

簡易仕様書

アプリ名： NearShoku - ニア食

- 対象OSおよびブラウザ/バージョン(テスト基準)

Windows10 /IOS10.0以上

GoogleChrome(123.0, 122.0), FireFox(124.3)/Safari

- 開発環境/言語

IDE: Pycharm

language：Python3.10/JavaScript/HTML/CSS

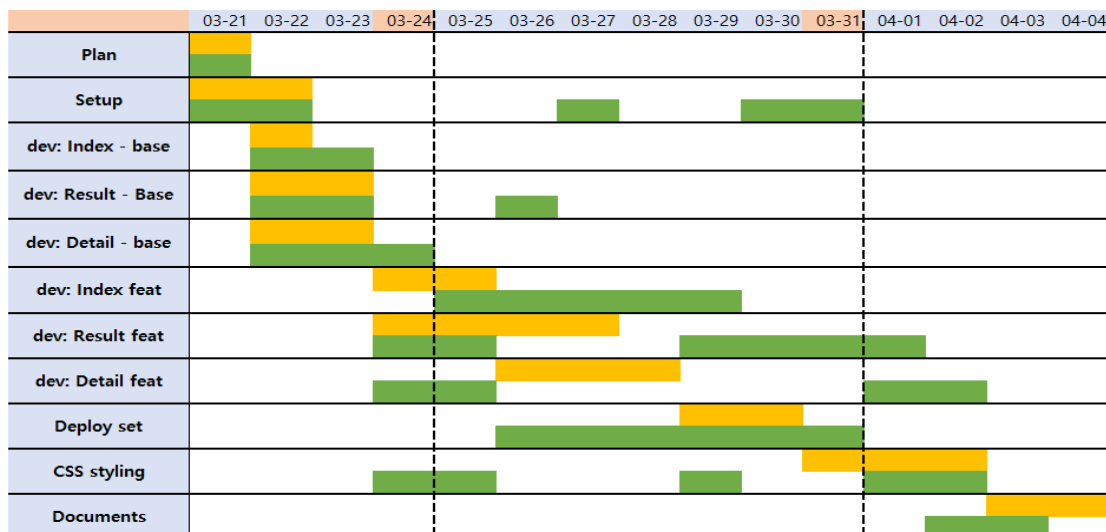
廃部環境：GCP-Server/Linux CentOS 9/Docker/NGINX/

