

気分記録アプリケーション「lantana」

開発動機

ライフログビューアアプリケーションに気分による検索機能が存在せず、ほしい

自作ライフログビューアアプリケーションには現状、気分を数値で記録・検索する機能が存在しない。
気分が良かったときには何をしていたのだろう、気分が良くなかったときには何があったのだろう。
これらを知るためにはテキスト検索では難しい。
そこで今回は、現在の気分を5段階で評価するアプリケーションを制作して機能追加をする。
気分には理由がある場合がある。それを記録するため、関連するテキストを複数個添えられるといい。

ドキュメント作成練習

前回開発時の反省点を活かして、「見せるため」のドキュメント作成の練習をする。
今回はWordではなくExcelで設計書を作成し、画面設計図とアプリケーション全体像の共有に力を入れる。
また、実装することによって、実現可能なドキュメントとして磨いていく。
この開発のゴールは実装完了報告完了、かつ完了動画の完成とする。

要求・要件

気分10段階評価記録アプリケーション

気分評価画面

アプリケーションは気分評価画面を表示する
ユーザは現在の気分を10段階評価して送信する
アプリケーションは情報を保存する

記録一覧表示画面

アプリケーションは気分評価画面を表示する
ユーザは記録一覧表示画面表示アイコンを選択する
アプリケーションは記録一覧表示画面を表示する

記録したデータはライフログビューア（別アプリケーション）で表示可能である
製作動機なので必ず満たす

どこでも動く

アプリケーションはWindows,Linux,Android上で動作するウェブアプリケーションである

lantanaアプリケーションランチャー

能動的に気分を記録する場合は、いい気分・悪い気分に関りが出ると予測できる。
そこで、たまにlantanaを起動でユーザに気分入力促すランチャーも合わせて制作する。
（鬱陶しいアプリケーションとなることはなんとなくわかっているがそれでも作る・・・）
（手続きのなプログラムになると思われるので本ドキュメントには設計を載せない）

tentomushi

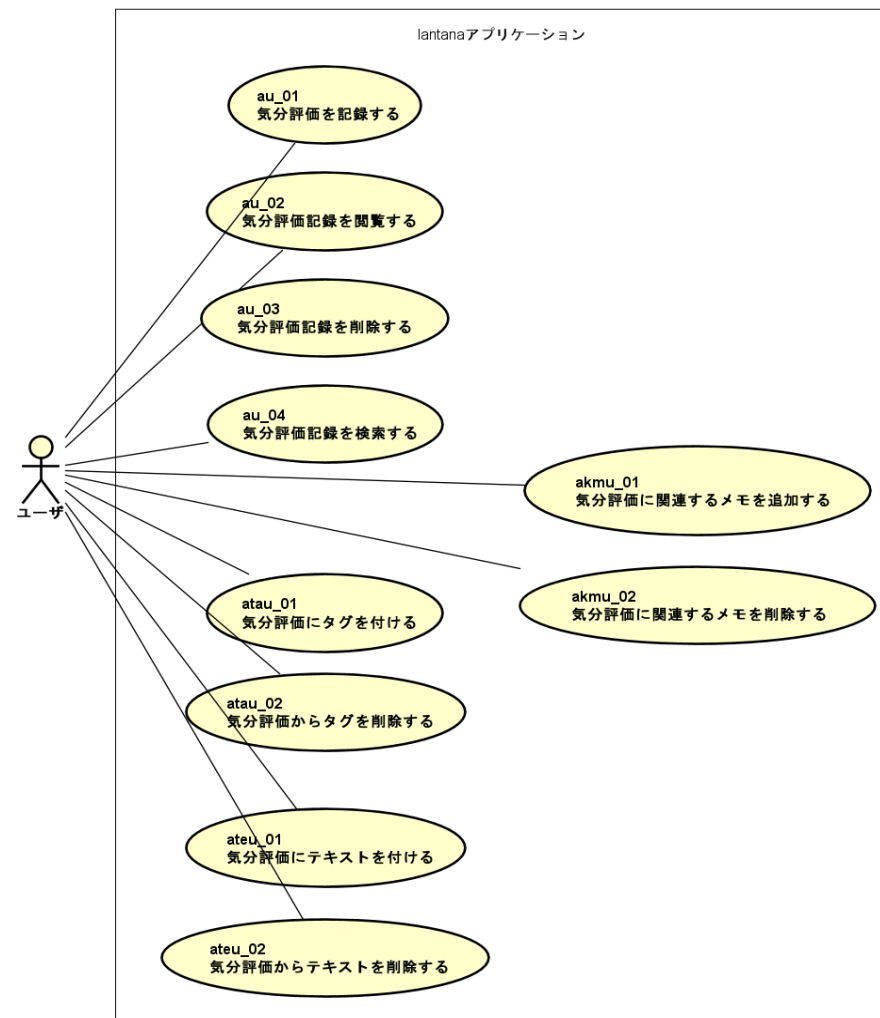
デスクトップ用lantanaアプリケーションランチャー
ランダム時間経過後に、前回記録から一定時間経過していればlantanaを起動する。
Go言語で実装する。

hato

Android用lantanaアプリケーションランチャー
画面ロック解除時に、前回記録から一定時間経過していればlantanaを起動する。
Taskerで実装する。

ユースケース図

左の要求をユースケース図に起こした（下図）。



index

このドキュメントの読み方

シート

左シートから順に見て行ってください。
右に行けば行くほど実装に近づいていきます。

図

上から下へ、左から右へ見てください（なるべく整列しました）

シートへのリンクと概要

シートタイトル	説明
lantana	開発動機・要件・要求
index	ドキュメント内リンク。このページ
開発環境	開発に使用する言語・フレームワークの一覧
論理ER図	論理データベース設計。説明用
サーバサイド概要クラス図	サーバサイド重要クラス名一覧。説明用
画面概要クラス図1	画面重要クラス名一覧。説明用
画面概要クラス図2	画面重要クラス名一覧。画面重要コンポーネント。説明用
サーバサイド論理クラス図	サーバサイド論理クラス図。説明用
画面論理クラス図	画面論理クラス図。説明用
物理ER図	物理ER図。実装対応
サーバサイド物理クラス図	サーバサイド物理クラス図。実装対応
画面物理クラス図1	画面物理クラス図。通信用データ型定義。実装対応
画面物理クラス図2	画面物理クラス図。画面コンポーネント定義。実装対応
シーケンス図1	Lantana画面初期化までのシーケンス図
シーケンス図2	LantanaLogViewerPage表示までのシーケンス図
シーケンス図3	LantanaLogViewerPageのLantanaDetailView表示までのシーケンス図
APIアドレス一覧	サーバAPIアドレスとリクエスト・レスポンスデータ型一覧
所感・著者紹介・リンク	所感・著者紹介・リンク

開発環境

エディタ

- Visual Studio Code (Vue.js開発)
- Vim (Go言語開発)

