

H30Fenrir_簡易仕様書

仕様

- 作者 :YukiOtake
- アプリ名 :RightNow(旧:H30Fenrir)
- URL :開発者アカウントを持っていないため公開できない
- git : https://github.com/Yuki-Otk/H30Fenrir
- 対象OS :Android
- 対象Ver :Android 4.4以上
- 開発環境 :Android Studio 3.0.1
- 開発言語 :Java (java version "9.0.1")
- テスト端末(実機): Nexus 5(Android 6.0.1 API 23)
- テスト端末(エミュレータ):Pixel 2(Android 8.0.0 API26)

機能一覧

大機能名	中機能名	機能概要
検索	レストランを検索	ぐるなびAPIと端末のGPSを使用し現在地周辺レストランを検索する
検索	絞り込み検索	フリーワードを入力することにより検索結果を絞り込むことができる
検索	絞り込み検索	チェックボックスを選択することにより検索結果を絞り込むことができる
表示	レストランの詳細確認	検索結果から詳細を詳細を読み込み表示する
表示	絞り込み表示	検索結果を営業時間から更に絞り込むことができる
次へ	次のステップにつなげる	詳細画面から電話やホームページを見たり共有することができる
省エネ	制限モード	通信量を最小に抑えることができる

画面一覧

- app/src/main/res/ にレイアウトとアクションバーに表示するためのメニューが配置されている

画面名	画面概要	ファイル名
パーミッション取得	位置情報の許可をもらう画面	./layout/activity_main.xml
検索	検索する際のパラメータを設定	./layout/activity_location.xml
制限モード	制限モードにするためのボタン	./menu/main_menu.xml
検索結果表示	検索した結果を表示	./layout/activity_list_show.xml
検索結果表示	検索結果の1行分のカスタムレイアウト	./layout/format_row.xml
詳細	検索結果から選んだ店舗情報を表示	./layout/activity_details_show_scroll.xml
次のステップ	詳細画面に出ている店舗に電話したり情報をシェアする	./menu/details_menu.xml

クラス構成

- app/src/main/java/com/example/otkyu/h30fenrir/ に 直下にActivity class、更にmodel等にほかのclassが配置されている

クラス名	概要	ファイル名 (java)
MainActivity	位置情報のパーミッションを取得・チェックする	./MainActivity
LocationActivity	検索パラメータを入力	./LocationActivity
ShowListActivity	検索結果一覧を表示	./ShowListActivity
ShowDetailsActivity	検索結果詳細を表示	./ShowDetailsActivity
GnaviAPI	ぐるなびAPI呼び出しとデータパース	./asynchronous/api/GnaviAPI
GnaviRequestEntity	ぐるなびAPIリクエストデータモデル	./asynchronous/api/model/GnaviRequestEntity
GnaviResultEntity	ぐるなびAPIレスポンスデータモデル	./asynchronous/api/model/GnaviResultEntity
ImgAsyncTaskHttpRequest	画像を取得する非同期処理	./asynchronous/img/ImgAsyncTaskHttpRequest
ChangeModel	文字列を変換するクラス	./model/ChangeModel
CheckModel	文字列等を判定するクラス	./model/CheckModel
CasarealRecyclerViewAdapter	検索結果一覧Adapter	./view/CasarealRecyclerViewAdapter

使用したAPI

- ぐるなびAPI

コンセプト

- 「今」どこに行きたいか決めるアプリケーション。
- RightNowの名前のように今まさに、自分が調べたかったお店、正解のお店を知ることができる。

ターゲット

- 都内など「徒歩」で行動する人。ぐるなびAPIを使用する特性上、どうしても店舗が固まっている方が使いやすいため都市部をターゲットとする。
- また、都市部ということで徒歩で行動する人が大半であるため、徒歩のユーザーにスポットを当てて開発を行った。

こだわりのポイント

インターンシップ時

- ソースコードの参考にした部分をmemoして置き、あとからなぜこのようなコードにしたのか思い出せるようにした。
- listの中にオブジェクトを埋め込むことによりC言語でいう構造体のように複数の形を入れ検索結果や検索する際のパラメータを扱いしやすくした。
- なるべくクラッシュしないような設計にした(pagingの際に「前のページ」や「次のページ」のボタンを押せなくする)
- 詳細画面のところはコピーが行えるような設定にした。
- 詳細画面をスクロールビューにすることで拡張性を高めた。
- githubをこまめに更新して途中でローカルプロジェクトが破損しても被害が最小限でとどまるようにした。

選考時

- 徒歩で行動するため表示を何分圏内かをメインに表示した。
 - 100m圏内がどのくらい行きやすいのわからないため
- 店を検索する際に気になる店を発見できても営業時間でなければ入ることができないため「開店中」かどうか分かるように実装した。
 - 更に昼(12時)と夜(18時)に営業しているかどうかもわかる。
- 通信量を節約できる制限モードを実装
 - 外での利用がメインターゲットにしているため通信量制限(ギガが減る)問題を軽減するために画像の読み込み等をキャンセルする「制限モード」を実装した。
- コンセプトに合わせてあえて場所を指定する検索は実装を見送った

アピールポイント

- 営業時間の取得
 - 今回のメインの開発は「営業時間の抜き出し」となった。各店舗の営業時間のフォーマットがバラバラだったため文字列の取り扱いには苦戦した。何度かアルゴリズムを変更して今回の実装に落ち着いた。曜日の判定を今回は見送った。
- JSONのバース方法を変更
 - JSONの解析方法を変更し自らが管理しやすいようにした。
- インターンシップでアドバイスを元に積極的にメソッドを追加した。(1メソッドは1画面に収まるように)
- 画像を1度に複数読み込む
 - 非同期処理の部分がある程度理解し複数枚読み込めるように実装できた。(リファレンスやデバックツールを教わって)
- キーワード指定の方法が2つ
 - キーワード指定の方法がテキストボックスに入れる方法にプラスしてチェックボックスを用意した。複数の指定方法を用意して使いやすいほうを使用してもらう。
- 友人に使用してもらってレビュー
 - 開発者は自身でボタンを配置しているためわかるが、初見の人がどれくらい予想通りに使用してくれるのかを知るため
- GithubのIssuesを活用しタスクを忘れずに生産性を維持できたと考える。

実装予定(実装しなかった)機能

機能名	機能概要	理由
曜日判定	営業時間の曜日部分も抜き出す	正確な営業時間を提供しなかったため
チェックボックスのカスタム	検索時にしているチェックボックスを各々でカスタムできる	個人によってよく使用する検索値は異なるため
全検索結果の読み込み	一度の検索で検索結果をすべて取得する	時間の絞り込む際に現状では表示されているページでしかできないためすべての検索結果で絞り込むのが理想
プログレスバーの実装	APIを呼び出し時に発生する待ち時間にプログレスバーを表示する	フリーズしているのか通信をしている待ち時間なのか分からないため