- フレームワーク

Django 5.3.0：Full-Stackフレームワークとして全体的な機能を開発しました。

Bootstrap 5.3.3：CSSに加えて全体的なデザインを開発しました。

- 開発期間



黄色は計画で、緑色は実際にかかった時間を表示しました。

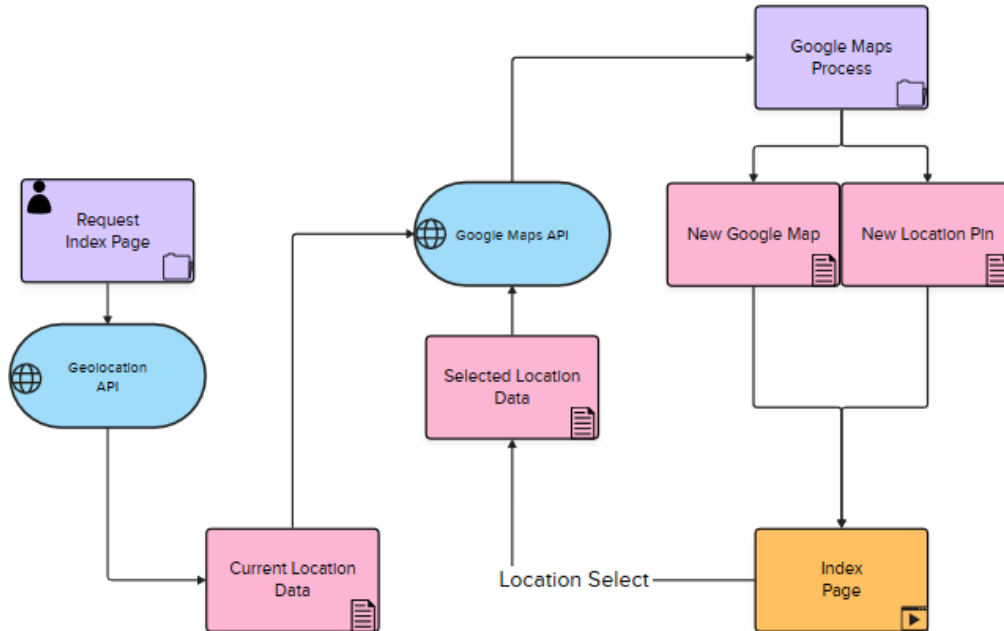
- 機能概要

- タイトル： Indexに移動
- Index： 情報検索条件選択
 - 検索条件①： 範囲（Range）
 - 検索条件②： 検索結果整列基準（Order）
 - 現在位置基盤検索
 - 選択位置基盤検索
 - ◆ 位置選択: GoogleMaps For Javascript API
 - ◆ 位置選択後検索ボタンの有効化
- Result： 検索結果整出力
 - エラーページ出力
 - ◆ APIによる検索結果がない場合：No Search Result
 - ◆ 間違った要請を受けた場合：Bad Request
 - 検索結果出力
 - ◆ ページング： ページごとに最大10個の結果出力
 - ◆ お店の情報
 - 店名
 - サムネイル
 - ジャンル
 - アクセス
 - ◆ 詳細情報ページへ移動： 情報をクリックすると移動する
- Detail： 詳細情報ページ
 - Resultに戻るボタン
 - 詳細情報出力
 - ◆ 店名
 - ◆ ジャンル
 - ◆ 写真
 - ◆ 営業時間
 - ◆ アクセス
 - ◆ 位置マップ: Static GoogleMaps イメージ
 - ◆ 住所

- 設計ドキュメント

- result_index

- Data-Flow-Diagram



- Index 基礎設計図

Near Shoku

title

current location : ~ by Geolocation

range option

[search]

select location : country . state

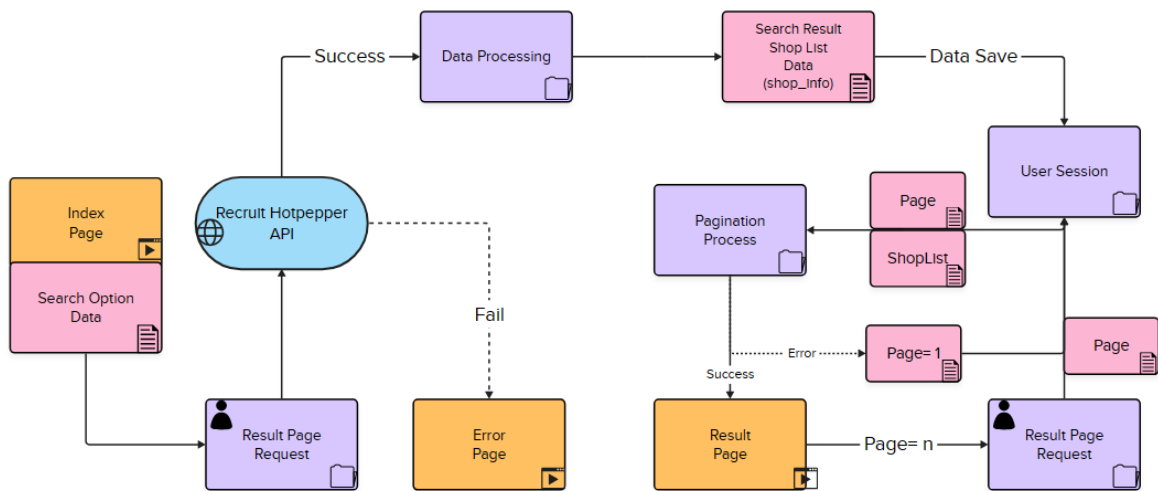
city . street

[search]

