒のレビュー



追加ボタンから、新たな酒レビューを追加する。ここでは、dynamic な（レビューするたびに内容が変化する）情報を記入する。基本情報、見た目、香り、味、その他がある。





香りでは、フローラル、フレッシュ、穏やか、ふくよか、マイナスの5種類から選択できる。



味は、基本的には5段階評価になっている。



その他には、場面（ビール2杯後、利き酒の練習等の備忘録）、合わせた料理、自由記述式の

感想を書ける。また、好きか嫌いかという評価もできる。



レビューが終わると、レビュー日と感想が表示される。

酒情報およびレビュー情報はデータベースとして管理される。

開発状況まとめ

19-12-22

一通り使えるものが出来上がった。現状と課題をまとめる。

できたこと

解決した技術的な課題は以下のとおり。

- Activity-Fragment 連携による状態遷移
- Fragment の Listener を Activity で処理する方法
- カメラ・ドキュメントからの画像の登録 (SakeInformation)
- RecyclerView + CardView によるデータのリスト表示 (SakeList)
- 1 要素選択式の ExpandableListView (都道府県)
- ヘルプ付きの ExpandableListView + Checkbox (酒分類, 香り)
- 動的な LinearLayout の追加 (上のヘルプ)
- 酒情報, レビュー情報の SQLite3 による DB 化
- ちょっとイケてる風の UI (酒レビュー)
-

課題

- カメラ・ドキュメントを使用するための権限の確認
今は, 何も確認していないから, 権限がないと落ちる
- 画像の保存ディレクトリを任意にする
今は, Pictures と限定されている. MediaStore の使い方がイマイチ分かっていない.
- DB の更新処理
DB にアクセスする回数を減らしたい. できるだけ手元のリストを操作して, 適切なタイミングで変更をコミットするようにしたい.
- 各種例外処理
落ちそうだと思うところはあるけど, いまは何もしていない. 落ちたら修正する感じにしている.
- 画像の削除
DB からリストを削除した時に, 関連する画像も削除したい. また, 一度撮影したけど, DB に登録しなかった写真はストレージに残り続ける. これを一掃する処理を実装したい.
- ハードコード

サイズや文字列をハードコードしている箇所が多々ある。できれば、res/values を使いたい

- 検索機能

せっかく DB にしたのだから、検索したい。例えば、値段で酒をソートしたり、好き嫌いでソートしたり。

- DB 解析機能

DB の内容を解析して、好きなタイプの酒を提案したり。

- 順番入れ替え

DB の順番を入れ替える作業

- ヘルプ機能

利き酒のやり方や、香りに関する説明はまだない。

- エクセル書き出し機能

エクセルに書き出して PC で解析したい場合もあると思う。バックアップとしても使える。

-

要件定義

1) 利き酒アプリにはどのような機能が必要かを検討する。2) 利き酒の評価指標からアプリに盛り込む内容を決める。

機能

- ・色, 香り, 味などの指標があるので, それぞれ「選択式」で選ばせる
- ・自由記述も設ける
- ・自分の評価をデータベース化し, いつでも振り返れる
- ・その指標の説明が見れるとよい (初心者なのでそれがどんな物を指すか分からない)

追加機能

- ・同じ酒への評価結果をまとめ, 評価のばらつきなどを見る
- ・ネット上の評価との突合せを行う
- ・写真を添付できる
- ・利き酒のコツややり方なども載せる
- ・指標が多すぎるので, ボタン一つで簡略モード, 上級モードなどにしたい

まとめると,

- ・利き酒機能
- ・ライブラリ機能

➔ 既存のテイスティングノートを参考にしてアプリ化する方針で。

文献[6][7]が具体的, かつ網羅的な内容で, とても参考になる。

お酒の基本情報

プロフィール情報もお酒の理解には重要。どのような内容をアプリに記録するか, テイスティングノートを参考に決定する。

利き酒の評価指標

調査結果は「利き酒のやり方」に示す。この中から抜粋して, どのような評価指標を盛り込むかを決定する。

マイナスとプラスの評価が混ざっているが、分離が難しかった。具体的にバナナとか言う方法と、華やかな果実、のように言う方法がある。まずは前者で作ってみる？

色)

澄み方：

青冴え、透明、ぼけ、混濁

色調：

無色、ほぼ無色、やや黄色、黄金色（山吹色）、番茶色、白色

香り)