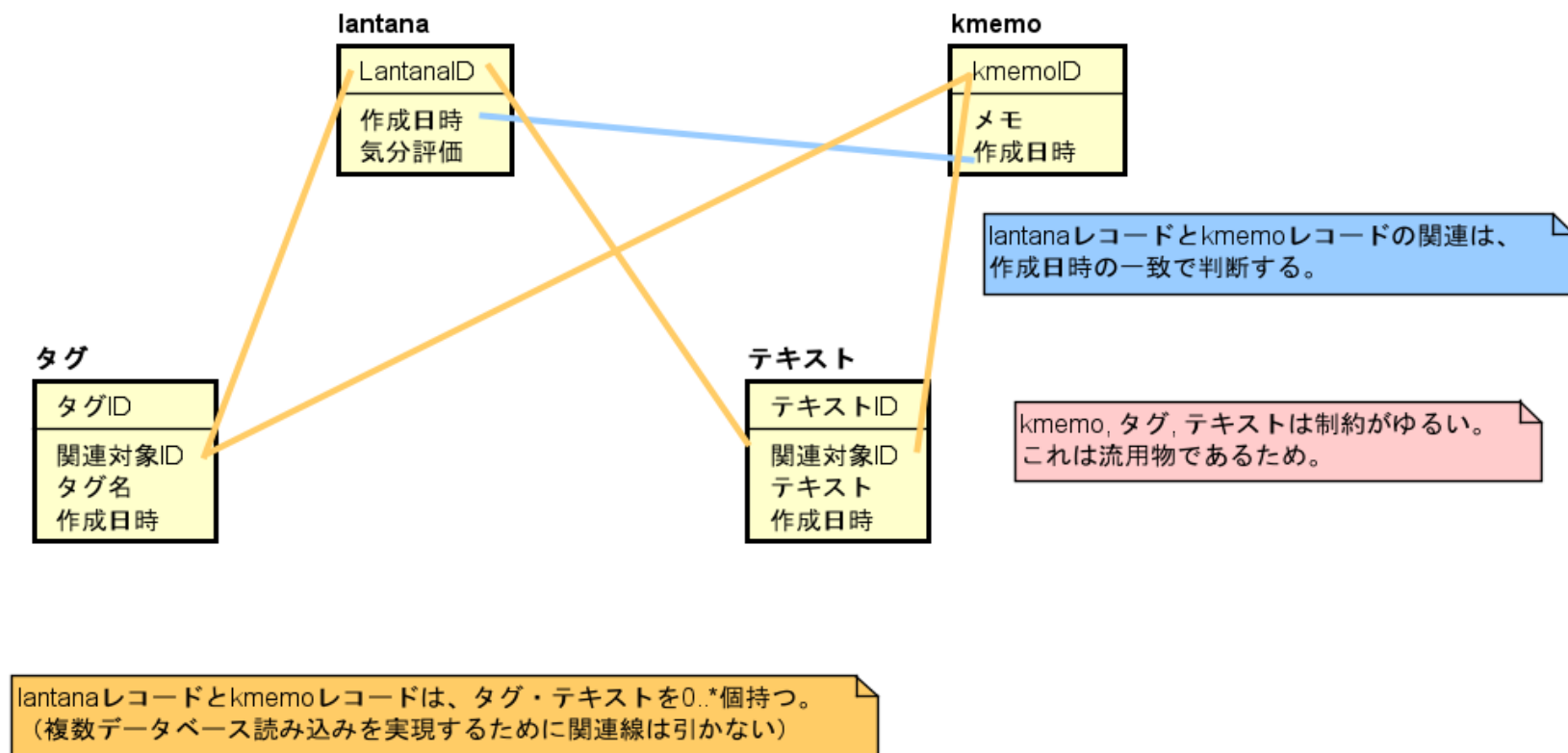
言語

- SQLite (データベース)
- Go言語 (サーバ)
- Type Script (画面処理)
- Vue.js (画面・画面処理)
- Vuetify (画面・画面処理)

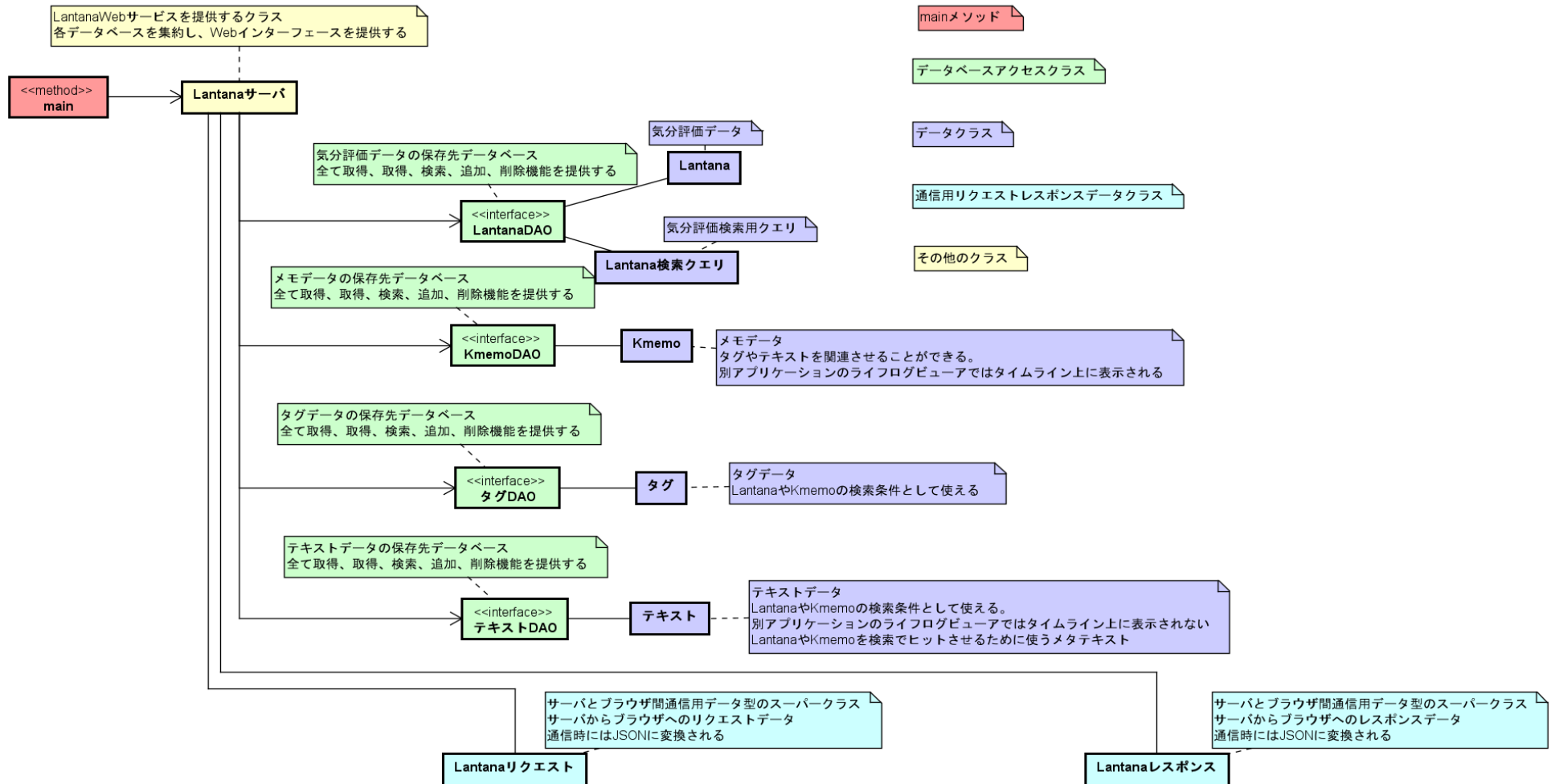
ツール

- Astah* (モデル作成)

