

ソフトウェア開発企画書

1. 企画書のタイトル

プロジェクト名:
USD/JPY 分析ダッシュボード
作成日: 2024.12.11
作成者: 中島麻莉

2. プロジェクト概要

目的:
毎日のUSD/JPY分析情報を見やすく
するためのダッシュボードシステムを開発する。
交換レートの分析や売買シグナルを
表示し、ユーザーに利益情報を提供。

3. ターゲットユーザー

学生、フリーランス、一般の方

4. 機能要件

主な機能:
リアルタイムのUSD/JPY値表示。
移動平均線、RSIの計算と表示。
売買シグナルの決定と表示。
利益計算、運用利益の表示。

5. システム構成

フロントエンド: HTML, CSS
, Dash (Python)
バックエンド: Python ver3.12.4
データベース: SQLite ver 3.26.0

6. 開発スケジュール

設計: 2日
実装: 1週間
テスト: 1日

7. リソース

使用時間:
週に10時間程度
進捗管理方法:
GitHubでバージョン管理

8. リスクと対策

予測されるリスク:
モチベーションの低下
対策:
切りの良い目標設定と進捗共有

9. 品質管理とテスト

テスト戦略:
単体テスト、組み合わせテスト、UIテスト
品質基準:
バグなし、直感的で操作しやすいUI

10. 納品物

納品物の概要:
USD/JPY 分析ダッシュボード
納品予定日:
2024.12.19

11. 運用・保守計画

運用サポート:
バグ修正、実行中のメンテナンス。
保守計画:
ユーザーのフィードバックに基づき機能追加を検討

12. レビューと反省

自己レビュー: 情報の整合性、システムの操作性およびユーザビリティを評価
今後の課題: ユーザー数増加に伴う負荷対策と新機能の開発計画

詳細設計書

■ 画面遷移図: fx_analysis.py(<http://127.0.0.1:8050/>)

1. 初期画面 (Dashboard)

表示項目:

- タイトル「USD/JPY Analysis Dashboard」
- 移動平均クロスオーバー戦略による総利益 (total_return)
- 純利益表示 (データベースから取得した総利益)
- 直近の取引履歴 (最新の3件の取引を表示)
- 売買ボタン (買ボタン、売ボタン)
- 価格チャート (USD/JPYの価格、移動平均線、売買シグナル)
- RSIチャート (RSIの値とアラートライン)

2. 売買ボタン押下

アクション: 買ボタンまたは売ボタンがクリックされる。

遷移: 取引履歴が更新される。

新しい取引情報 (日時、アクション、価格、利益) がデータベースに追加される。

直近の取引履歴が再表示され、取引情報が更新される。

3. 取引履歴表示

表示項目: 最新の取引履歴 (取引日時、アクション、価格、利益)

売買ボタンが表示されたまま。

4. 総利益表示

アクション: 売買ボタンがクリックされると、取引ごとに総利益が更新され、ダッシュボード上に表示される。

■ データベース構造:

テーブル構造 (次項で示す ER 図と連携)。

■ システム構成:

- サーバー: Flask (Python)
- データベース: SQLite
- クライアント: Webブラウザ

■ コントローラの仕様:

1. データベース操作関連

- create_db
- add_trade
- calculate_total_profit
- get_latest_trades

概要:

- データベースとテーブルを作成します。
- 新しい取引をデータベースに追加します。
- 総利益を計算します。
- 最新の取引履歴を取得します。

2. 為替データの取得と分析

- fetch_fx_data
- calculate_sma
- calculate_rsi
- generate_signals

- backtest_strategy

- Yahoo Financeから為替データを取得します。
- 移動平均線 (SMA) を計算します。
- 相対力指数 (RSI) を計算します。
- 移動平均クロスオーバー戦略で売買シグナルを生成します。
- 移動平均クロスオーバー戦略をバックテストし、戦略のリターンを計算します。

3. ダッシュボード表示関連

run_dashboard

Dashを使用してダッシュボードを起動し、ユーザーインターフェースを表示します。

update_profit

売買ボタンがクリックされたときに、取引を記録し、利益を更新します。

4. エラー処理

ER図

1. 取引 (Trade)

id: 主キー (取引を一意に識別)

date: 取引日時

action: 取引の種類 (「買い」または「売り」)

price: 取引価格

profit: 利益 (前回の取引との比較による計算)

2. 取引履歴 (Transaction)

transaction_id: 主キー (取引履歴を一意に識別)

amount: 取引金額

status: 取引の状態 (例: 完了、未処理)

取引 (Trade) と 取引履歴 (Transaction) は1対多の関係

| trades | |
|--------|--------|
| 123 | id |
| A-Z | date |
| A-Z | action |
| 123 | price |
| 123 | profit |

テスト表

テストケースID、機能名、入力、期待される結果、結果

TC001: 取引の追加->新しい取引 (売り、買い) をデータベースに追加する

->取引が正常にデータベースに追加される->OK

TC002: 利益計算->直前の取引と現在の取引を比較して利益を計算する->利益が正しく計算される(売り: 現在の価格 - 前回の買値、買い: 前回の売値 - 現在の価格)->OK

TC003: 総利益の計算->すべての取引の利益を合計して総利益を計算する->総利益が正しく合計される->OK

TC004: 取引履歴の表示->取引履歴の表示が正しく行われるか確認->最新の取引履歴がリスト表示される->OK

TC005: グラフの表示 (価格チャート) ->価格チャートに最新データが正しく表示される->為替レートのチャートが表示される->OK

TC006: RSIチャートの表示->RSIチャートにオーバーボートやオーバーソールドのラインが正しく表示される->RSIが70を超えるとオーバーボート、30を下回るとオーバーソールドが表示される->OK

TC007: データベース接続->データベースに正しく接続できるか確認->データベース接続が成功する->OK

TC008: エラーハンドリング->取引追加時のエラー処理が正しく機能するか確認->エラー発生時に適切なエラーメッセージが表示される->OK

TC008: 取引ボタンの動作確認->売買ボタンをクリックした際に取引が記録される->ボタンをクリックすると取引がデータベースに記録され、画面に反映される->OK

操作手順書

1. アプリケーションの起動

2. 取引の入力

「買い」または「売り」を選択します。

3. 取引履歴の確認

4. 利益計算の確認

総利益が表示されます。

5. グラフ表示（為替レート、RSIチャート）