華やかさ：

花：梅の花、アカシア、チューリップ

果実：ライチ、メロン、りんご、バナナ、洋なし

爽やかさ：

花：さざんか、りんどう、水仙

ハーブ：桜の葉、タイム、レモンバーム、天草、月桂樹の葉

果実：レモン、ライム、オレンジ、グレープフルーツ

菜類：クレソン、三つ葉、瓜、山菜

木：青竹、笹の葉、新芽、松の葉

穏やかさ：

鉱物：ミネラル、岩清水、木炭、オイル

菜類：ふき、白菜、菜の花、大根、人参、ごぼう

木：朴葉、ヒノキ、松ヤニ、カエデ、和紙

ナッツ：栗、ピーナッツ、松の実、銀杏、胡麻

ふくよかさ：

穀物：稲穂、つきたての餅、白米、葛、豆腐、蕎麦

ナッツ：栗、カシューナッツ、落花生

スパイス：クローブ、シナモン、キャラウェイ、香木、バニラ

乳製品：バター、生クリーム、カスタードクリーム、ヨーグルト

熟成香：磯の香り、ドロップ、水ごけ、青のり、カステラ、昆布、蜂蜜、干し椎茸

マイナス：

アルコール、カビ

...

➡ ここをオリジナルにする必要はないと思った。本気でアプリリリースする気になったら考える。まずは既存のテイスティングノートを再現する。

利き酒のやり方

まずは本業の人のやり方を知ることが大切。利き酒師はどうやって利き酒をするのか？

参考: 文献[1][4]

手順

1. 色を見る
2. 香りを嗅ぐ
3. 味を見る

ポイント

- ・とにかく表現する
- ・その表現に再現性がある
- ・青い蛇の目は日本酒の黄色みを強調する働きがある

上達のコツ

- ・良いものを飲む
- ・ひたすら訓練（記録を取る）
- ・利き酒用語を覚える

香りには3種類 **エラー! 参照元が見つかりません。**

- ・上立ち香 (ウワタチカ): 鼻で嗅いだ香り
- ・基調香 (キチョウカ): 口の中で転がしたときの香り
- ・口中香 (含み香) (コウチュウカ): お酒を飲みこんだ後に鼻から抜ける香り
- 3つの香りがほぼ一緒だと, 「香りのバランスがよい」と表現される

メモ

- ・純米大吟醸が最高とは限らない, 吟醸香はフルーティーさを出す

評価指標

評価はプラスとマイナスの2通りがある [3].

色

透明 光沢 混濁 茶褐色 黄金色 琥珀色 テリ 水様 冴 [1]

-) 異常着色, 濁り [3]

[5]

冴え（さえ）	美しく透き通った光沢のこと。やや青みがかって見える状態を青冴え（あおざえ）といい、高く評価されます。
照り（てり）	うっすら山吹色に艶の出た状態。おおむね好まれる。
ぼけ	少々混濁して、色彩がぼやけていること。
澄明度	自然に造られた、澄んだみずみずしいきらめきを指す。
透明度	（酒蔵によって人工的に）造られた（精製された）透明さを指す。
黄金色（こがねいろ）	照りの中でも最も好まれる色調。
番茶色（ばんちやいろ）	古酒などに多い、やや濃く熟成した色調。
色沢良好（しきたくりょうこう）	鑑評会などで語られる、色合いが好ましいさまの定番表現。
色沢濃厚（しきたくのうこう）	かなり色がついている状態。好ましいとの印象を受ける愛飲家も多い。
混濁	いろいろな色調に濁っていること。見た目の評価は得がたいが、味もまずいとは限らない。

香り

果実香 甘酒臭 紙臭 木香 老香 酸臭 吟醸香 力強い 穏やか 華やか 重い 新酒香 炭臭 枯れた スパイス 麴臭 調和 蒸米臭 [1]

+) 新酒香・吟醸香・熟成香・樽香 -) 老香・生老香・ジアセチル臭・紙臭・濾過癖・酢工チ臭・カビ臭・酸臭 [3]

[5]