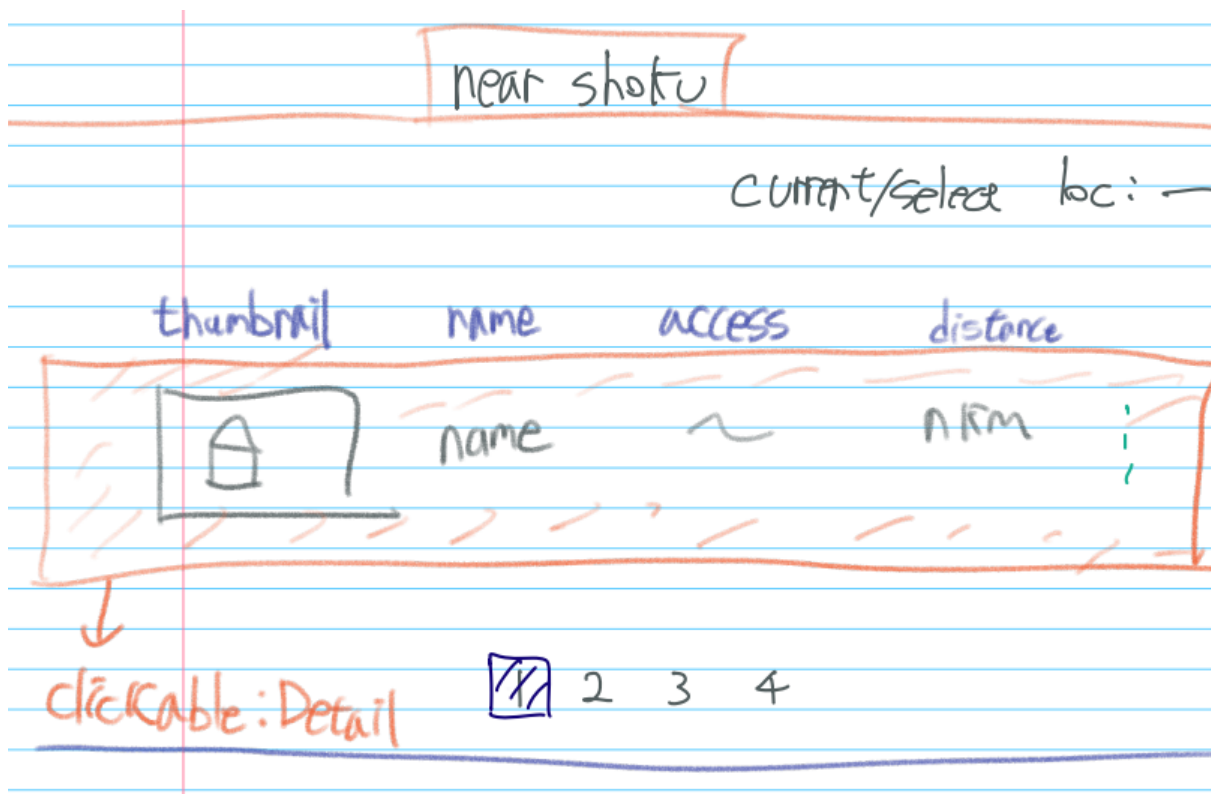
Select from map.