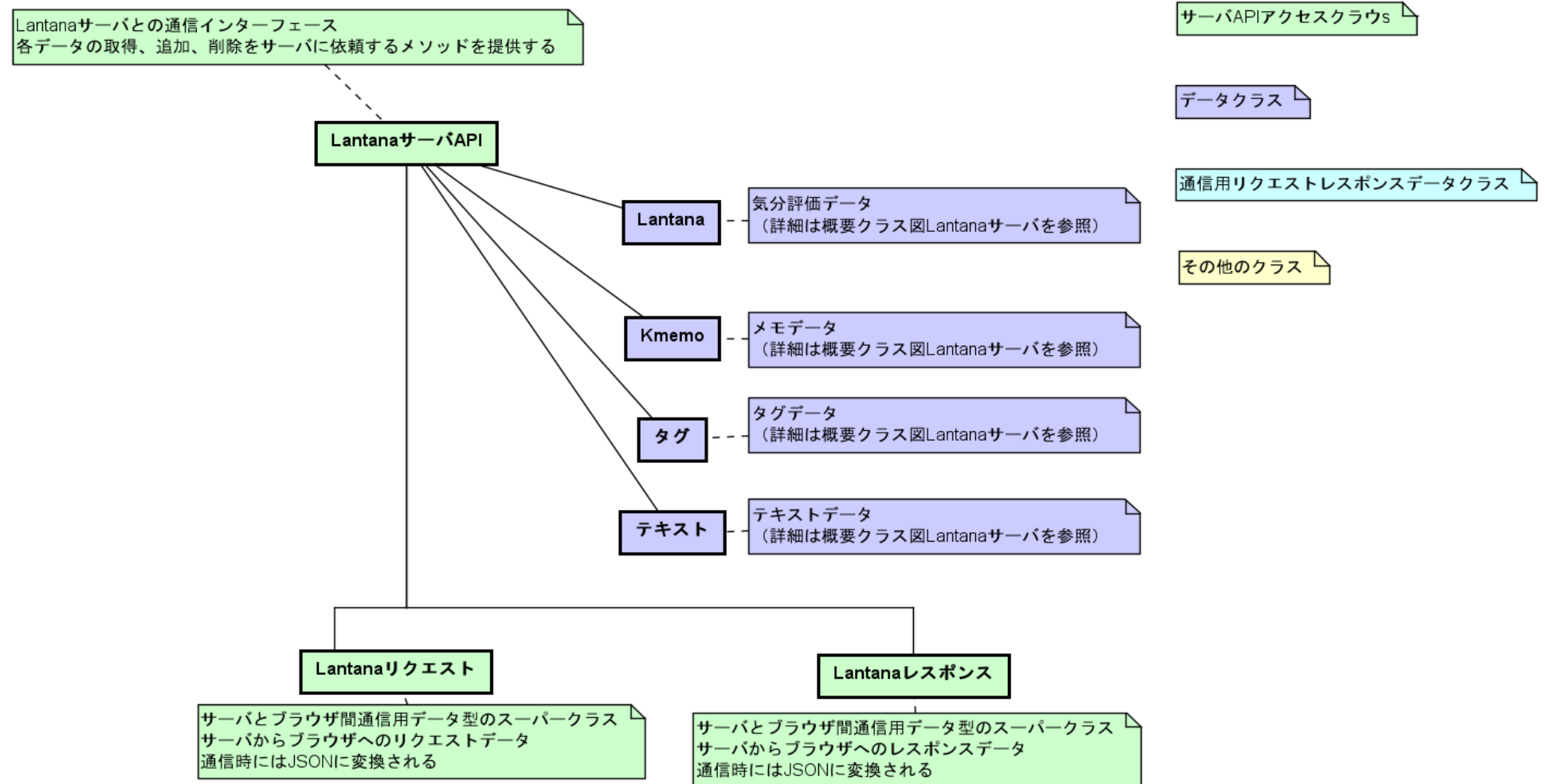
論理ER図



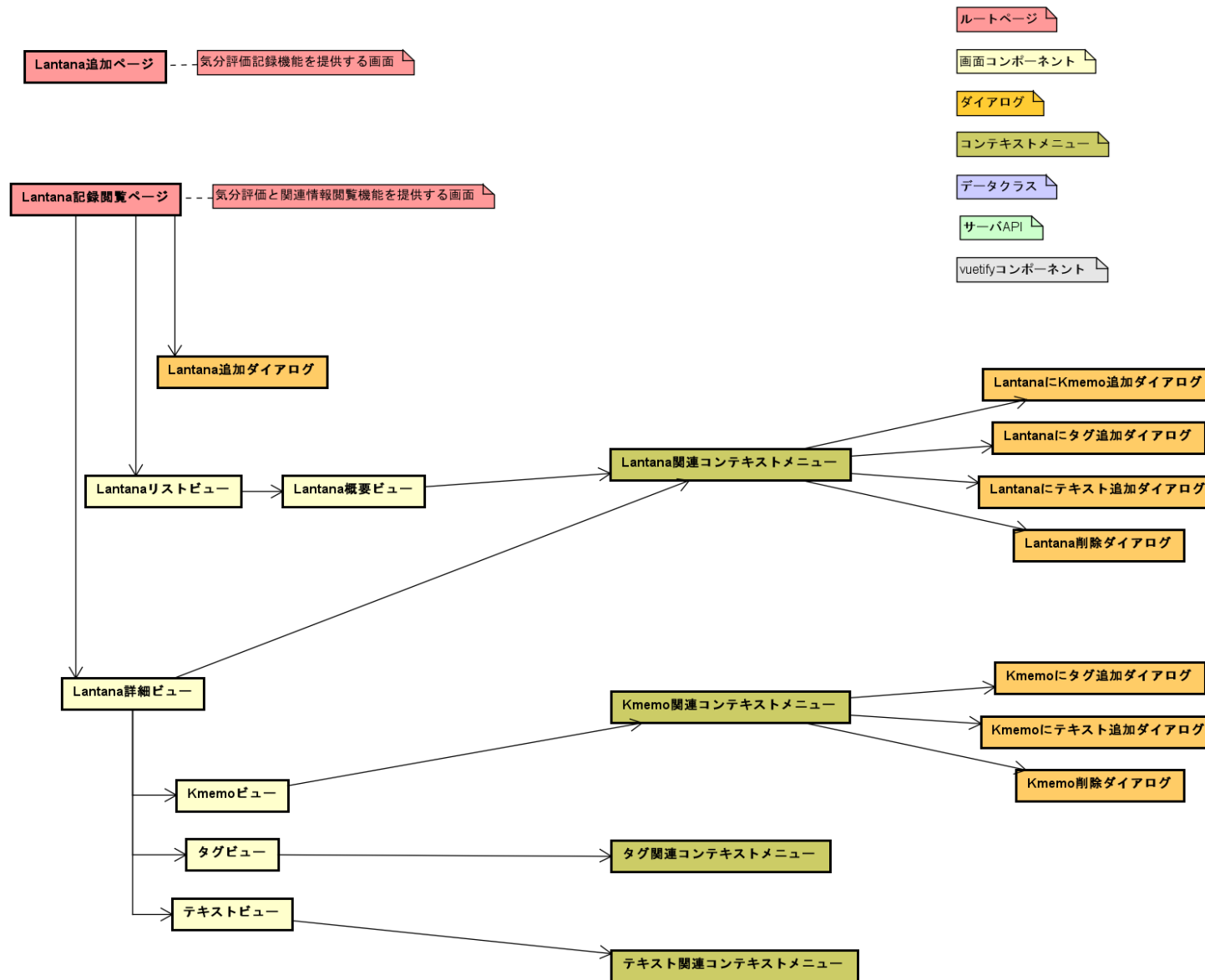
サーバサイド概要クラス図



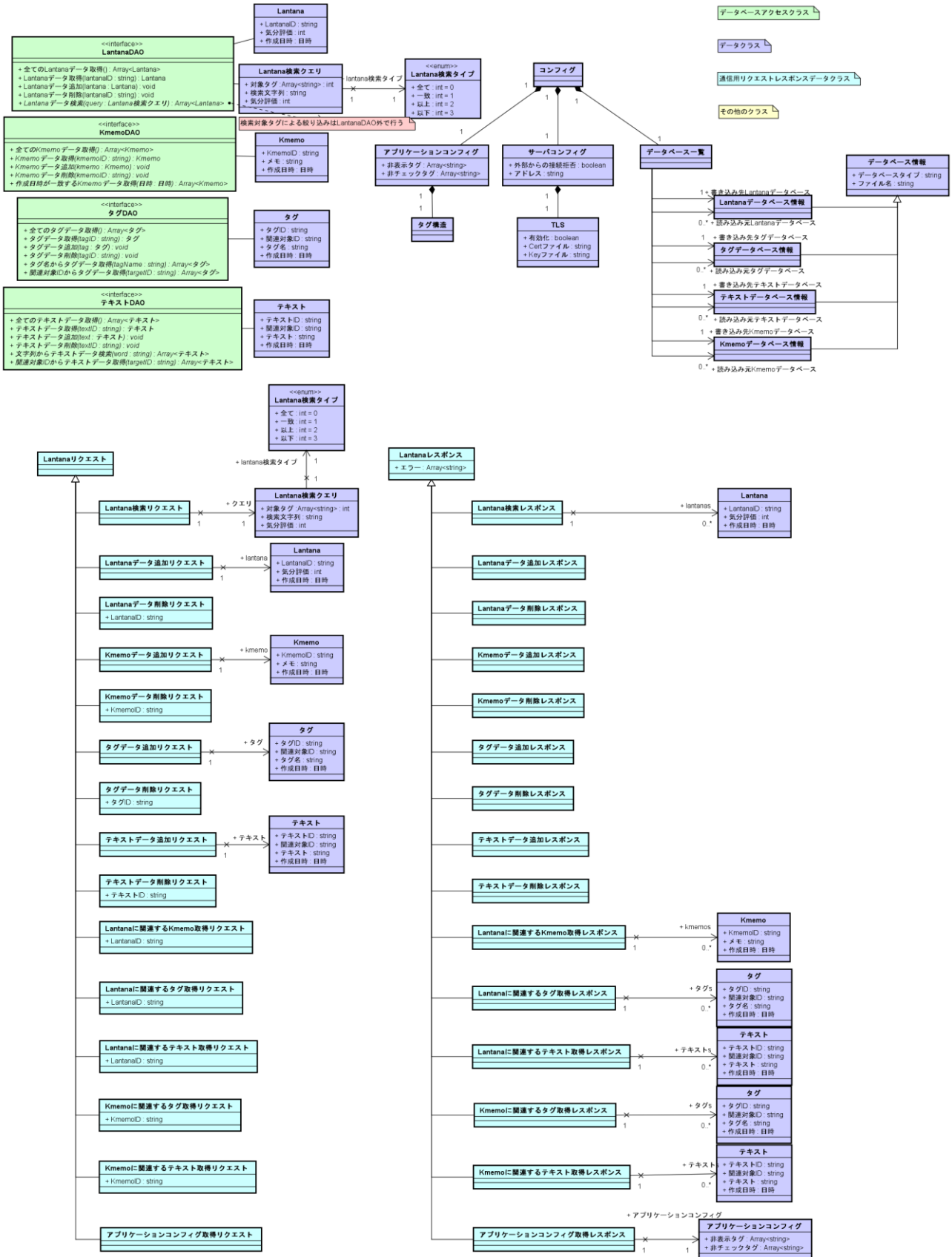
画面概要クラス図



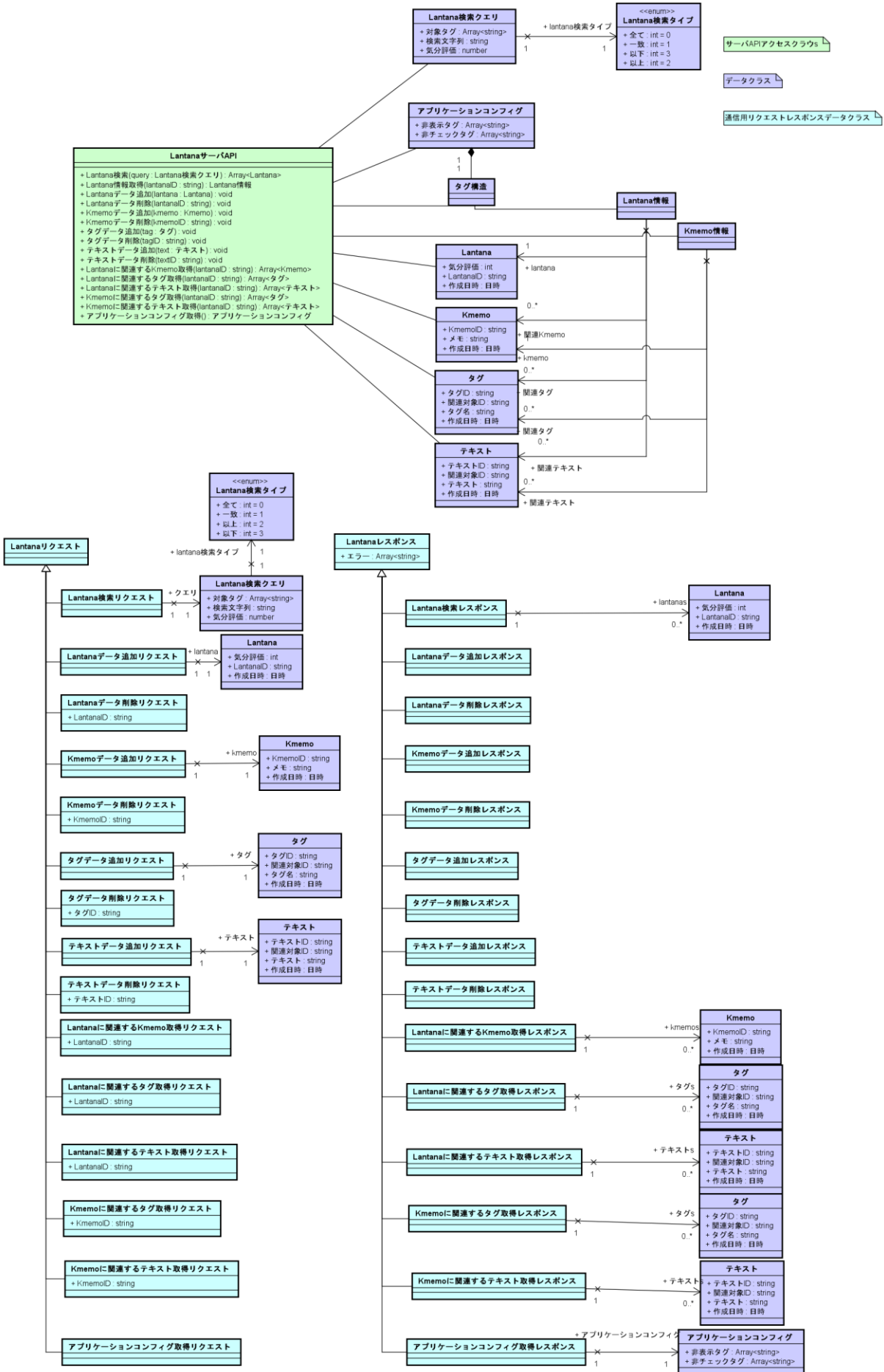
画面概要クラス図



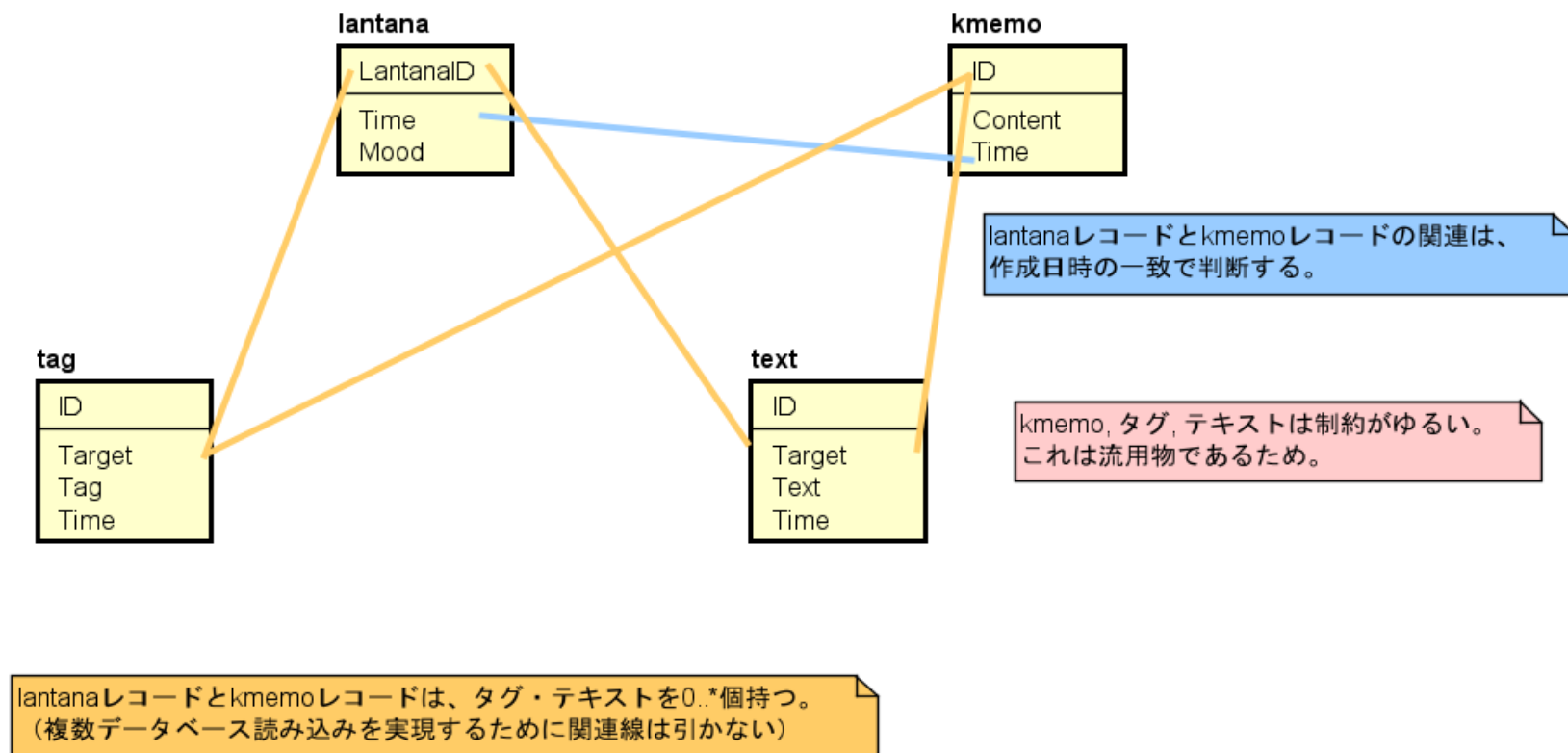
サーバサイド論理クラス図



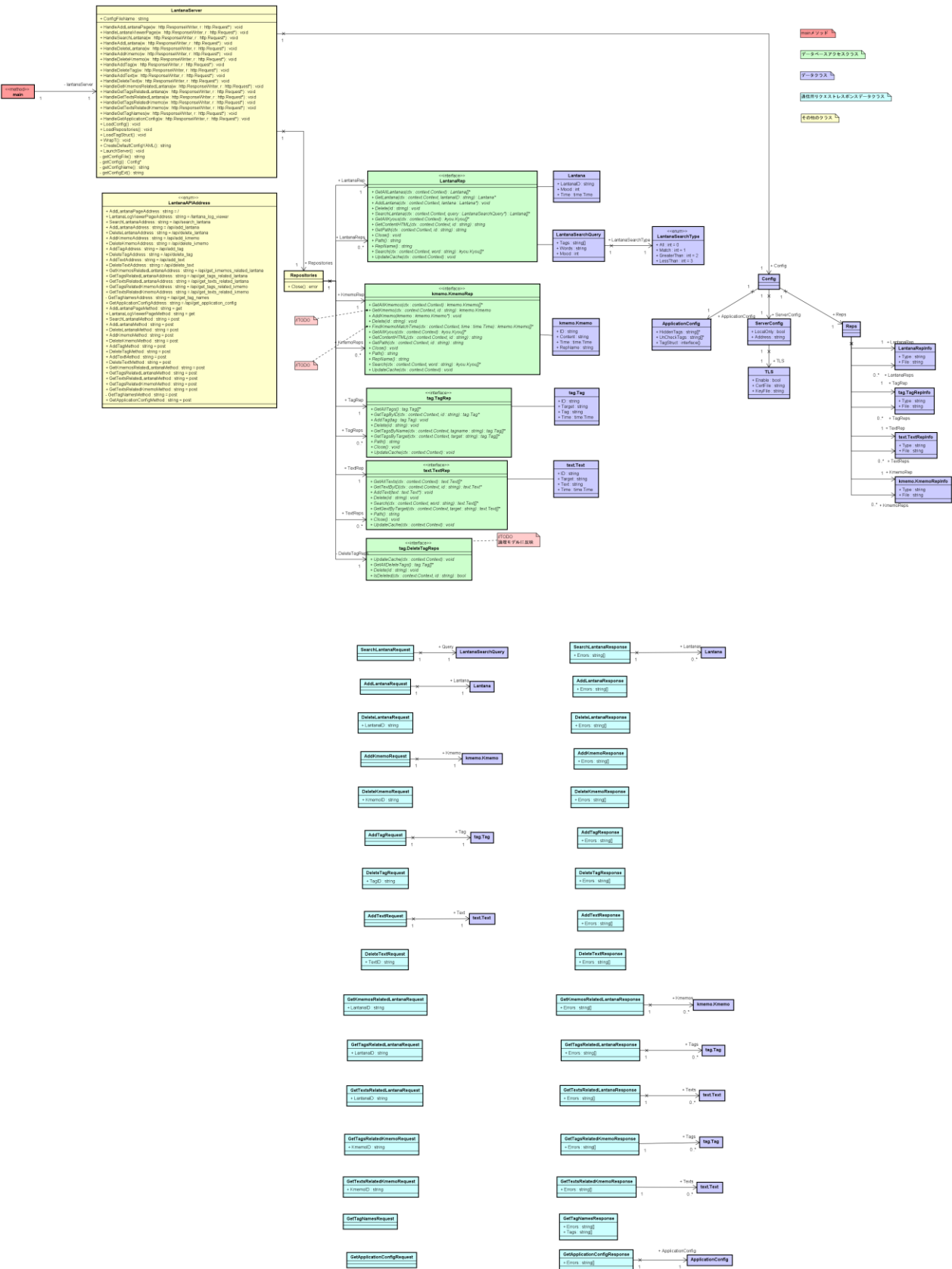
画面論理クラス図



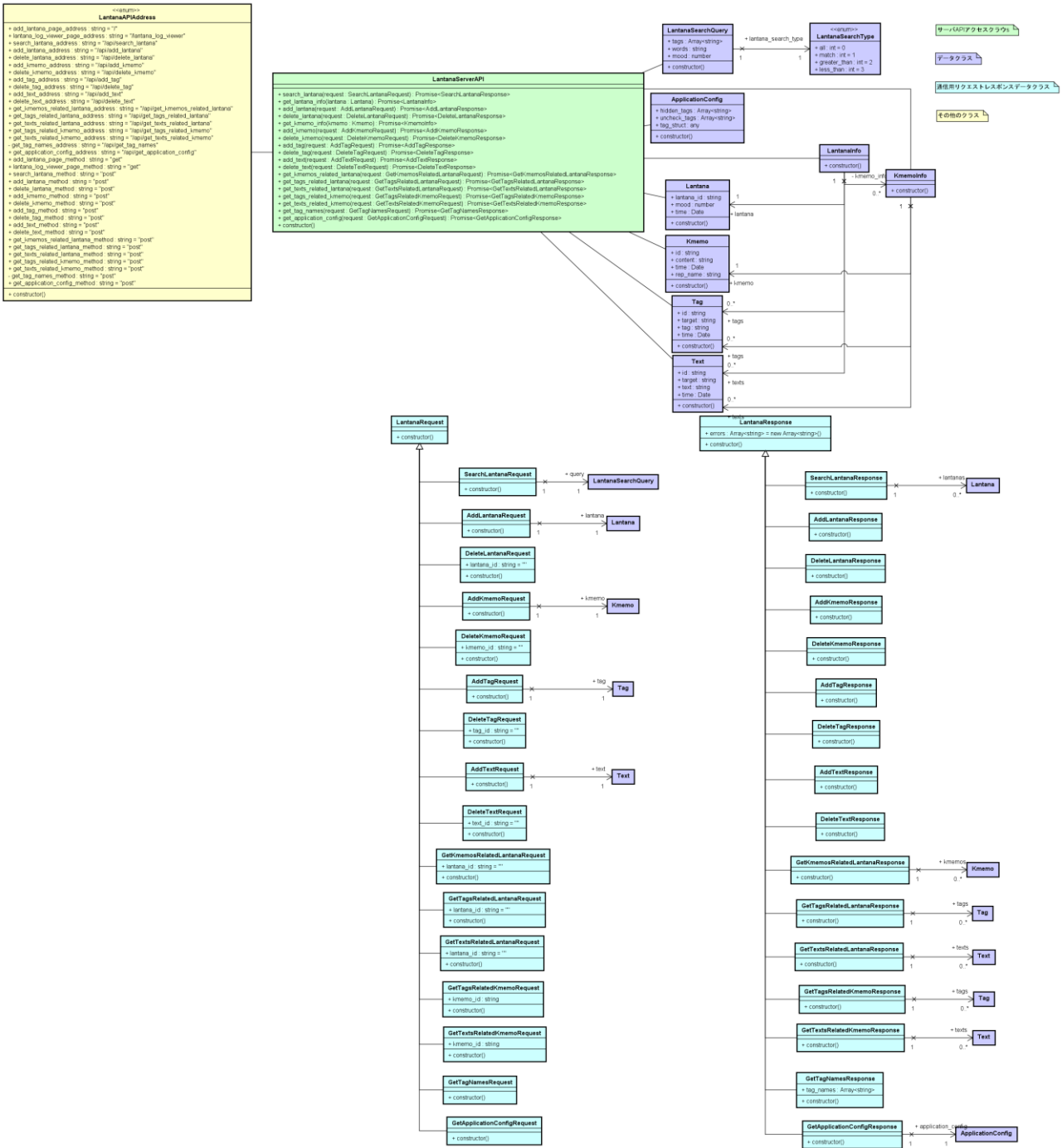
物理ER図



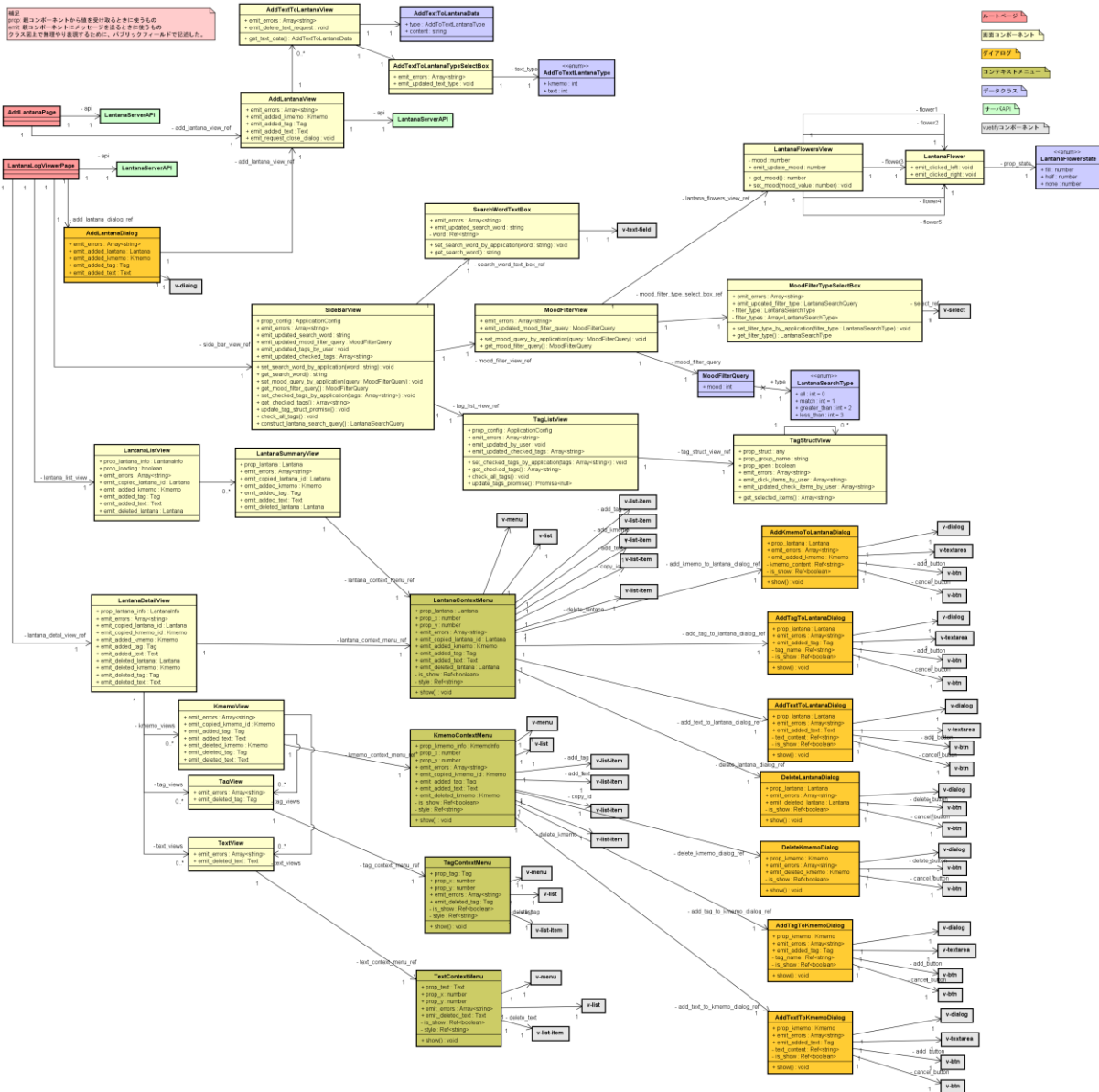
サーバサイド物理クラス図



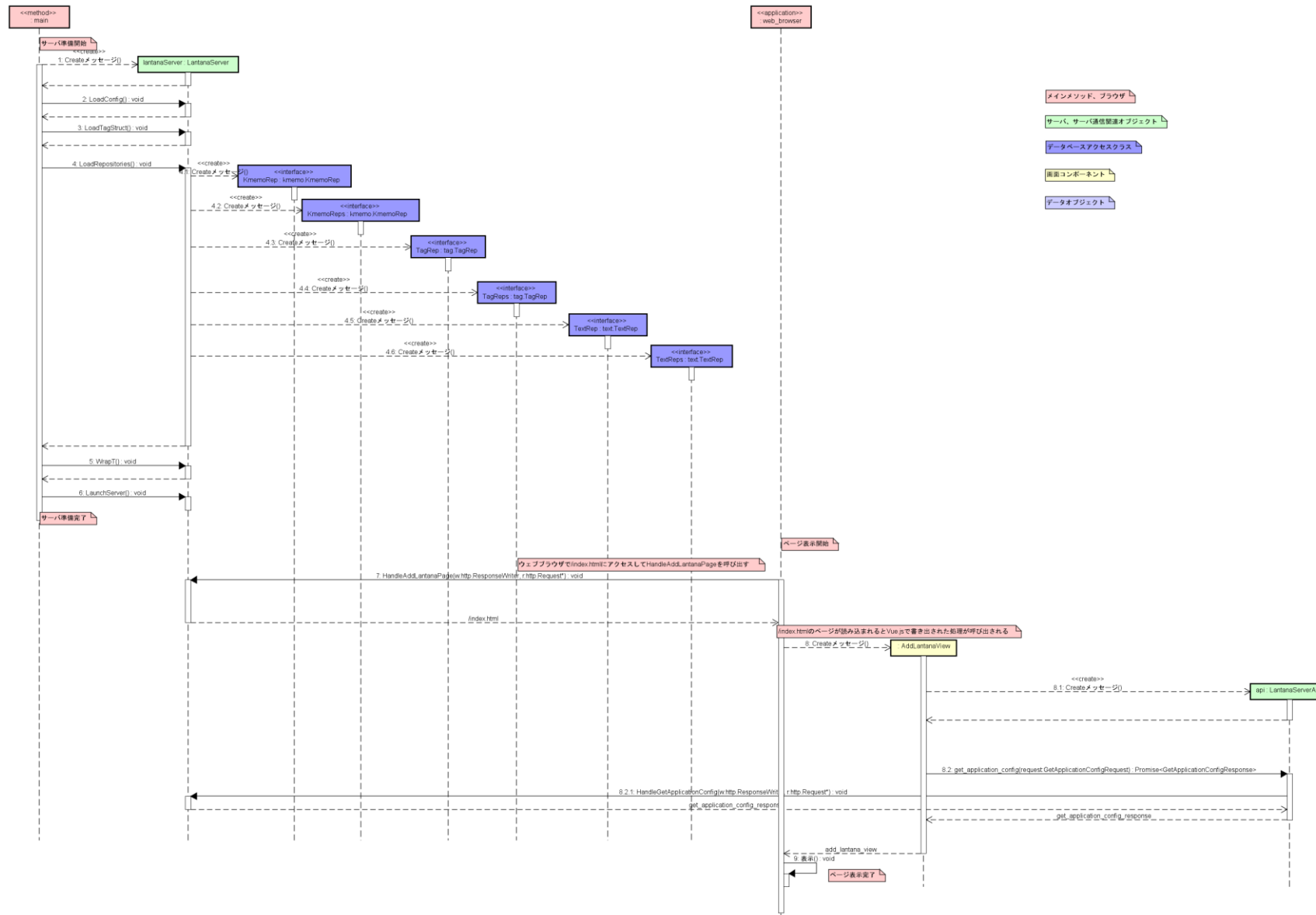
画面物理クラス図



画面物理クラス図