華やかな香り	花系	梅の花、アカシア、チューリップ 等
--------	----	-------------------

	果実系	ライチ、メロン、りんご、バナナ、洋なし 等
爽やかな香り	花系	さざんか、りんどう、水仙 等
	香草系	桜の葉、タイム、レモンバーム、天草、月桂樹の葉 等
	(ハーブ)	
	果実系	レモン、ライム、オレンジ、グレープフルーツ 等
	菜類系	クレソン、三つ葉、瓜、山菜 等
	木系	青竹、笹の葉、新芽、松の葉 等
穏やかな香り	鉱物	ミネラル、岩清水、木炭、オイル 等
	菜類	ふき、白菜、菜の花、大根、人参、ごぼう 等
	木系	朴葉、ヒノキ、松ヤニ、カエデ、和紙 等
	ナッツ類	栗、ピーナッツ、松の実、銀杏、胡麻 等
ふくよかな香り	穀物類	稲穂、つきたての餅、白米、葛、豆腐、蕎麦 等
	ナッツ類	栗、カシューナッツ、落花生 等
	スパイス類	クローブ、シナモン、キャラウェイ、香木、バニラ 等
	乳製品類	バター、生クリーム、カスタードクリーム、ヨーグルト 等
	その他	磯の香り、ドロップ、水ごけ、青のり、カステラ、昆布、蜂
	(熟成香)	蜜、干し椎茸 等

味

こくあり こくなし 芳醇 丸い あらい 濃い 薄い 重い 軽い 淡麗 滑らか 綺麗
渋味 キレ 旨味 後味 押し 幅 含香 酸味 調和 粘り [1]

+) 淡麗・軽い・なめらか・キレが良い・旨味がある・濃醇 -) うすい・酸うく・雑味・渋味・苦味 [2]

[5]

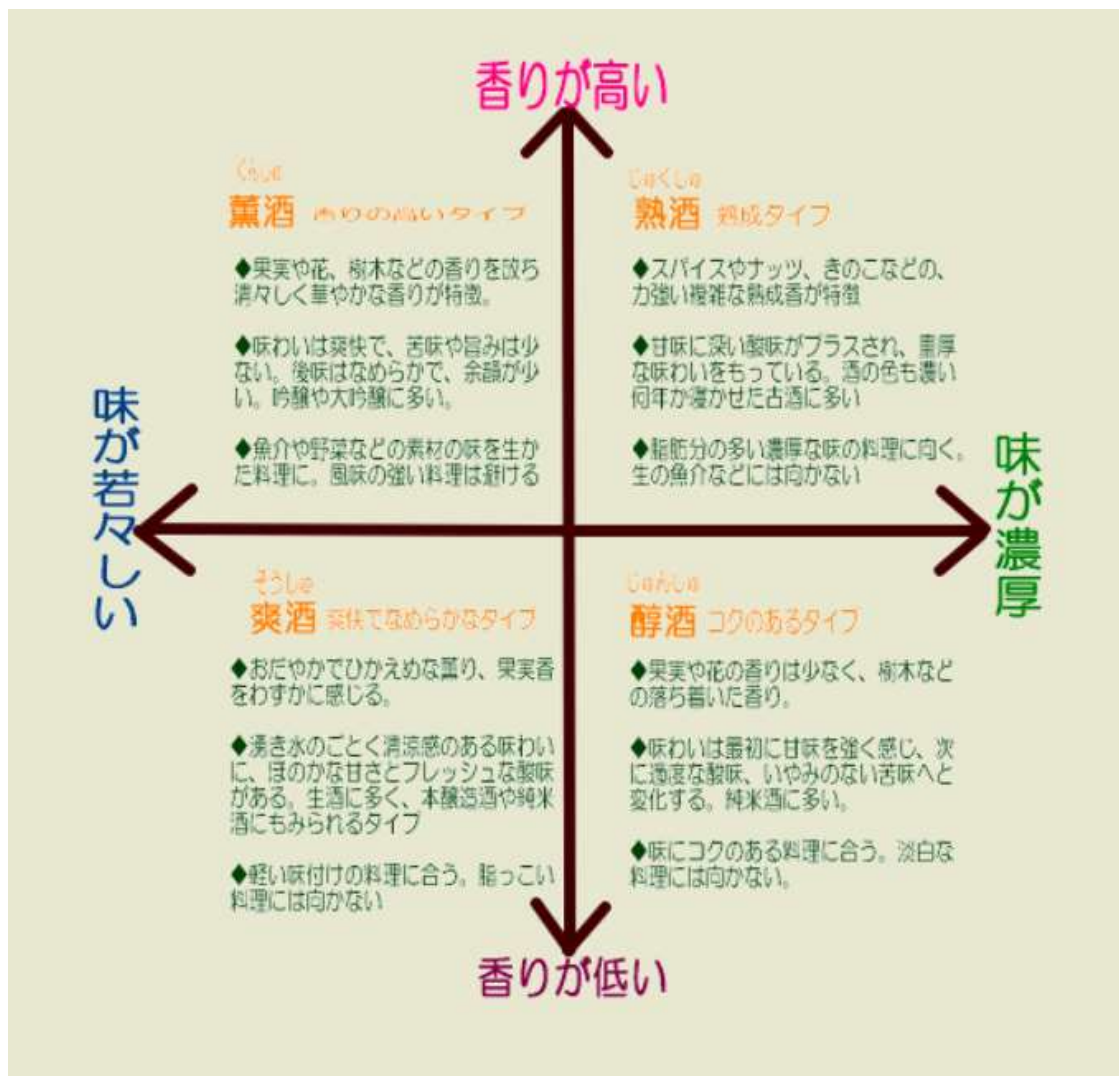
飲み口	サラサラした、サラリとした、艶やかな、キメの細かい、絹のような、緻密な、シャープな、角のない、まるみのある、穏やかな
甘み	強い、弱い、ほんのりとした、なめらかな、まろやかな、優しい、爽やかな、すっきりとした、落ち着いた、柔らかな

