Finding Yummy：企画書（Ruby）

高月　晴輝

## コンセプト・目的

・大手サイトとの差別化（乗せる情報は必要最低限、ページもシンプル、出先でスマホで入力無しで！！）

・スマホサイズで崩壊しないレイアウト設計

・申し訳程度のマネタイズ

## サイトの特徴

余計な事は一切せず現在地から近いお店を知りたい、今外に居て直ぐお店を調べたい、に寄り添ったサイト。

検索結果一覧で確認できるのは距離と値段とやっているかだけ。

他サイトとよりレビュー価値を上げる為課金ポイントでしかレビューが出来ない（予定：お店からお金貰えない上に広告付けて画面をごっちゃごちゃさせたく無いけど、一応サービスとして提供するからにはどっかしらでマネタイズしないといけないから無理くり考えただけです）

## リポジトリURL

## 開発環境/言語

Visual Studio Codeバージョン: 1.86.1

OS: Windows\_NT x64 10.0.19045

・Ruby 3.2.2

## 動作対象端末・OS

Visual Studio Codeでしか試していないので何とも言えませんが、ウィンドウサイズだけで言えば、google chrom開発者ツールで見れる  
pc/ノートパソコン/スマートフォンサイズには対応しています、タブレットサイズだけやや微妙です

## 開発期間

2/5～2/11

# 機能一覧

・現在地の取得：緯度経度で取得

・現在地からの検索

・検索結果一覧：ソート、地図を開く、店舗URL、ページング

・店舗詳細画面

・ブックマーク

・ポイント評価

・ログイン

コンセプト的にログインの必要性が無いため、ヘッダーのマイページをわざわざ押さない限りログイン画面には入れません、ブックマークも表示されません

・ログアウト

・マイページ

・ユーザー情報編集・閲覧

・ポイントの獲得

・ブックマーク履歴の参照

# 画面一覧

・検索方法選択画面

・現在地取得＆距離設定画面

・検索結果一覧画面

・店舗詳細画面

・マイページ画面

・ログイン画面関連（railsのdeviceを95割流用）

# 使用API・ライブラリ等

ホットペッパーグルメサーチAPI

bcrypt：パスワードのハッシュ化で使用。

devise：ユーザー認証機能で使用。

dotenv-rails：.envファイルから環境変数をロードする為に使用。

geocoder：現在位置取得・店舗との距離を計算する為に使用。

## テーブル定義(ER図)などの設計ドキュメント

これじゃ無かったらごめんなさい。

超簡易コントローラ、業務フロー図

<https://gitmind.com/app/docs/bicboa56>

Table定義はエクセルファイルに記述しています

ApplicationController

after\_sign\_in\_path\_for:ログイン後のリダイレクト指定

configure\_permitted\_parameters：ユーザー登録時にusernameの登録を許可する

HomeController

index: 検索方法選択画面

SearchController

index: 検索結果表示画面

new: 現在位置を取得・検査距離を指定するページ

show: 店舗詳細画面

MypageController

show: マイページ

edit and update: ユーザー情報の編集

new: ブックマークの登録

destroy:ブックマークの消去

create:レビュー（ポイント）を付ける

RegistrationsController　を始めとしたdeviceを使う事で作られるコントローラがいくつか

## 未機能一覧

・検索方法選択画面：

別の検索方法の追加：Pythonの方で作った上、そもそも今回求められていなかったので後回しになった

・検索結果一覧画面：

画面リロードした時のポジションの固定：検索結果一覧結果の横に、ブックマークボタンを作ったが、ブックマークボタンを押す度に画面上部に戻る為消した。

今店が開いているかを一目で表示：言語処理を考える、製作する余裕が無かった。コンセプト的に絶対に実装したかったが求められていた訳では無いので断念。

・マイページ：

レビュー履歴：ブックマーク履歴とほぼ変わらない為後回した

・課金システム：ポイントを実際に購入する画面、実際に課金してちゃんと動作するか試す必要があった為作っていない。

## 未検証

・家の周りがド田舎な為、登録されていた店舗が合計八店舗しか無かった為、ソート機能や店舗一覧画面は重大なバグが残っている可能性があります

・サイトから飛べる全ページにホットペッパーAPIの文を入れたつもりですが、抜けがあるかもしれません。入れる意思は強くありました。

# アピールポイント

・二月五日に軽い気持ちで採用ページを見たら「Ruby」と書いてあったので慌てて作り直した。

・全て未経験から初めて、約一週間で動いている。（ruby,ruby on rails）

・もし何かの間違いでこのサービスが流行ったとしても、一応マネタイズ方法がある

・出先、スマホ画面でサッと店を調べる、にそぐわない機能は入れていない。

・身の程を弁えた機能。技術的に難しい事はしない、もう既に大手サイトが存在するので昨日や目的は尖らせる、ソート機能を犠牲にDB情報を減らしたので多分サーバー代も安い、コンセプトとも会わないし、ユーザーが大量に付く訳ないサイトなので人数が居ないと成立しない写真投稿やコメントといった機能も排除。

・検索結果一覧から地図に飛べる、機能的には簡単だけど普通にこのサイトを使う時は多分便利

## デザインアピールポイント

・スマホ画面で見れる

・目に優しい

・ごちゃごちゃしてない

・基本入力が要らない

# 自己評価

・プロ相手におこがましいし、過大評価だとも思うが、めっちゃ頑張ったしそれなりの物が出来たと思う。

・webサイトとwebアプリケーションの違いすら怪しかった二週間前の自分に自分のサイトを自慢したい程度には満足している。

・肝心な店が開いているか〇か×で分かる機能が未実装・そもそも未経験で技術的アピールが出来ていない、を除けば最初っから最後までコンセプトを前提とした開発ができて作品作り的に満足している。就活的には微妙。

・デザイン面は散々注意してた身の丈や期限を考えず、つい楽しくなって無駄に一ページだけ凝ってしまったので反省している。次は全部のページを似たようなクオリティにしたい。

・検索結果一覧のちょっとグレーっぽい白色本当に好き、ポイントカラーの赤と青の色もちょっと暗くて目が痛くならないのに雰囲気会っていて好き。本当にこんな所に時間をかけるなら別のページ作ればよかった。

# アドバイス

## 技術的

言語処理についてです  
月～木、祝日、祝前日: 17:00～22:00 （料理L.O. 21:30 ドリンクL.O. 21:30）金、土: 17:00～23:00 （料理L.O. 22:30 ドリンクL.O. 22:30）等の文を  
月なら1、木なら4、

みたいに事前に使われそうなワードを処理しやすい文字と数字に変換した後、単語ごとにリスト化してif文を使って一個一個判定していくパターンと  
今回途中で書くのを諦めた正規表現を抜き出すパターン、どちらを使ってアプローチするべきでしたか？

DBを企業として運用していく規模感が分からず極力使わない様に作りましたが、絶対にホームページを乗せる以上契約する物なので惜しまず使う物なのか、極力抑えるべき物なのか、全体の風潮としてはどちらが主流ですか？

## デザイン面

料理の写真を載せる、予約の為のカレンダー、を表示しない場合店舗詳細画面はどうゴージャスにするべきだったでしょうか。APIから引っ張れる情報も少なくとても困りました。