シーケンス図Lantana画面初期化まで



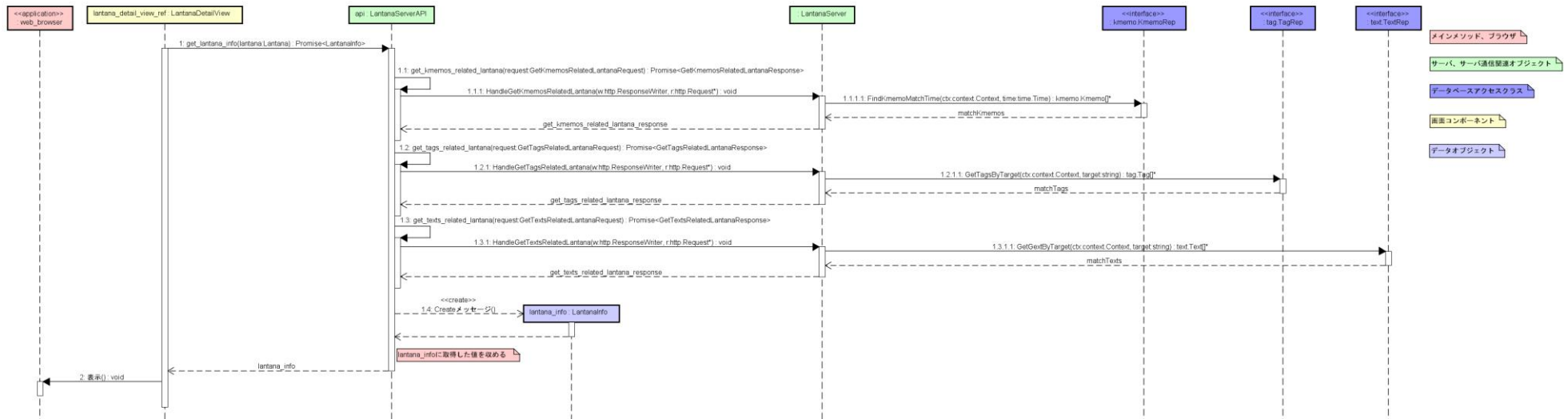
シーケンス図 LantanaLogViewerPage表示まで

Lantana画面初期化までは完了しているものとする



シーケンス図 LantanaLogViewerPageのLantanaDetailView表示まで

LantanaLogViewerPage表示までは完了しているものとする



APIアドレス一覧

用途	アドレス	リクエストデータ型	レスポンスデータ型	メソッド
気分記録画面	/	なし	html	get
気分記録閲覧画面	/lantana_viewer	なし	html	get
Lantana検索	/api/search_lantana	SearchLantanaRequest	SearchLantanaResponse	post
Lantana追加	/api/add_lantana	AddLantanaRequest	AddLantanaResponse	post
Lantana削除	/api/delete_lantana	DeleteLantanaRequest	DeleteLantanaResponse	post