酸味	強い、弱い、シャープな、刺激的要素の強い、スッキリした、サラリとした、フレッシュな、爽やかな、清涼な、キメの細かい、クリーンな
苦味	強い、弱い、スッキリした、サラサラした、柔らかな、清々しい、爽快な、清涼なふくらみのある、スパイスのような、ハーブのような、根菜類のような
旨味	強い、弱い、しっかりした、ふくらみのある、力強い、充実した、ボリューム感のある、深みのある、芳醇な、存在感のある、コクのある、厚みのある、練れた

X-Y 軸上にプロットするやり方もある[4]

日本酒サービス研究会・日本酒匠研究会連合会のもの。

これは、評価後の可視化に使いそう。ペンタグラフもあったような・・・？



テイスティングノート

文献[6][7]

Android アプリにする

18-12-19

定義した要件をもとに、どんな機能で実現するかを考える。

Android での実現方法

- ・色, 香り, 味などの指標があるので, それぞれ「選択式」で選ばせる
- ・その指標の説明が見れるとよい (初心者なのでそれがどんな物を指すか分からない)
- ・自由記述も設ける
- 割勘魔人と同じく, RecyclerView が適する考える
- ・自分の評価をデータベース化し, いつでも振り返れる
- SQLite が使いやすいと思う. butterknife ってまだ使える?
- Kotlin Extension があるから必要ない

追加機能

- ・写真を添付できる
- カメラと連携し, 写真も DB として保存できるように.
- ・同じ酒への評価結果をまとめ, 評価のばらつきなどを見る
- ・ネット上の評価との突合せを行う
- DB 化が難しい. これは今後の課題とするのが現実的.
- ・利き酒のコツややり方なども載せる
- HELP view として実現できそう. あり.
- ・指標が多すぎるので, ボタン一つで簡略モード, 上級モードなどにしたい
- これは追加課題. 多分やらない.

開発ステップ

Fragment で RecyclerView を実現する.

SQLite を操作できる.

- ここまでで, 使えるものが出来上がる

写真を UP できる

- これはそう難しくない考える

実装上の課題

19-09-30

開発では勉強のために Kotlin を使うことにした。Kotlin は Java ほどサンプルコードが多くな
くて躓いている。以下がクリアすべき課題。

- fragment と activity の使い分け
- (完)fragment のボタンを操作して別の fragment に移動する方法(19-10-01)
- (完)データベースと連動(19-10-05)

SQLite with Anko の動作確認 OK

- (完)画像のデータベース化(19-10-06)

カメラで撮った画像をうまく取得できない。ファイル選択なら OK.

