

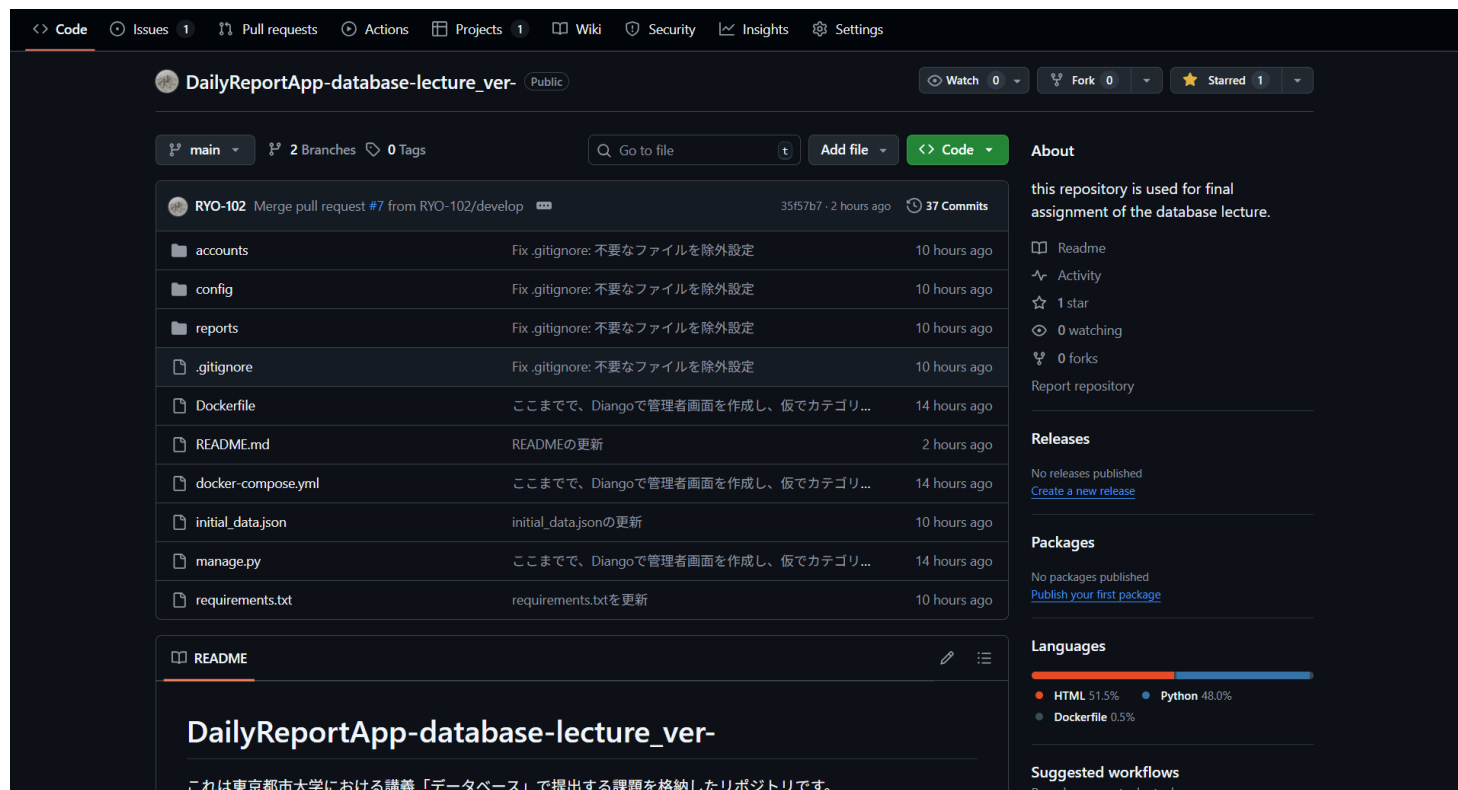
日報アプリプレゼンテーション

チーム名：夜行性

チームメンバー：渡邊諒（2442102）・田中誠真（2442054）

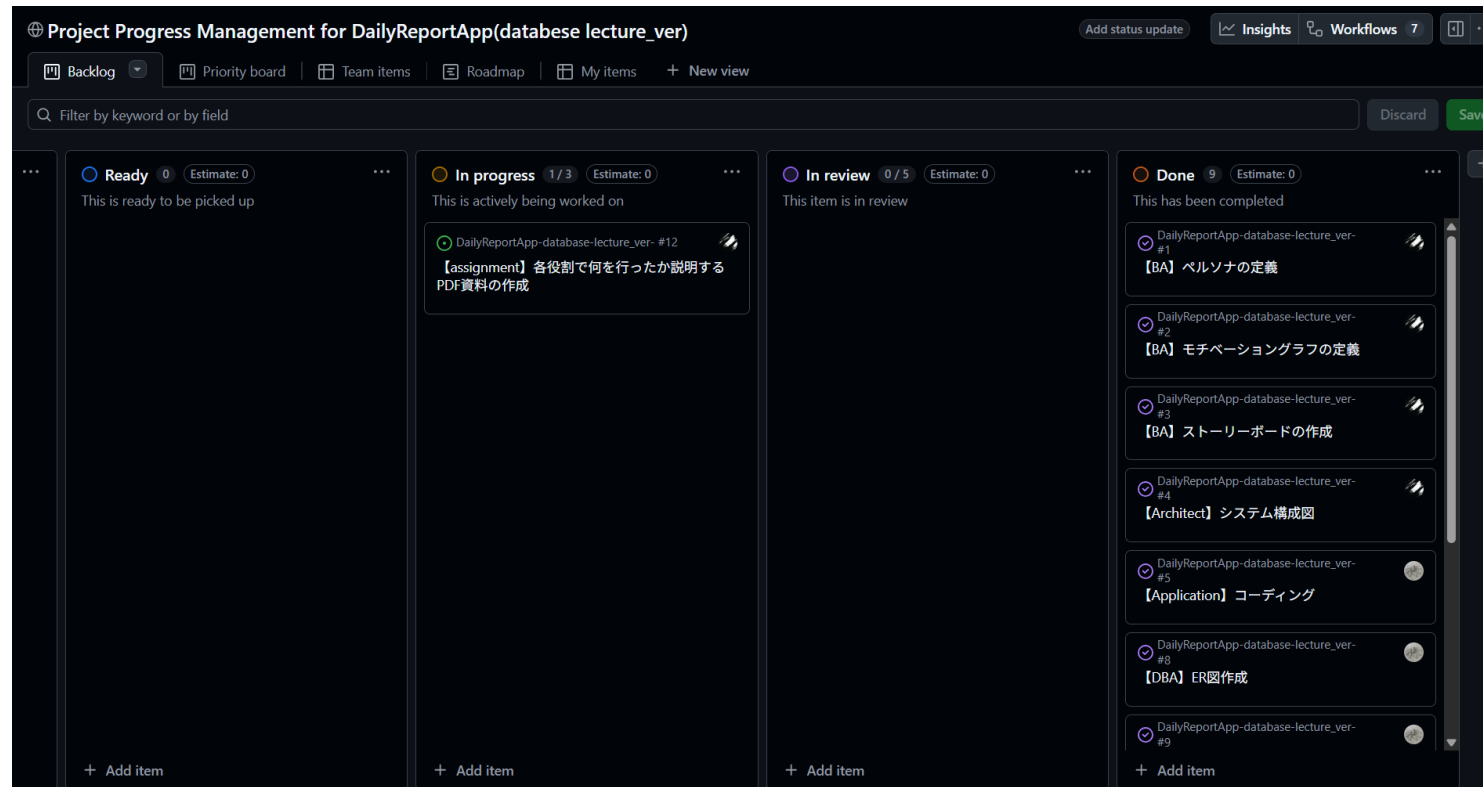
GitHubリポジトリによる構成管理

<リポジトリトップ画面>



Kanban Boardによるタスク管理

<Kanban Board画面>



ペルソナ

氏名: 田中 健一

属性: 38歳 / システム開発会社 PM兼テックリード / 年収 750万円

役割: 自身の開発タスク(30%) + メンバー10名のマネジメント(70%)。

性格・価値観:

- 「チームのメンバーに対しての関心は必要」という考え。
- 「メンバーのSOSはいち早く察知したい」という意識が強い。
- 日報は「読み捨て」ではなく、後から検索できる「チームの資産」であるべきだと考えている。

