簡易仕様書

・アプリ名

NearEats

・対象 OS およびブラウザ（バージョン情報含む）

動作対象 OS

・Windows10 バージョン：22H2

・Windows11 バージョン：23H2

動作対象ブラウザ

・Google Chrome バージョン：131.0.6778.265

・Microsoft Edge バージョン：132.0.2957.127

・開発環境/言語

開発環境

Ruby on Rails バージョン：7.1.5.1

開発言語

Ruby バージョン：3.2.3

・開発期間

2025/01/16 ～ 2025/01/28（13 日間）

・機能概要（機能一覧）

機能	概要
レストラン検索	Geolocation API を使って現在地情報を取得し、ホットペッパーグルメサーチ API を使用して、現在地周辺の飲食店を検索する。検索条件は 検索範囲(現在地からの半径距離)，オプション，店舗名 の項目がある。
レストラン情報取得	ホットペッパーグルメサーチ API を使用して、飲食店の詳細情報を取得する。最大 50 件（お気に入りでは最大 20 件）の飲食店情報を取得できる。
Google Map 遷移	飲食店の所在地を Google Map(Web)で開く。
サインアップ・ログイン	アプリケーションに対してサインアップ・ログインができる。ログインすることで、一部機能が使えるようになる。
コメント投稿・閲覧 （投稿はログイン必要）	飲食店それぞれに対してコメントを投稿でき、各飲食店に投稿されたコメントは、飲食店の詳細情報画面で閲覧でき

	る。コメント投稿はログインが必要だが、ログインしていなくてもコメントを閲覧することはできる。
お気に入り登録 (ログイン必要)	ログインしている状態で詳細情報画面右上の星マークをクリックすることで、飲食店を最大 20 件までお気に入り登録することができる。お気に入り登録された飲食店は専用の一覧画面から詳細情報にアクセスできる。

・画面一覧

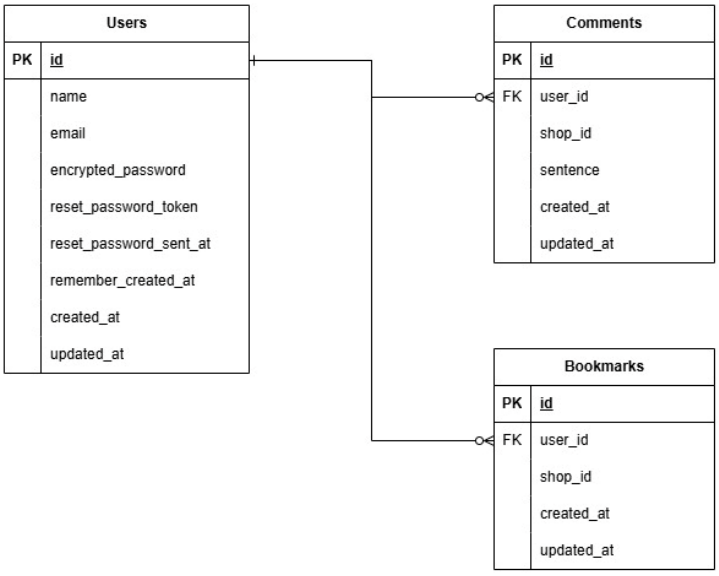
画面	概要
検索条件入力画面	条件を指定してレストランを検索する。
検索結果画面	検索結果の飲食店を一覧表示する。ページングに対応し、10 件区切りで表示される。
店舗詳細画面	対象飲食店の詳細情報を表示する。お気に入り登録・解除とコメント閲覧・投稿も行える。
サインアップ画面	名前, メールアドレス, パスワードを入力してユーザーを登録する。
ログイン画面	登録したメールアドレス, パスワードを入力してログインする。
投稿コメント一覧画面 (ログイン必要)	現在ログインしているユーザーが投稿したコメントを一覧表示する。コメント編集と削除が可能。
ユーザー情報変更画面 (ログイン必要)	名前、メールアドレス、パスワードの変更を行う。アカウント消去もここから行う。
お気に入り店舗一覧画面 (ログイン必要)	お気に入り登録した飲食店を一覧表示する。ページングに対応。
お気に入り店舗詳細画面 (ログイン必要)	お気に入り登録した対象飲食店の詳細情報を表示する。
サイドメニュー画面 (ログイン必要)	ログイン時、ヘッダーにあるアイコンから展開する。ユーザー関連の各ページへの遷移リンクがあり、ログアウトもここから行う。