- (完)リスト表記 (19-10-22)
- 検索機能
- (完)カメラの扱い (19-10-26)

- database の更新方法 (DB の version の話)
- 開発プロジェクトのパッケージ分けの話
べたにコードを置いていくと、醜くなる・・・
- 今回はやめたけど、Fragment inside Fragment @kotlin ができない
<https://qiita.com/eaglesakura/items/2ab707755f5a504d9cf9>
これが参考になりそうなのだけど、otto って何？あと全ソースが載っていない
java で言うところのこれ。
<http://tsuyushiga.hatenablog.jp/entry/2014/09/30/235931>
-

開発

19-10-01

Kotlin で Fragment 遷移

以下が参考になる。これで Fragment と Activity の連携が学べる。

フラグメントのボタンを押して、Activity で状態遷移を規定する方法が分かった。

<https://github.com/araemon/AndroidExercise/tree/master/TryFragment/app/src/main/java/com/pppppp/tryfragment>

トランザクションについては以下。

<https://qiita.com/tomo1139/items/62902093d1850085742d>

DB

19-10-05

Android Kotlin Extensionのおかげで、ButterKnifer はいらなくなったっぽい。

今回も、SQLite を使用する。

Anko を使うと良いらしい。

<https://qiita.com/naoi/items/6e6152c90c4219bbb055>

<https://github.com/Kotlin/anko/wiki/Anko-SQLite>

<https://hirauchi-genta.com/kotlin-sqlite/>

これ見てコードを綺麗にした。最初、ID って？と思ったけど、SQLite だと自動的に振られてインクリメントされていくらしい。

あれ？タップしてリストを削除して DB から削除するのってどうするの？

DB から削除して、DB を再読み込みしてリストに反映している例があった。

- ID なら一意に定まるのだけど、リストには乗っていないから取得できない？

おそらく、次のやり方がある。

1. 隠しパラメータ

透明な感じで id の値だけ保存できれば、それを取得して消せばよい。

2. リストの情報を使って削除する

同じデータが入ってくると NG だけど、全部を合わせると一意に定まるのならば、これらを使って消すことは可能。

状態遷移のことも考えると、1の方法の方がよさそう。

割勘魔人の時は、..

- 分かった。リストにはアダプタが紐づいていて、アダプタに選択している Idx を渡せば、元のリスト (ListData) を照会できる。

adb shell で、SQLite で作った DB を覗く。

```
C:¥WINDOWS¥system32>C:¥Users¥User¥AppData¥Local¥Android¥Sdk¥platform-tools¥adb
shell
generic_x86:/ $ run-as psycho.mountain.tastinggenie
generic_x86:/data/data/psycho.mountain.tastinggenie $ ls
cache databases files
generic_x86:/data/data/psycho.mountain.tastinggenie $ cd databases
generic_x86:/data/data/psycho.mountain.tastinggenie/databases $ ls
personal.db personal.db-journal
generic_x86:/data/data/psycho.mountain.tastinggenie/databases $ sqlite3
SQLite version 3.9.2 2017-07-21 07:45:23
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open personal.db
sqlite> .table
android_metadata  personal
sqlite> .table personal
personal
sqlite>
```

run-as でパッケージの権限を取得するのがコツ。

DB がおかしかったら、コンソールから消してあげればよい。

DB で画像を扱う (カメラとファイル選択)

<http://blog.livedoor.jp/singudea/archives/13423736.html>

画像の管理方法をどうしようかと思う。直接 DB に画像を blob として保存するのではなくて、ファイルの場所と名前だけを保存する方法もあるけど・・・？

画像管理ソフトじゃないから、そんなに画像数がない、ということを考えて直接 Blob で保存してみる・・・

<https://stackoverflow.com/questions/47674672/kotlin-save-image-to-database>

DB の Blob 形式と、表示のための drawable の変換が面倒。

drawable -> bytearray -> blob

blob -> bytearray -> bitmap -> drawable
という操作が必要.

やってみたら, parse で死んでしまう.

org.jetbrains.anko.AnkoException: Can't initialize object parser for
psycho.mountain.tastinggenie.ListData, no acceptable constructors found

クラスを blob にしても bytearray にしてもダメっぽい.

<https://discuss.kotlinlang.org/t/blob-fields-in-anko-sqlite/4603/2>

classParse だと blob が定義されていない? 自分で parser を組めば大丈夫?

- MapRowParser を定義し, image は as Blob で取得してあげたら落ちなくなった.

画像追加のために, Fragment を変更する.

<https://sites.google.com/site/technoute/android/icon/drawable>

これをクリックして選ばせる.

<https://android.keicode.com/basics/multimedia-choose-pictures.php>

Intent は Activity に追加してあげたほうが良い? Fragment で完結できるならば, そうしようと思うけれど. まずは, 記事をなぞる.

DB の id, image は, Integer, Blob のはずだが, Long, ByteArray になっている.

Long -> Int にして, Blob の代わりに ByteArray にする.

