|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 科目名 | 年度 | レポート番号 | クラス | 学籍番号 | 名前 |
| API実習 | 2024 | 4 | A | 20123020 | 五味帆翔 |

レポート(5)で開発するAPIを使ったシステムについて企画、要件定義を行ってください。ページ数や文字数よりも、読んでわかりやすく書けているかどうかが、点数アップの分かれ目です。本レポートにおける要件定義項目は、本来の要件定義項目から抜粋した簡易な内容になっています。

# 種別（API開発+アプリ開発なのか、API連携からのサービス開発なのか識別するため必須）

どちらかに〇をつけること。API連携は楽ですが、評価は低くなりやすいです。

独自API+ APIを使ったアプリ開発　 / API連携◯

# 業務要件

## 概要

OpenWeatherMap APIを利用したweb天気予報アプリ開発

## 背景

私は気圧に弱く、iPhoneの天気予報では低気圧や湿度などの詳細なデータがないため、困っており、自分が欲しいデータを拡張できる天気予報アプリが欲しいと考えた。

## 目的

晴れや雨などの天気の情報に加え、低気圧予防に気圧情報、服の乾きなども知るために湿度の情報も付け加えたい。

## 想定利用対象者

低気圧に弱い人

湿度が気になる人

## 業務フロー

|  |  |  |
| --- | --- | --- |
| ユーザー | フロントエンド | OpenWeatherMap API |
| 検索バーに都市名を入力し、「検索」ボタンを押す。  別の都市を選択  ・  ・  ・ | OpenWeatherMap API にリクエストを送信。  現在の天気と未来の天気予報を画面に表示。 | 指定された都市の天気予報データ（気温、湿度、気圧など）を JSON 形式で返す。 |

# 機能要件

## システム構成図

ユーザー

画面に天気情報を表示

ブラウザアクセス

フロントエンド

天気データ取得

APIリクエスト

OpenWeatherMap API

## 使用外部サービス一覧

・OpenWeatherMap API

・Next.js

・Firebase Cloud Messaging

## 想定画面

### 画面一覧

・ホーム画面

### 画面レイアウト

検索

◯◯市の現在の天気

天気　晴れ

湿度　40%

気圧　1000hpa

20◯◯年　◯月◯日　0時

天気　晴れ

湿度　40%

気圧　1000hpa

20◯◯年　◯月◯日　3時

天気　晴れ

湿度　40%

気圧　1000hpa

20◯◯年　◯月◯日　6時

天気　晴れ

湿度　40%

気圧　1000hpa

## データ定義（API開発している場合は必須）

### テーブル一覧

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| dt\_txt | main.temp | main. pressure | main. humidity | weather[0].description |
| weather[0].icon |  |  |  |  |

# 非機能要件

## 拡張性

・多言語対応

・風速、降水量、UVインデックスなど天気の詳細化

## 稼働環境

### システム側

・OpenWeatherMAPAPI 無償版

### 利用者側(クライアント環境)

・121.0.6167.85（Official Build） （arm64）

・Google Chromeが利用できる端末

iPad,iphone,ipodtouch:iOS15以降・Android8.0(Oreo)以降・Windows10以降・macOS Catalina10.15以降・

６４ビットUbuntu18.04以降、Debian10以降、openSUSE15.2以降、Fedora Linux32以降

## 保守

1. 保守担当者

担当者: システム管理者

責任範囲: サーバー運用、APIの状態監視

2. 保守頻度

定期的な点検: 月に1回

システム更新: 新しいバージョンやパッチがリリースされた場合、随時対応

3. 保守内容

API監視: 天気予報APIの正常性を監視し、障害が発生した場合には即時対応

バグ修正: ユーザーからのフィードバックやエラーログに基づいて、発生したバグの修正を行う

パフォーマンスの最適化: アプリケーションのパフォーマンス（表示速度、レスポンスタイムなど）を定期的に確認し、最適化する

セキュリティパッチの適用: 使用しているライブラリやフレームワークにセキュリティの脆弱性が発見された場合には、速やかにパッチを適用する

サーバーメンテナンス: サーバーが最適な状態で動作するよう、定期的にメンテナンスを行う（バックアップ、リソースのモニタリング、ログの分析）