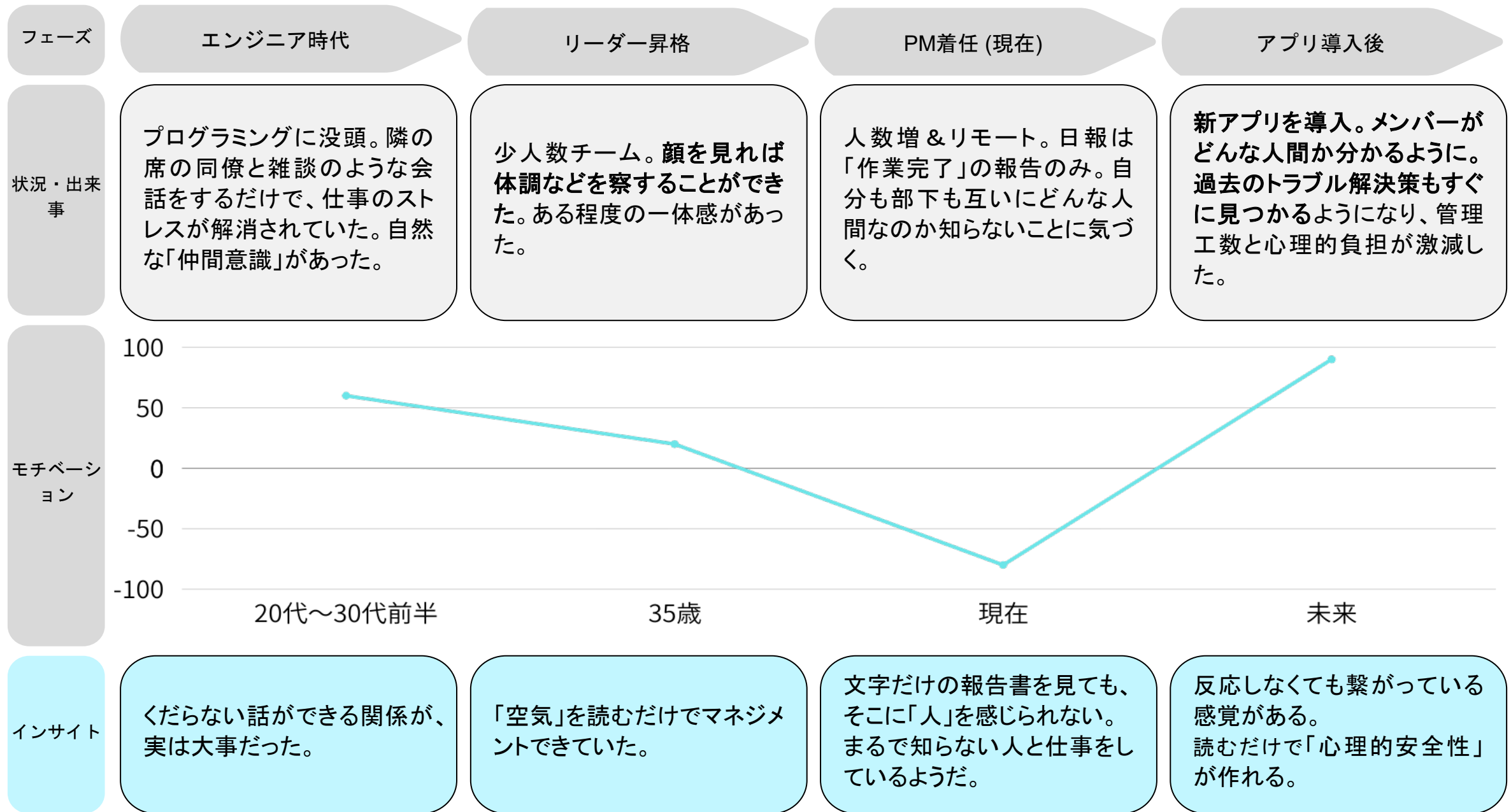
ペルソナ

現状の悩み:

- お互いに「仕事の顔」しか見せないため、雑談が生まれず、困ったときに相談しにくい空気がある。
- 過去に起きたエラーの解決策がチャットで流れてしまい、同じミスが繰り返されることがある。

ゴール:

