

お天気アプリ詳細設計書

1. 概要

本アプリは、都市名を入力することで、その都市の天気情報（気温、天候、湿度）と「傘が必要かどうか」のアドバイスを表示するシンプルな天気情報アプリです。

2. 機能要件

主な機能

- 都市名入力機能**
ユーザーが天気情報を取得したい都市名を入力する。
- 天気情報取得機能**
OpenWeatherMap APIを利用し、以下の情報を表示する。
 - 気温
 - 天候（例：晴れ、曇り、雨）
- 傘の必要性表示機能**
天候情報に基づき、「傘が必要かどうか」のアドバイスを表示する。

実装されていない機能（未実装事項）

- 天気アイコン表示機能**
天候に応じたアイコンを表示する機能は未実装。
 - 理由:** Font Awesomeの導入がうまくいかなかったため。
 - 現状:** テキストベースで天候の状態（例：「晴れ」や「雨」）を表示することで代替している。

将来的な追加機能（改善計画）

- 天気アイコン表示**
天候に応じたアイコンを表示し、視覚的にわかりやすくする。
 - 対策:** Font Awesomeの再導入または他のアイコンライブラリを検討する。
- レスポンスデザイン対応**
デバイス（PC、タブレット、スマートフォン）ごとに最適な画面表示を実現する。

3. システム構成

アーキテクチャ

- クライアントサイドのみのシンプルな構成
- フロントエンド: HTML, CSS, JavaScript
- API: OpenWeatherMap API（天気情報取得）

技術スタック

- HTML/CSS:** 画面の構築とスタイル適用

- **JavaScript:** 天気情報の取得と表示
- **OpenWeatherMap API:** 天気データの取得

4. UI/UX設計

画面構成

1. 天気情報表示画面
- 都市名入力フォーム
 - 「天気を表示」ボタン
 - 結果表示エリア（気温、天候、傘の必要性）

未実装事項に関する説明

- **天気アイコン表示エリア:** 将来的な機能拡張時に実装を検討する。

5. 開発スケジュール（計3日 + 1日発表）

日程	作業内容
1日目	- 設計およびOpenWeatherMap APIの動作確認
2日目	- 基本機能（都市入力、天気情報表示）の実装
3日目	- 「傘の必要性」の表示追加とデバッグ
4日目	- 発表準備、最終確認

6. リソース

- **使用時間:** 1日3時間の作業時間
- **進捗管理:** GitHubによるバージョン管理

7. リスクと対策

リスク	対策
APIのレスポンスが遅延する	予備データを準備して対応
Font Awesomeが動作しない問題	別ライブラリの導入や手動表示に切り替え
開発時間が不足する	優先順位を決め、機能を限定する

8. 品質管理とテスト

テスト戦略

- **単体テスト:** 都市名入力と天気情報表示のテスト

- **統合テスト:** APIとJavaScript連携の確認

品質基準

- テキスト表示による天気情報の正確な取得・表示
 - エラーメッセージが適切に表示されること
-

9. 納品物

- **天気情報アプリ最終版**
 - 天気情報（気温、天候）および傘の必要性を表示
 - **天気アイコン表示は未実装**
-

10. 運用・保守計画

保守計画

- **天気アイコン表示の追加:** Font Awesomeや他のライブラリを導入し、視覚的な改善を図る
 - **レスポンシブデザインの対応:** モバイル・タブレット向け表示の最適化
-

11. レビューと反省

自己レビュー

- **良かった点:** 基本的な天気情報表示機能は実装できた
- **改善点:**
 - Font Awesomeの導入失敗を解決できなかった
 - 将来的には、天気アイコンやレスポンシブデザインを追加することで、ユーザビリティを高める