

簡易仕様書

1 アプリ名：タベナビ（近くのレストラン検索 Web アプリ）

目的：

本アプリは、現在地を基に周辺の飲食店を検索し、ユーザーに最適な店舗を提案することを目的とすることです。多言語対応のユーザーインターフェースを提供することで、日本語がわからないユーザでも直感的に操作でき、旅行者や外国人利用者にとっても使いやすいサービスを目指します、アプリのインストール不要で即座に周辺の飲食店情報を得られる利便性を提供します。

主なターゲットユーザー

- 年齢層：10代～30代
- 現在地周辺で飲食店を探している日本人、訪日外国人
- 食事のジャンルや価格帯を比較・検討したいユーザー

2 対象 OS およびブラウザ

- デスクトップ：Windows 10 以上、macOS 10.15 以上
- スマートフォン：iOS 13 以上、Android 8.0 以上

3 開発環境

- エディタ：Visual Studio Code
- ローカルサーバー：MAMP
- ブラウザ：Google Chrome

4 開発言語

HTML5/CSS3、SCSS/ JavaScript

5 機能概要(機能一覧)

機能	機能内容
言語対応	日本語／英語の切り替え機能
現在地取得・地図表示	Geolocation API+Google Maps を利用して、現在地取得と地図上表示
周辺おすすめ	現在地周辺のランダムおすすめ店舗表示、リロード機能付き
レストラン検索	現在地周辺のレストランを Hot Pepper API を利用して検索
条件検索	予算、料理ジャンル、禁煙対応、絞り込み検索
キーワード補助検索	キーワードボタンを使った検索補助機能
検索結果表示	検索結果リスト表示&地図上マッピング
並び替え機能	価格順・おすすめ順で並び替え
お気に入り機能	localStorage を利用したお気に入り追加／削除機能
詳細カード管理	気になった店舗を詳細カードページで一括管理
店舗基本情報表示	店舗名、住所、ジャンル、営業時間等の基本情報表示
店舗詳細ページ遷移	詳細ページへのリンク（ホットペッパー予約リンク付き）
距離比較表示	店舗と現在地との距離を表示
ルート案内	Google Maps を通じた店舗までのルート案内
地図内操作	地図上から直接「お気に入り追加」「詳細ページ表示」可能
レスポンス対応	モバイル最適化（ハンバーガーメニュー搭載）

拡張構想（未実装機能）

機能カテゴリ	機能内容
ユーザーレビュー機能	ユーザーがレビュー投稿・評価スコア登録できる機能
お気に入りランキング表示	ユーザーのお気に入り数を基にした人気ランキング機能

6 開発期間

約 10 日間 + 整理・仕様書制作 2 日

7 フレームワーク (ver. 含む)

フレームワークは使用していません (jQuery 3.7.1 使用)

8 テーブル定義 (ER 図) などの設計ドキュメント

【localStorage テーブル】：

項目	型	説明
Id	string	店舗 ID
Name	string	店舗名
budget	string	平均予算
genre	string	ジャンル
Lat	number	緯度

項目	型	説明
Lng	number	経度
address	string	住所
Open	string	営業時間
access	string	最寄駅アクセス
photo	string	画像 URL
Link	string	ホットペッパーリンク

説明:

本アプリはローカル環境（フロントエンド）のみで開発を行っており、データ保存には localStorage を使用しています。

現時点ではデータベースやサーバーサイド技術（PHP 等）は使用していません。

しかし、将来的にサーバーサイド機能（データベース保存、API 通信等）を追加することを想定し、仮想的なテーブル設計および ER 図を作成しました。

以下にその想定設計を記載しました。

テーブル名	説明
shops	検索された飲食店情報を格納するテーブル
favorites	お気に入り登録された店舗を格納するテーブル
detail_cards	詳細カードに追加された店舗を格納するテーブル

shops テーブル

カラム名	データ型	制約	説明
id	VARCHAR	PRIMARY KEY	店舗 ID
name	VARCHAR	NOT NULL	店舗名
budget	VARCHAR		予算範囲
genre	VARCHAR		ジャンル
address	TEXT		住所
access	VARCHAR		最寄駅などのアクセス情報
open_hours	VARCHAR		営業時間
photo_url	TEXT		写真 URL
lat	FLOAT		緯度
lng	FLOAT		経度
non_smoking	VARCHAR		禁煙対応有無
card	VARCHAR		クレジットカード対応有無
wifi	VARCHAR		Wi-Fi 対応有無
parking	VARCHAR		駐車場の有無
private_room	VARCHAR		個室の有無
link_url	TEXT		ホットペッパーへのリンク URL

favorites テーブル

カラム名	データ型	制約	説明
id	INT	PRIMARY KEY, AUTO_INCREMENT	お気に入り ID
shop_id	VARCHAR	FOREIGN KEY (shops.id)	関連する店舗の ID

detail_cards テーブル

カラム名	データ型	制約	説明
id	INT	PRIMARY KEY, AUTO_INCREMENT	詳細カード ID (自動採番)
shop_id	VARCHAR	FOREIGN KEY (shops.id)	関連する店舗の ID