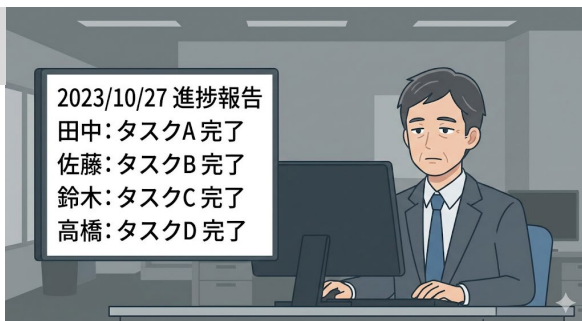
- 読むだけで相手の人物像や調子がなんとなく分かる状態を作りたい。
- 雑談や画像も含めた記録が、**将来の誰かの助けになる仕組み**を作りたい。



ストーリーボード

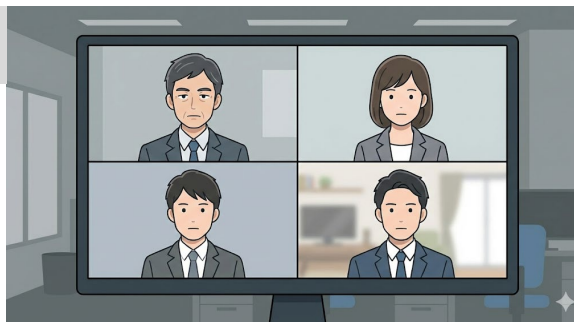
現在(アプリ導入前)

1



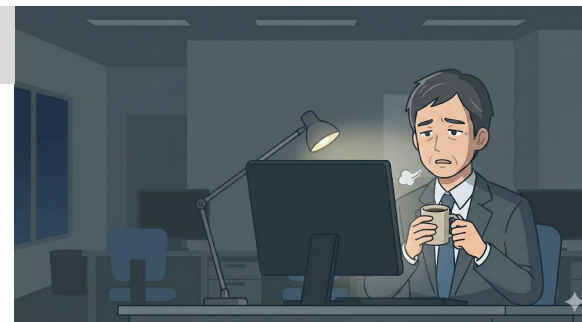
「届くのは『記号』のような報告だけ。」
進捗は把握できるが、その向こうにいるメンバーの顔が見えてこない。

2



「余白のないチーム。」
業務以外の会話はなく、雑談がない張り詰めた空気。

3



「見えない壁。」
トラブルはないはずなのに、チームとしての冷たさや孤独感を感じてしまう。

ストーリーボード

アプリ導入後

1



「自分の『人となり』をさらけ出す。」
業務報告のついでに、プライベートな一面や今の気分を書き添える。

2



「読むだけ。」
特別なことはせず、ただ読んで、「〇〇さんもそんなことするんだ」などと知るだけ。

3



「知っているから、話しやすい。」
日報で得た「相手の情報」が潤滑油になり、業務の相談もスムーズに始まる。

技術スタック

開発言語・フレームワーク:

Python 3.11 / Django 5.0: 迅速な開発が可能なMVTモデルを採用し、強力なORMと管理画面機能を活用しました。

データベース:

PostgreSQL 15: 複雑なクエリやトランザクション処理に強く、信頼性が高いリレーショナルデータベースです。

インフラ・フロントエンド:

Docker: 開発環境と本番環境の差異を極小化できます。

HTML5/CSS3: レスポンシブ対応により、PC・スマホ双方での利用を想定。

システム全体構成図

外部公開:
ホスト環境のngrok
を経由し、インター
ネット側からのリク
エストをDockerコン
テナ内のポート
8000へ転送。

- 【機能要件】
- 日報投稿・編集・削除
 - コメント・画像アップロード
 - SOS検知・活動ランキング
- 【非機能要件: 性能・保守性】
- N+1問題対策: select_related / prefetch_related
 - 開発効率: Django Adminによる管理機能
 - エラーハンドリング: 403/CSRF等の適切な制御

利用デバイス:
PCブラウザ / スマートフォン
(レスポンシブデザイン対応)

一般ユーザー / 管理者

HTTPS Request

インターネット

Forwarding

「Tunneling」
ngrok

- 【アクセス制御】
- HTTPSによる暗号化通信
 - 外部からのセキュアなアクセス
 - CSRF信頼済みオリジン設定

Tunneling to localhost

ホスト環境 (Local Machine)