- Retail

- Data-Flow-Diagram

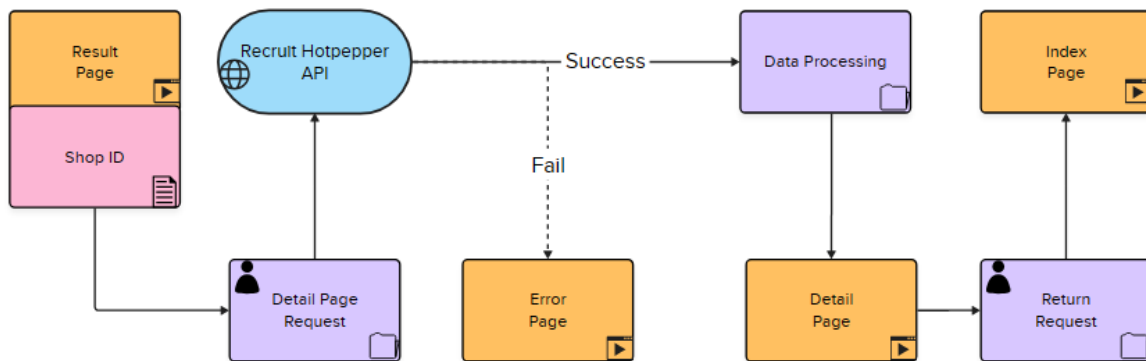


- Result 基礎設計図

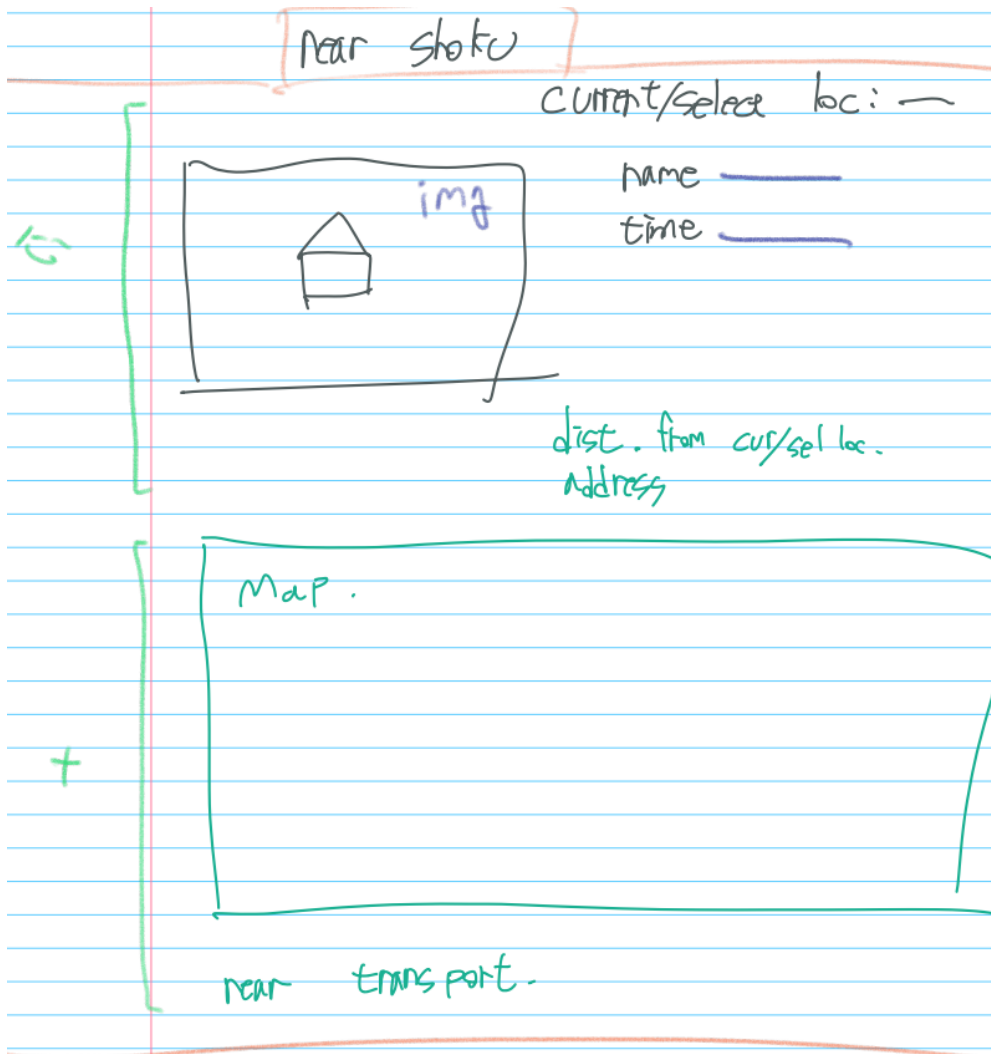


- Detail

- Data-Flow-Diagram

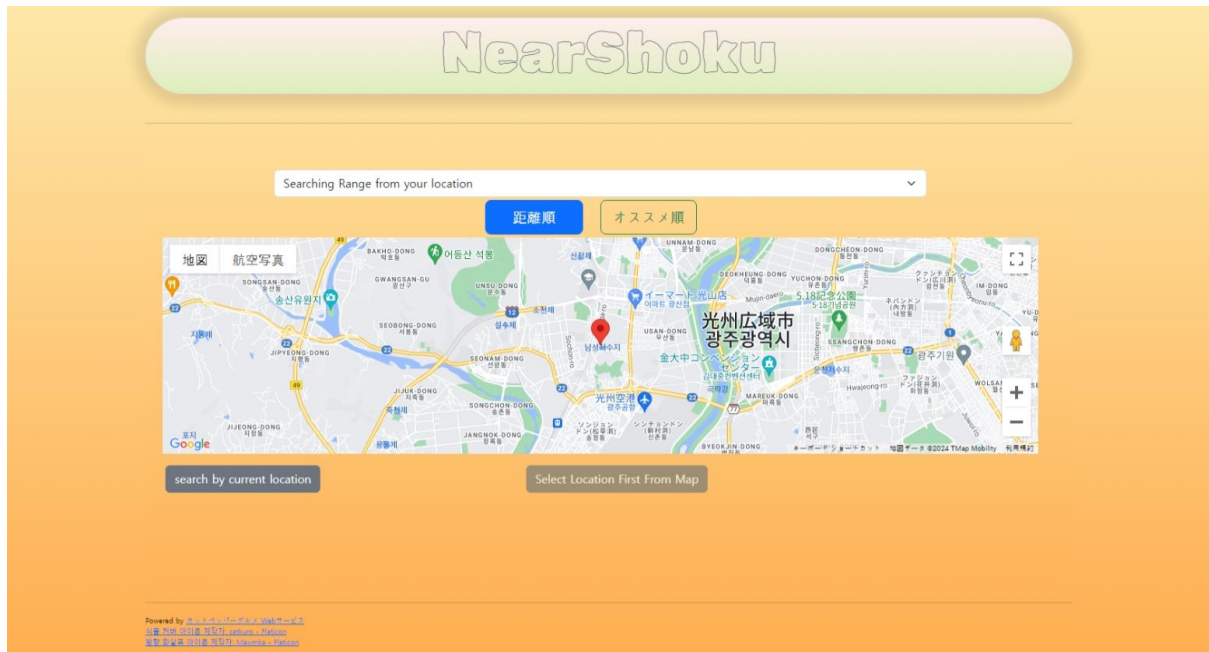


- Result 基礎設計図



- 実際サービス画面 <https://op6160.xyz>

- Index



- Result



■ Detail

NearShoku

戻る

やきにくどらひらのてん

焼肉どら 平野店

焼肉
おまかせ



営業時間

月～水、金～日、祝日、祝前日:
17:00～22:30 (料理L.O. 22:00 ドリンクL.O. 22:00)

→ J R 関西本線平野(J R)駅南口より徒歩約8分
→ J R 関西本線東部市場前駅より徒歩約13分



大阪府大阪市平野区平野馬場 2-3-32 1 階

Powered by [NearShoku](#) [NearShoku](#) [NearShoku](#)
Copyright © 2024 NearShoku. All Rights Reserved.
近所食 近所食 近所食 近所食 近所食 近所食 近所食 近所食 近所食 近所食

● コンセプト

ユーザーの現在地、またはユーザーが希望する場所から近くの飲食店を探索し、情報を提供するサービス

- こだわったポイント

- 現在地だけでなくGoogle Maps APIを追加で活用したため、ユーザーから探索位置の選択が可能です。
- Recruit Hotpepper APIからもらった店の情報を活用して、課題の必須要求事項以外にも自らが飲食店を探す時にあれば良いと思う追加情報を提供するように気を使いました。
- ユーザーの利便性と可視性を向上させるためにデータ処理を追加しました。