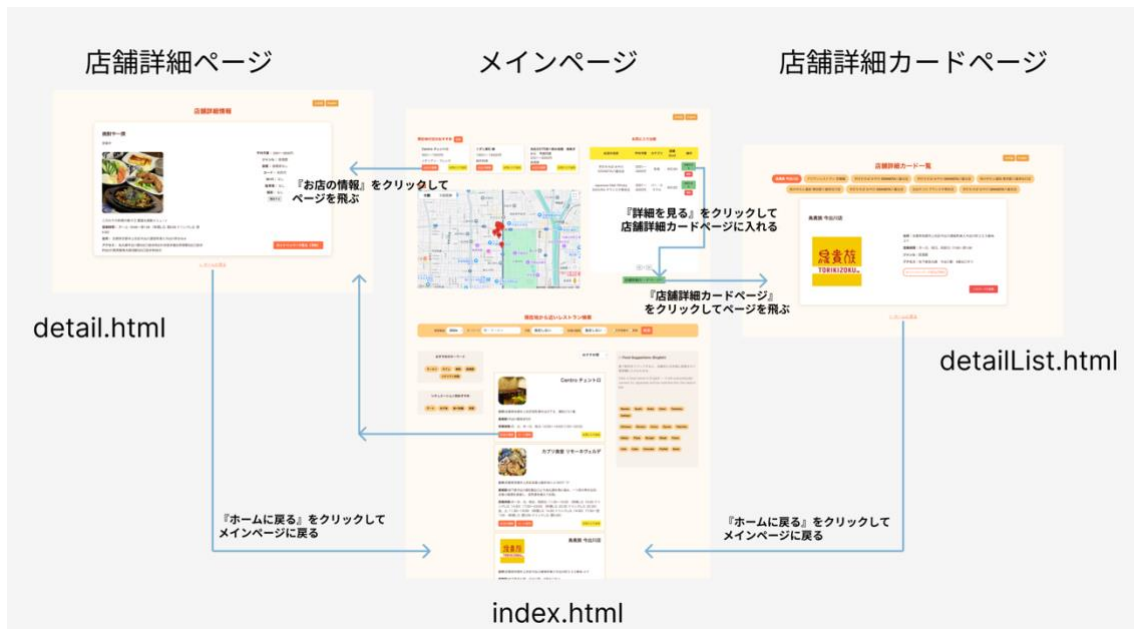
テーブル間のリレーション (ER 図)

shops (1) ----対---- (多) favorites
shops (1) ----対---- (多) detail_cards

【画面一覧】

画面	機能概要
ホーム画面 (index.html) 主な構成 : 検索エリア 地図表示エリア 結果画面リスト表示エリア キーワードエリア 店舗比較エリア	レストラン検索、結果表示、地図表示
店舗詳細画面__単体 (detail.html) 店舗詳細カード一覧画面 (detailList.html)	店舗ごとの詳細情報確認 お気に入り店舗をまとめて比較、管理

【ページ遷移図】：



(Figma で作成)

9 開発環境構築手順

1 MAMP のセットアップ

MAMP をインストールし、ローカルサーバー環境を構築。htdocs フォルダ配下に本プロジェクトフォルダを配置します。

2 API キーの取得と設定

Google Maps API キー、Hot Pepper グルメ API キーを取得し、index.js 内に設定します。

3 CORS 問題の対策

外部 API 通信時の CORS 制限を回避するため、cors-anywhere 等のプロキシサーバーを利用し、リクエスト URL に適用。

4 ブラウザで動作確認

ローカル環境にて `http://localhost/プロジェクトフォルダ名/index.html` をブラウザで開き、動作確認を行います。

10 コンセプト

多言語対応や地図連携により現在地から近いレストランを「素早く・直感的に」検索・比較できる Web アプリケーション。訪日外国人や在住外国人を含む幅広いユーザー層に向けて、食の選択肢をより手軽に提供しています。

11 こだわったポイント

- 現在地情報を活用し、即座に近くのレストランを提案。
- 日本語・英語の多言語対応により、外国人ユーザーにも配慮。
- お気に入り保存・比較機能を搭載し、ユーザー体験を向上。
- レスポンシブ設計：モバイルファーストで UI 最適化
- アクセシビリティ・グローバル対応：翻訳済みキーワード検索機能を提供
- 利用効率向上：条件検索・キーワードクリック補助機能を搭載
- エラー対策：位置情報取得失敗時のデフォルト座標設定、API 失敗時のエラーメッセージ表示
- 外部 API 通信における CORS 制限対策