出来たのだけど, メモリが足りないとかで落ちる. 圧縮する必要があるかも.

```
/art: Alloc concurrent mark sweep GC freed 3(72B) AllocSpace objects, 0(0B) LOS objects, 5%
free, 73MB/77MB, paused 1.030ms total 32.492ms
W/art: Throwing OutOfMemoryError "Failed to allocate a 63967892 byte allocation with 4193848
free bytes and 54MB until OOM"
D/AndroidRuntime: Shutting down VM
E/AndroidRuntime: FATAL EXCEPTION: main
    Process: psycho.mountain.tastinggenie, PID: 28815
    java.lang.OutOfMemoryError: Failed to allocate a 63967892 byte allocation with 4193848
free bytes and 54MB until OOM
        at dalvik.system.VMRuntime.newNonMovableArray(Native Method)
        at android.graphics.BitmapFactory.nativeDecodeByteArray(Native Method)
        at android.graphics.BitmapFactory.decodeByteArray(BitmapFactory.java:561)
        at android.graphics.BitmapFactory.decodeByteArray(BitmapFactory.java:591)
```

読み込み時点で画像を圧縮 (リサイズ) することで解決.

ただ, 幅と高さを固定で設定するだけだと画像が歪んでダメ.

カメラ操作

カメラが選択されない・・・？

<https://qiita.com/yukiyamadajp/items/137d15a4e65ed2308787>

カメラが選択されたとしても、うまく画像を取得できない。

自分でカメラの処理を実装するっていう方法もあるっぽい。

<https://qiita.com/nomunomu0504/items/05bb7aefcb6a04a8f6fc>

19-10-16

- まずは最小限の労力で PoC を作る。

Intent でカメラ or ファイル選択をすることができた。

https://qiita.com/furusin_oriver/items/b6875f0703dbd49e7737

19-10-20

- そういえば、画像管理のこの実装方法って拡張性が低い

例えば、正面から撮った写真、裏のラベル、というように画像を複数登録したい場合が考えられる。

ということで、blob とし保存するのではなく、次のように画像を管理する。

- 写真を撮影する
画像フォルダを作ってそこに保存する。画像サイズを大きくなりすぎないように圧縮する。
- ファイルを選択する
選択したファイルを画像フォルダにコピーする。
- 画像ファイルは csv で管理する
例えば、photo1.jpg,photo2.jpg のような感じ。

流れは次のイメージ。

写真を撮影 or ファイル選択 → ImageView に設定

DB の保存 → 写真をフォルダに保存

だから、Activity では画像を ImageView に設定できれば良い。

縮小とか

<http://y-anz-m.blogspot.com/2012/08/android.html>

メモリがあふれるので、画像は元の画質で保存したい。でも、ImageView にリサイズして設定しちゃうともう元の画像は手に入らないっぽい。

それならば、カメラ画像はとりあえず保存しつつ表示して、ドキュメントからの画像は表示だけにする。そして、元の URI をリストに保存しておいて、DB になかったら追加しつつ、ドキュメントの画像はコピーしてあげればよいかな？

DB 構造

どうするかをちゃんと考える。

static な情報と dynamic な情報に分ける

リレーション名	ID [主キー]	酒の名前	酒類	分類	画像	酒造	酒造2	精米歩合
SakeList	id	name	grade	type	image	maker	prefecture	pol_rate
		1 越の誉	純米吟醸	ひやおろし	1.jpg,2.jpg	原酒造株式会社	新潟県	60%
		2 男山	純米	生酛	3.jpg			

リレーション	Review ID [主キー]	ID [relation]	日付	温度	色	上立ち香	基調香	口中香
SakeReview	review_id	id	date	temperature	color	scent1	scent2	scent3
	1	1	2019/10/20	3	1	1,2,4,6	1,3	1,2,3,4
	2	1	2019/10/22	3	1	1,2,4,6	1,3	1,2,3,4

DB を使うたびに helper を宣言するのはメモリの無駄。そこで、activity や fragment ごとに DB の使い分けを考える。

リスト表示

Recycler view が良いと思う。ListView だけでも行けそうだけど、なんとなく。

<https://qiita.com/Todate/items/297bc3e4d0f3d2477ed3>

FAB ボタンってやつを使って、モダンな感じにしたい。

- CoordinatorLayout にしないと、FabButton が表示されなかった。

あとは特に問題なし。

ダイアログ

<https://qiita.com/hokutonikukyu/items/ebbc0318c07035e04bf7>

酒類の選択のためにカスタムダイアログを作りたい。

分類とリストを受け取って、動的にダイアログを生成してほしい。

だけど、adapter とか使う必要があって、けっこう面倒くさそう。

<https://medium.com/@makkenasrinivasarao1/android-custom-dialog-with-list-of-items-ba1ab0e78e16>