- ‘営業時間’の場合

```
'open': '月、火、日、祝日: 12:00 ~ 22:00 (料理 L.O. 21:30 ドリンク L.O. 21:30) 水、木: 12:00 ~ 23:00 (料理 L.O. 22:00 ドリンク L.O. 22:30) 金、土、祝前日: 11:30 ~ 23:00 (料理 L.O. 22:30 ドリンク L.O. 22:30) '
```

このような形式のデータを

営業時間

月～金:

11:00～14:30 (料理L.O. 14:30 ドリンクL.O. 14:30)

17:00～23:00 (料理L.O. 22:15 ドリンクL.O. 22:30)

土、日:

17:00～23:00 (料理L.O. 22:15 ドリンクL.O. 22:30)

上のイメージのように区分する方法を使ってみました。

- ‘アクセス’の場合

→ JR京橋駅北口 徒歩5分
→ 京阪本線京橋駅 徒歩5分
→ 地下鉄長堀鶴見緑地線 京橋駅4番出口 徒歩2分

→ JR東京駅八重洲南口 徒歩1分
→ メトロ銀座線京橋駅 徒歩3分
→ メトロ有楽町線銀座一丁目駅 徒歩5分

複数のアクセス項目をリスト形式で出力できるようにしてみました。

- デザイン面でこだわったポイント

- 主題に合わせて食欲を落とさない色をテーマに選んでみました。
- Bootstrapを活用した反応型ウェブデザインで、様々な環境で使用することができます。



- APIデータの中にサムネイルがないデータがあり、他のイメージに自動的に交換する作業を追加しました。
- 追加説明がなくても機能が分かるように直観的なUI構成のために気を使いました

- アドバイスして欲しいポイント

PythonとDjangoが一番自信があるのでDjangoを活用しましたが複雑ではないプロジェクトにお勧めするフレームワークやツールがあれば知りたいです。

様々な理由で計画していたものとは異なり、DBの使用をあきらめました。が、本プロジェクトのようにユーザー認証が別にあるアプリ環境でもDBを活用する方法があれば、アドバイスをいただきたいです。そして、サーバーのSessionを活用したデータ保存以外にもデータ保存に関するアイデアがあれば、アドバイスをいただきたいです。

● 自己評価

ネットワークとサーバーについて詳しくなかったため、ネットワークセキュリティ構成で発生した問題で大変なことになりました。しかし、配布は課題で必須要求事項ではなかったこともあり、結果的に問題の原因は私個人的に考えた追加機能であるGoogle Maps APIのインターネット接続セキュリティと関連した問題でありました。何とか解決しましたが時間をたくさん消費したため、必須要求事項にもっと近いと思うBootstrapのSCSSを活用したデザイン構成に力を入れられなかったのが残念です。

プロジェクトの途中、データベースの誤った活用や、Geolocation API処理をFront-EndではなくBack-Endで進めて、ユーザーでないサーバの位置を読み込むなどの大きなミスがいくつかあったため、自分の足りなさを発見するようになった一方、様々なテストをしてみて、結局エラーを見つけて修正し、構想した内容をすべて実際に具現することができ、一段階成長できる機会になったと思います。

● 確認されたエラー

1. アドレスバーで/resultや/detailを入力すると、Bad Request Errorが発生するようになっていますが、ユーザーがまたPOST検索をする以外は、特別にSessionの終了時点を設定していないため、ユーザがブラウザを閉じないか、または新しい検索を試みない限り、他の画面に移動してもアドレスバーに/resultを入力すると、以前に検索した結果が表示されます。Sessionの寿命を設定することで解決できます。
2. ユーザがAdBlock類のアプリケーションを使用すると、ブラウザコンソール上で Failed to load エラーが発生し、google-maps-API keyが露出していますが、機能上のエラーはありません