以上のポイントにこだわることで、「快適」「速い」「誰にでも使える」アプリケーションとなるよう設計を進めました。

12 デザイン面でこだわったポイント

直感的なインターフェース設計によるユーザー体験向上

検索から結果表示までの操作フローを極力シンプルに設計しました。さらに、人気の高いキーワードをボタン化し、ワンクリックで検索できる補助機能を実装することで、文字入力の手間を削減し、スムーズな操作感を提供しました。さらに、日本語・英語を導入することで、訪日外国人や在住外国人でも利用しやすいユーザビリティを追求しました。

レスポンシブデザインによるマルチデバイス対応

スマートフォンからPCまで幅広い端末で快適に利用できるよう、画面サイズに応じてレイアウトや機能表示を最適化しました。スマートフォンでは、ハンバーガーメニューを導入し、限られた画面領域でも操作性を損なわない工夫を行いました。

食欲を喚起する配色

食欲を促進されると言われる赤と黄色を基調に配色設計を行いました。明るく温かみのあるトーンを採用することで、ユーザーに親しみやすさを感じさせ、飲食店検索という目的にマッチした雰囲気演出しています。また、主に赤色を「行動を促すボタンカラー」に用いることで、操作誘導性も高めています。

将来的な拡張性を考慮したコンポーネント設計

デザインコンポーネントを機能ごとに整理し、今後新たなジャンル追加やフィルター機能拡張などにも対応しやすい構成を意識しました。例えば、検索条件やおすすめキーワードなどは増減できる設計となっており、サービス応じたスケールアップを想定しています。

13 技術面でアドバイスして欲しいポイント

1 コードの可読性・保守性向上のための、クラス、ID 命名規則、コメント整理、関数分割、ファイル構成の標準化など、学校のチーム制作や将来の仕事場で実務レベルで特に意識すべきことは何かありますか？

2 個人開発では機能単位で一つずつに追加してきたが、実務に近い開発手順の意識の持ち方、進め方を教えて欲しいです。

3 機能追加に伴い、ページの表示速度や処理負荷が徐々に増加していくことになりました。このような場合、できる工夫や、設計段階で意識しておくべきポイントはありますか？

4 店舗情報が Hot Pepper API 依存のため不足感があり、ユーザーが次に何をすればいいか迷う場面もある。次回、まだ似たようなアプリを制作する際に、追加情報やガイド導線設計について知れば良い API もしくは知識はありますか？

5 今回のアプリはサーバーサイド技術を使用せずに開発しましたが、この場合でも意識すべき基本的なセキュリティ対策や、クライアント側で最低限守るべきポイントが知りたいです。

14 自己評価

本アプリ制作を通じ、以下の点で成長できたと感じています。

1 外部 API (Hot Pepper Gourmet API) を活用し、店舗情報取得から検索・比較・お気に入り登録まで、一連のフロントエンド開発フローを実践的に学ぶことができました。

2 言語切り替え機能) や、PC／モバイルに対応したレスポンシブデザインを実装し、将来的な機能拡張や多様なユーザー層への対応を考慮した取り組みをしました。

3 UX/UI への配慮では利用者の視線誘導を意識した配置やボタン設計、スマートフォンでの利用を意識したレイアウト最適化など、実際の利用シーンを想定したデザインに取り組みました。

【課題/まだ不足な点】

1 店舗決定後の行動導線の不足

検索結果から店舗を選んだ後、次の行動、例えば、店舗までの経路確認・即予約・周辺施設との比較検討などにスムーズに繋げる仕組みが不足しており、ユーザー離脱のリスクがあると感じました。

2 コードの構造改善余地

機能単位で段階的にコードを追加したため、全体としてリファクタリングの余地が大きいと認識しています。特に、DOM 操作の一元管理や共通処理のモジュール化などの知識や機能ごとの知識が不足のため、メンテナンス性がまだ改善が必要だと思います。

3 検索結果と地図エリアの連動性が低い

現在、検索結果リストと地図マーカの位置関係をユーザーが直感的に把握するためには、スクロール操作が必要であり、UX に若干の不便さが残っています。

【感想】：

本プロジェクトを通じて、「一つのアプリを最初から最後まで自走して作りきる」という経験を得られたことは、今後エンジニアとしてさらに成長するための大きな財産になったと思います。

開発を進める中で、実装力だけでなく、設計段階からユーザー体験を意識することの重要性や、拡張性・メンテナンス性を見据えたコード設計の必要性も強く実感しました。

今後は、今回の制作を通じて発見した課題をもとに、デザイン力、コーディング力の両面から学習と実践を重ね、より品質の高い開発ができるエンジニアを目指していきたいと考えています。