**アプリの全体設計およびガイドライン**

**１．アプリの概要**

本アプリは、最低限の機能として画像を機械学習のモデルを用いて分類してタグを付け、データベースでタグを検索する機能を有する。

ランクBの追加機能として、ログイン機能を備える。

ランクCの追加機能として、画像にリンクしたテキスト情報をデータベースで管理し、テキスト検索する機能を備える。

**２．機械学習モデル**

　既存の人工知能の画像分類問題のモデルを利用する。学習に用いる画像のデータセットは、自ら収集するのではなく、既存の機械学習用の画像データセットを用いる。既存の画像データセットは、すでに画像のラベルが付いているものを用いる。

　画像分類の種類は、開発の難易度を考慮し、２～５種類程度とする。

**３．実装のガイドライン**

　バックエンドは以下のサイトのコードをベースにFlaskを用いて開発する。

<https://www.python.ambitious-engineer.com/archives/1640>

<https://github.com/Atsuhiko/Web-App/tree/master/FlaskApp/analyzer>

　フロントエンドは、部分的であっても Vue.js を用いる。

　データベースは MySQL を用いる。

　ローカルPC上のサーバーで動作することを最終目標とする。クラウドへのデプロイは今回の開発では見送る。

**詳細仕様**

**１．サーバー構築方法（担当：井伊）**

**1-1. 使用言語・フレームワーク**

Python, Flask

参考Webサイト／コード：

<https://www.python.ambitious-engineer.com/archives/1640>（主に使用予定）

<https://github.com/Atsuhiko/Web-App/tree/master/FlaskApp/analyzer> (コード)

<https://flask.palletsprojects.com/en/1.1.x/tutorial/#tutorial> (参考)

**1-2. ログイン機能**（オプション、ランクB）

**2．機械学習モデル（担当：井伊）**

**2-1.データセット**：

Kaggle の過去コンペ「Dogs vs. Cats」(<https://www.kaggle.com/c/dogs-vs-cats/overview>) のデータセットで分類モデルを作る。（理由：データセットを探してタグ付けをする必要がない。モデル作成のお手本のコードがすでに大量にある。→　開発時間短縮になるから）

**2-2. 機械学習と新しいデータの予測のコーディング方法**

モデル作成は、アプリのコードとは別途行い、学習済みモデルを保存する。アプリのコードでは、「モデルのフレームの構築」、「学習済みモデルの読み込み」および「新しいデータの予測」をコードする。

**3．データベース（担当：張）**

**3-1. 使用DB**

MySQL

**3-2. Python コードとの連携方法**

技術調査後に決定（張さん担当）

**3-3. データベース項目（案：最終設計は張さん担当）**

ID、画像データ格納URL、画像タグ、画像説明テキスト（オプション、ランクB）、データ登録日時（オプション、ランクC）、ユーザー情報（オプション、ランクB）

**3-4. 検索エンジン**

タグの検索、画像説明テキストのワード検索（オプション、ランクB）

**3-5. データ入力方法**

張が主担当。井伊がサポート

**４．フロントエンド（担当：柿埜）**

**4-1. 使用言語・フレームワーク**

HTML, CSS, JavaScript, Vue.js (無理をしても、Vue.jsは必ず使う)

**4-2. 機能デザイン作成**

ページ構成、ページ間のリンク、各種オブジェクトの配置

**4-3. 精密デザイン**

アプリの最終画面をシミュレーションした資料を作成