Kmemo追加	/api/add_kmemo	AddKmemoRequest	AddKmemoResponse	post
Kmemo削除	/api/delete_kmemo	DeleteKmemoRequest	DeleteKmemoResponse	post
タグ追加	/api/add_tag	AddTagRequest	AddTagResponse	post
タグ削除	/api/delete_tag	DeleteTagRequest	DeleteTagResponse	post
テキスト追加	/api/add_text	AddTextRequest	AddTextResponse	post
テキスト削除	/api/delete_text	DeleteTextRequest	DeleteTextResponse	post
Lantanaに関連するKmemo取得	/api/get_kmemos_related_lantana	GetKmemosRelatedLantanaRequest	GetKmemosRelatedLantanaResponse	post
Lantanaに関連するタグ取得	/api/get_tags_related_lantana	GetTagsRelatedLantanaRequest	GetTagsRelatedLantanaResponse	post
Lantanaに関連するテキスト取得	/api/get_texts_related_lantana	GetTextsRelatedLantanaRequest	GetTextsRelatedLantanaResponse	post
アプリケーションコンフィグ取得	/api/get_application_config	GetApplicationConfigRequest	GetApplicationConfigResponse	post

所感

所感は適当に書きます。

設計

今回は設計と実装の行き来に制約をかけずに開発しました。ゆるゆるのウォーターフォール開発です。手戻り回数は1度しか発生しなかったのが嬉しいです。前回の短期間ウォーターフォール開発の反省を活かして、ユーザ操作の再生をイメージしながら設計しましたが、それによる結果なのだとしたら、もっと嬉しいです。

