要件定義書

1. システム概要

東急東横線・みなとみらい線の時刻表データをスクレイピングし、JTB時刻表形式のダイヤグラムをPDF形式で自動生成するシステム。

2. 背景と目的

- 現状、全駅の時刻情報を一覧できるダイヤグラムが存在しない。
- 乗り換え検索サービスでは、列車の比較がしにくい。
- 列車種別や直通運転の複雑さがあるため、一目で把握できるダイヤグラムが有用。
- 年に一回のダイヤ改正に対応できる仕組みを構築する。

3. ユースケース

- ユーザー（開発者本人）が手動でスクリプトを実行し、PDFのダイヤグラムを生成する。
- 必要に応じてプログラムを改修し、将来的に他路線にも拡張可能にする。

4. 要件一覧

4.1 前提要件

項目	内容
対象路線	東急東横線・みなとみらい線
対象駅	渋谷～元町・中華街の全駅
対象曜日	平日、土曜、休日（祝日は除く）
対象時間帯	始発～終電
列車の種類	全種類（特急、通勤特急、急行、各駅停車など）
ダイヤ改正	年に1回の改正に対応

4.2 機能要件

機能名	説明

データ取得機能	NAVITIMEの各駅時刻表ページからHTMLを解析し、時刻データを取得
データ整形機能	取得したデータをダイヤグラム生成に適した形式に変換
ダイヤグラム生成機能	ReportLabでPDF形式のダイヤグラムを作成
エラー処理機能	取得失敗時にエラーメッセージを出しつつ、取得可能なデータで処理を進行
ログ出力機能	デバッグレベルの詳細なログを実行ディレクトリに出力

4.3 非機能要件

項目	内容
出力形式	PDF
データソース	NAVITIME東急東横線時刻表
プログラミング言語	Python（requests, BeautifulSoup, ReportLabを使用）
実行環境	CLIベース
データ保持	ローカル保存不要。毎回スクレイピングを実行
URL変更時の対応	URL体系が変わった場合はコードを修正
依存管理	requirements.txt を用意

5. 想定するシステムの流れ

1. ユーザーがスクリプトを実行。
2. NAVITIMEの各駅時刻表ページをスクレイピング。
3. 取得データを整形し、ダイヤグラムを作成。
4. ReportLabでPDFを生成。
5. PDFファイルを出力し、ユーザーが確認。

基本設計書

1. システム構成

1.1 モジュール構成

モジュール名	役割
`data_acquisition.py`	時刻表データをNAVITIMEから取得
`diagram_generator.py`	ダイヤグラムの描画とPDF出力
`main.py`	メイン処理（データ取得・整形・描画の統括）
`logger.py`	ログ出力を管理

1.2 データ取得の詳細

- BeautifulSoupでHTMLを解析し、各列車の発車時刻・列車種別・始発駅・終着駅を抽出。
- 取得データを辞書型リストとして保持し、ダイヤグラム生成に利用。
- URL体系が変わった場合は手動で修正。

1.3 ダイヤグラム生成の詳細

- ReportLabを使用してPDFキャンバスを作成。
- 縦軸: 駅（上から渋谷、下へ元町・中華街の順）
- 横軸: 時刻（10分刻み）
- 列車種別ごとに色分けした線を描画。
- 必要に応じて、列車の始発駅・終着駅・運休情報を表示。

1.4 エラー処理

- 取得失敗時はエラーメッセージを出力。
- 取得できた範囲で処理を進行。

1.5 ログ出力

- `logger.py` でログ管理。
- デバッグレベルの詳細なログを実行ディレクトリ内のファイルに出力。
- 主要な処理（データ取得、変換、PDF生成）のタイミングでログを記録。

1.6 シーケンス図

ユーザー → main.py : スクリプト実行
main.py → data_acquisition.py : NAVITIMEから時刻データ取得
data_acquisition.py → diagram_generator.py : データ整形後に渡す

diagram_generator.py → ReportLab : PDF生成
ReportLab → ユーザー : PDFファイル出力

2. 実行手順

1. Python環境をセットアップ。
2. `requirements.txt` から必要ライブラリをインストール。
3. `main.py` を実行。
4. 生成されたPDFを確認。

このドキュメントを基に、今後詳細な設計を詰めていきます。追加で修正したい点があれば教えてください！