これは、都道府県の選択にも応用できそう

TODO: 見栄えは後回し。

- 実装完了

Activity と Fragment

使い分けが難しい。とりあえず今回は、DB のテーブルで Activity を切り分ける。

List, Information: SakeList

Detailed, Review: SakeReview

DBOpenHelper は最初の MainActivity で持っておいて、ListActivity, ReviewActivity に引数として渡すことで、メモリをちょっとケチる。(←これって本当?)

- Activity の遷移が遅い！

この Activity パツって切り替わる感じ、外部アプリを起動したり、GoogleMap が出たりする処理に似ている。つまり、大体のアプリって、1Activity で実装されていたりしない？

今回のアプリでは、1Activity で実装して、実装しきれなくなってきたら Activity を分けられるように工夫することにする。

addToBackStack()をしなければ戻るときに一つ飛ばせると思ったけど、なぜか戻る前の TextView が残ったままになっている。

<https://symfoware.blog.fc2.com/blog-entry-2037.html>

- pop して addStack することで解決。

imageView が正しくセットされない！！！！

今、遷移が

A -> B -> C (sake information) -> D -> B -> C

となっているが、初回の B->C では imageView にセットできるのに、1 回セットすると 2 度とセットできなくなる。

19-10-26

listener 関数に引数として imageView を渡し、Activity 側のメンバ変数として処理させることで解決。

SakeList を作っていく

URI <-> String

<https://qiita.com/superman9387/items/56ef1bd3f98f5eab5244>

情報の渡し方

Activity -> Activity: Intent

Activity -> Fragment: Bundle

Fragment -> Activity: Listener

DBManager を Activity が保持し、それぞれの Fragment に getList した値をあげたり、それぞれの Fragment から受け取った List を insert したりする。これで省メモリになると思う。

データクラスを Parcelable にする必要がある。kotlinx に Parcelize があるようなので、それを使う。

<http://tech.furyu.jp/blog/?p=6378>

画像ファイル削除

<https://erikori.com/blog/2015/06/13/android%E9%96%8B%E7%99%BA%E5%89%8A%E9%99%A4%E3%81%97%E3%81%9F%E7%94%BB%E5%83%8F%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E3%82%92%E3%82%AE%E3%83%A3%E3%83%A9%E3%83%AA%E3%83%BC%E3%81%8B%E3%82%89%E5%89%8A/>

次にやること:

- DB の更新処理
- DB の削除処理
- SakeDetails の作成

DB の更新処理と削除処理

19-10-27

RecyclerViewHolder と RecyclerViewAdapter を触って, LongClick を追加.

これで削除処理を実装する.

View の変更が反映されない?

A -> B (sake information) -> C -> A

と遷移して List である A に戻ると, B で追加した情報が反映されない. これは, B を更新した時も同様.

<https://qiita.com/koujin/items/0bddffdab8f282a3cc6b>

これに書いてある. 再代入したものを変更しても意味がないという話. そりゃそうだ.

課題:

- SakeReviewFragment の実装
- SakeListFragment で毎回 DB にアクセスしているから遅い
- 写真を参照するとき, もともと Picture に入っているものを参照すると, 消したときに落ちる

Review の実装

DatePickerDialog の実装. Kotlin だと? 面倒?

<https://qiita.com/sass9872/items/b01d024f52249f79f316>

<https://www.javadrive.jp/start/calendar/index2.html>

- 普通に実装できた.

ただ, テキストをパースして Date->Calendar がちょっとめんどい.

<https://hacknote.jp/archives/24285/>

テキストのタッチ

http://blog.livedoor.jp/shizuku_kun/archives/51362627.html

MEMO

インクリメンタルサーチ. これ面白そう.