Docker Compose 環境

Webサーバー / アプリケーション
(Django 5.0 / Python 3.11)

機能モジュール

認証・権限管理 (Auth)

日報管理 (CRUD)

SOS・ランキング集計 (Aggregation)

検索・フィルタリング (Qオブジェクト)

データアクセス層

ORM (Object-Relational Mapping)

TCP接続 (SQL発行)

5432

データベース
(PostgreSQL 15)

画像保存

- 【データ整合性 (ACID)】
- transaction.atomicによる一貫性の保証
 - リレーショナル構造による正規化データ管理

データ書き込み/読み出し

Docker Volumes
(データ永続化領域)

Postgres Data
(db_data)

Media Files
(uploads)

- 【バックアップ・可用性】
- コンテナ再起動・破棄への耐性
 - dumpdataによるデータ保全
 - ホスト側へのデータマウント

Web3層構造:
「Webサーバー/ア
プリ(Django)」 「DB
サーバー
(PostgreSQL)」を分
離・連携。

データ永続化:
Docker Volumesを使用
し、コンテナを再起動し
てもデータが消えない
構成を構築。

Dockerを用いたコンテナ型開発基盤の構築

インフラ構成のポイント:

OSの統一: 開発者全員が同じLinux環境で動作することを保証。

ライブラリ管理:

PythonやPostgreSQLのバージョンをコンテナ内で固定し、ライブラリの衝突(Dependency Hell)を防止。

メリット:

「自分の環境では動くが、他の環境では動かない」という問題を根本的に解決。

プロジェクト構造と環境分離

ディレクトリ構成:

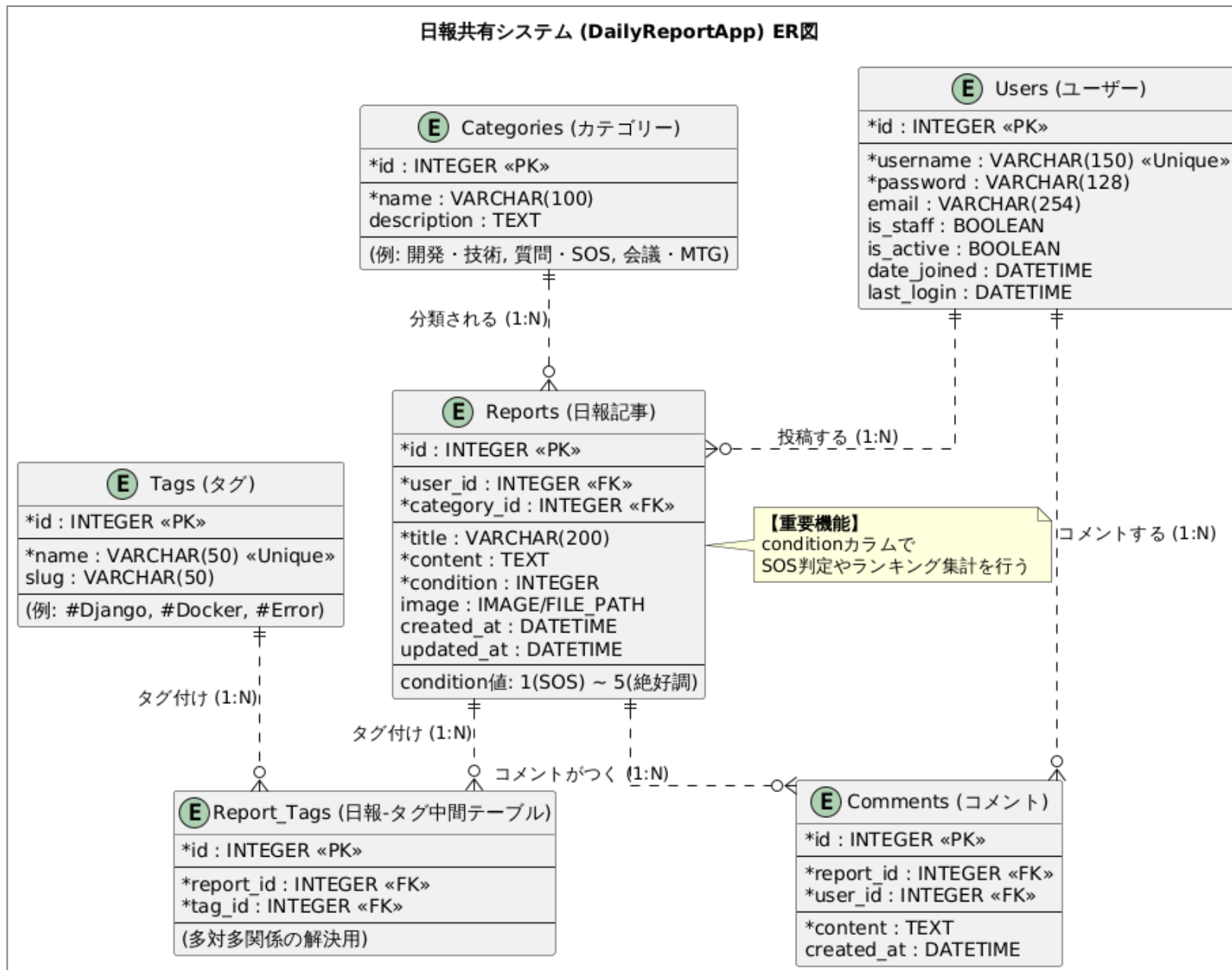
機能ごとにアプリケーション (reports, accounts) を分割し、見通しの良い構成を維持。