実装

今回の実装はまあまあ楽しさでした。評価するならば★★★★☆ぐらいです。前回の実装に比べて難易度が低かったからだと思います。コンポーネントも使い回すことができ爆速で実装できたものの、その分達成感を得られませんでした。最近は実装よりも設計のほうが楽しく感じます。

ドキュメント作成

「情報を網羅的に載せればいい」と思っていた時代が私にもありました。ですが、実は情報網羅はただの前提で、重要な情報をピックアップする力が必要であることに気づきました。「ただ情報を垂れ流されても困る」。私も人生上で何度感じましたが、情報共有をする側の立場になって初めて気づきます。情報共有難しすぎ……。私は頭の回転は遅い方なので、入念に準備することで今後の人生をなんとかかやっていこうと思います。

開発モチベーション

途中から開発モチベーションの低下に悩まされました。「Lantanaが完成したとして、使われるのか？」という思いと、睡眠不足によってです。さらにはLantanaよりも作りたいライフログアプリケーション案を思いついてしまったので、画面実装はグダグダでした。なんとかドキュメント作成完了までたどり着けたのは奇跡です。

ちなみに、Lantanaは使われています。「★☆☆☆☆つかれた」みたいなしょうもない使い方ですが、作って良かったです。

以上です。次のライフログアプリケーション案があるので、Lantana開発が落ち着き次第開発に取り掛かろうと思います。

制作期間

2023-05-19 考案。設計・実装・ドキュメント作成開始

2023-05-31 設計・実装・ドキュメント作成完了

計12日間

著者紹介



mt3hr

Go言語が好き。うなぎを食べるのも好き。

リンク集

- アプリケーション実行ファイル: <https://github.com/mt3hr/lantana/releases/latest>
- ソースコード: <https://github.com/mt3hr/lantana>
- 開発ドキュメント: <https://github.com/mt3hr/lantana/tree/main/document>
- 設計図: <https://github.com/mt3hr/lantana/tree/main/document/model>