<https://qiita.com/Xdaichi/items/e6234b336219120a76e4>

RatingBar

<https://www.google.com/search?client=firefox-b-d&sxsrf=ACYBGNTtZsrcRI3Vr-Rk7Q9Hhy8H6Xs3kg%3A1571747593664&ei=CfeuXc-NKPaQr7wPqeqh->

[Ak&q=android+rating+bar+kotlin&oq=android+rating+bar+kotlin&gs_l=psy-ab.3..0i8i30.5151.6461..6673...0.0..0.114.680.4j3.....0....1..gws-wiz.....0i203j0i4i30j0i8i4i30.FPZSBtEdN1I&ved=0ahUKEwiPtyH8K_IAhV2yIsBHSl1CJ8Q4dUDCAo&uact=5](https://play.google.com/store/apps/details?id=com.example.myapplication&hl=ja&gl=US)

DB がおかしい

renew すると落ちる。以後、起動するたびにパーサーのエラーで落ちる。

- 要素を変更した後に、ちゃんと DBO を作っていなかった。

全体的に、データの流れを洗う必要がある。今は無駄に DB に問い合わせている気がする。DB を触るのは、初回と更新があった時だけにとどめたい。

19-11-11

Activity でメンバ変数として MutableList を持っていて、それを Bundle として渡す。

やってみてわかってけど、これは、参照渡し。Fragment 側で削除要請を出して、Activity 側で削除。その後 Fragment でも removeAt で削除すると、二つ消えることになる。注意。

- 全然高速化しない！

重たい理由は、写真だ！

写真はオリジナルを保存しつつ、基本は圧縮して保存。この辺は設定にしようと思う。

設定を共有する仕組みを作らないといけない。

<https://maku77.github.io/android/fw/shared-preference.html>
<https://qiita.com/irgaly/items/4b149068b2f62289e7a5>

これすごい。自分でファイルに書き込んでパースしようと思ったけど、いらない。そもそも SharedPreferences とはアプリ設定を XML ファイルで管理する仕組みだから。多分、割勘魔人も使っている -> 使っていた。

<http://319ring.net/blog/archives/1504/>

画像のリサイズ方法

汚いながらも、圧縮と無圧縮を分けて保存できた。任意のディレクトリを指定する方法が課題。

- カメラ画像が落ちる！

```

val bos = FileOutputStream(dstFile)
try {
    val baos = ByteArrayOutputStream()

    val originalBitmap: Bitmap =
MediaStore.Images.Media.getBitmap(context.contentResolver, srcUri)
    val bmp = resizeBitmapToDisplaySize(originalBitmap, context)
    bmp.compress(Bitmap.CompressFormat.JPEG, 100, baos)
    bos.write(baos.toByteArray())
    bos.close()
} catch (e: IOException) {
    e.printStackTrace()
} finally {
    bos.close()
}
}

```

ここが悪さをしている。

カメラ画像の場合、src と dst が一緒になるから、src で bmp にした後に dst をオープンしないとまずい。

これで解決。

雑多な問題解決

コンテンツプロバイダ

無圧縮画像を登録させたい。

<https://akira-watson.com/android/external-storage-image.html>

ExpandableList

<https://tutorialwing.com/android-expandablelistview-using-kotlin-example/>

<https://stackoverflow.com/questions/6643625/is-it-possible-to-create-an-expandable-list-alertdialog>

解決。結構面倒だった。

ExpandableCheckList

<https://stackoverflow.com/questions/33301708/how-to-get-checkbox-state-in-expandable-listview-in-android>

アダプタをゴリゴリ実装した.

layout を角丸に

<https://qiita.com/masaibar/items/3df3ebbb981cd274fe78>

参考

- [1] “木村克己”, 日本酒の教科書, 新星出版社, 2010
- [2] 利き酒（ききざけ）のコツ
<http://www.kamofuku.co.jp/hissyohou/hissyohou.htm>
- [3] 日本酒テイスティングの訓練方法
<http://日本酒.biz/category20/entry204.html>
- [4] 利き酒のやり方
<http://wakabajirushi.fc2web.com/kikisake.html>
- [5] KURAND 「これであなたも利き酒師気分！日本酒を表現するための言葉（色・香り・味）」
<https://kurand.jp/blog/sake-taste-smell-color-kikizake/>
- [6] 極私的なワインテイスティングノート「日本酒テイスティングシートを作ってみました」
<http://tnishino.blogspot.com/2015/01/blog-post.html>
- [7] ネタ酒場研究会「テイスティングノートのススメ」
http://www.netasakaba.net/?page_id=5232
- [8] 灘酒研究会
<http://www.nada-ken.com/main/>
- [9]