- └─ config/ # 全体の設定 (settings.py)
- └─ reports/ # 日報機能 (models, views)
- └─ accounts/ # ユーザー認証・管理機能

環境設定 (settings.py):

ALLOWED_HOSTS や CSRF_TRUSTED_ORIGINS を設定し、Docker環境および外部公開 (ngrok) に対応。

<ER図>



デモ / 実際の画面

みんなの日報一覧

ログイン

キーワード...

全てのカテゴリ

検索

新 集計

UIデザインの修正案について

良い

admin さん | 2025/12/23 17:42 | 開発業務

python タラーメン



本日のミーティングお疲れ様でした。一輪漫画のレイアウトですが、スマホで見た時に崩れてしまう問題に対応するため、CSSのメディアクエリ (@media) を追加しました。PCでは横並び、スマホでは縦並...[もっと読む](#)

【SOS】Dockerコンテナが起動しません...助けてください

SOS

suzuki さん | 2025/12/23 17:38 | 開発業務

#エラー解決

docker-compose up をしたところ、以下のエラーが出て起動できなくなっていました。

Error: port is already allocated

一覧に戻る

【SOS】Dockerコンテナが起動しません...助けてください

カテゴリ: 開発業務 | 林 PV数: 14 | 調子: SOS (待機SOS)

docker-compose up をしたところ、以下のエラーが出て起動できなくなりました。

Error: port is already allocated

一度PCを再起動してみましたが直りません。どなたか、競合しているポートの探し方や、強制終了のコマンドを知りませんか？ 開発が止まってしまって焦っています....

#エラー解決

コメント (1件)

tanaka さん | 2025/12/24 02:25

【エラー解決についてのコメント】

コメントを書く

コメントを投稿するには ログイン してください。

チーム活動分析レポート

一覧に戻る

貢献度ランキング

DB技術: JOIN + GROUP BY + COUNT による高速集計

順位	名前	投稿数
1	tanaka	1回
2	admin	1回
3	suzuki	1回

SOS発信状況

DB技術: FILTER句を用いた条件付きCOUNT集計

※ 困っているメンバーを早期発見するためのリストです。

順位	名前	SOS回数
1	suzuki	1回

Django administration

Site administration

ACCOUNTS

社員

+ Add

Change

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Change

REPORTS

Comments

+ Add

Change

カテゴリ

+ Add

Change

タグ

+ Add

Change

日報

+ Add

Change

Recent actions

My actions

tanaka -> 【SOS】Dockerコンテナが起動しません...助けてください

Comment

+ tanaka game

日報

+ suzuki test

日報

+ tanaka test

日報

+ suzuki (システム開発部)

社員

+ tanaka (システム開発部)

社員

+ ラーメン

タグ

+ ゲーム

タグ

+ エラー解決

タグ

+ Python

タグ

アプリケーション概要