・フレームワーク（ver.含む）

Ruby on Rails バージョン：7.1.5.1

・テーブル定義(ER 図)などの設計ドキュメント

テーブル定義(ER 図)



・開発環境構築手順（他の開発者が提出物のウェブアプリを動かし確認できるようにするための具体的な手順）

開発環境/言語に記載しているバージョン以上の Ruby と Ruby on Rails をインストールしている状態で Github の NearEats リポジトリを任意のローカルディレクトリに clone し、コマンドプロンプト等のターミナルで clone した NearEats ディレクトリに移動した上で、以下の順序に従って環境構築を行う。

1. “bundle install”

ターミナルで「bundle install」コマンドを実行する。Ruby バージョンが上記と異なる場合はエラーが発生するが、Gemfile に記述されている「ruby "3.2.3"」をローカルの Ruby バージョンに書き直して（例: ruby "3.3.6"）再度「bundle install」コマンドを実行することでエラーが解決する。

2. "rails db:migrate"

ターミナルで「rails db:migrate」コマンドを実行する。

3. NearEats ディレクトリの直下に.env ファイルを作成して API キーを記述する

プロジェクトのルートディレクトリである NearEats ディレクトリの直下に.env ファイルを作成する。そして、.env の一行目に「HOTPEPPER_API_KEY=ホットペッパーAPI キー」を記述する。（例: HOTPEPPER_API_KEY="aaaaaa111111"）

4. "rails s"
ターミナルで「rails s」コマンドを実行する。しばらくするとサーバーが立ち上がるので、ブラウザで「http://127.0.0.1:3000」(localhost）にアクセスする。すると Web アプリが起動する。ブラウザで初回アクセスした際に「Errno::EACCES」エラーが発生することがあるが、発生した場合は一度サーバーを切って、再度「rails s」でサーバーを立て直すことで解決することを確認している。

5. 現在地情報取得を許可
Web アプリが起動すると、ブラウザが現在地情報の使用の許可を求めるので、これを許可する。以上で、Web アプリが正しく動作するようになる。

（スマートフォン画面を確認する場合は、ブラウザの開発者ツールを開き、画面サイズをスマートフォン機種に変更することで確認できる）

・コンセプト

飲食店探しで困ったら検索！ 簡単に周りのいろんなお店情報をキャッチ。

・こだわったポイント

いろんな検索オプションを揃え、要望に沿った飲食店を見つけやすくしました。また、飲食店の場所を Google Map で開く機能やコメント機能を実装することでより多くの情報を知ることができるようにしたり、お気に入り機能を実装することで飲食店情報を見直すための再検索の手間を省くなど、利便性を向上させて快適な飲食店探しができるようにしました。ルーティング設定やページ遷移の流れを注意深く考え、画面移動で行き詰まることがないようにしました。

・デザイン面でこだわったポイント

シンプルで分かりやすい構成を意識し、PC 画面とスマートフォン画面の双方で見やすいレスポンシブなデザインにしました。細かい部分でこだわったポイントとしては、ボタンなど、似た意味をもつ要素を同じデザインで揃えることによって、直感的に分かりやすいデザインにしています。複数のページに跨る機能の画面では、ページ上部に遷移履歴を表示し、前段階画面に戻りやすくしています。ログイン時のユーザーに関する機能画面へのリンクを開閉可能なサイドメニューに置くことで、通常時はスッキリしたデザインを保ちながらも、サイドメニューを展開することでユーザー関連機能にアクセスしやすくしています。

・技術面でアドバイスして欲しいポイント

検索条件入力などのフォームをより使いやすくするにはどのような構成・配置にするべきかお伺いしたいです。また、HTML と CSS の記述が自身の想定よりも複雑になってしまいました。より管理しやすい HTML, CSS の記述方法についてお伺いしたいです。

・自己評価

機能面について、アプリの根幹となる機能は実装することができ、追加で必要だと考えた機能もいくつか実装することができました。デザイン面については、位置情報を使うことからスマートフォンでの使用も多いと思い、レスポンシブなデザインにする必要があると考えましたが、PC 画面でもスマートフォン画面でも見やすいデザインを作成することができたと考えています。この Web アプリケーションを使うことによって、ユーザーの要望に沿った周囲のレストランを見つけることができると考えています。

今後実装すべき機能として

・アプリからのネット予約機能

・電話アプリ連携機能

・アプリ内にマップを表示する機能

これらを追加で実装していくことによって、より便利になると考えています。