要件定義書

動物保護マッチングアプリ

「あにマッチ」

目次

[1.改定履歴 3](#_Toc88645238)

[2.機能概要 4](#_Toc88645239)

[機能①　動物閲覧機能 4](#_Toc88645240)

[機能②　絞り込み機能 4](#_Toc88645241)

[機能③　ユーザー登録機能 4](#_Toc88645242)

[機能④　ログイン機能 4](#_Toc88645243)

[機能⑤　ログアウト機能 4](#_Toc88645244)

[機能⑥　メッセージ投稿機能 5](#_Toc88645245)

[機能⑦　メッセージ表示機能 5](#_Toc88645246)

[機能⑧　エラー表示機能 5](#_Toc88645247)

[3.対象ユーザー 6](#_Toc88645248)

[4.制約事項 7](#_Toc88645249)

[5.操作手順 8](#_Toc88645250)

[6.画面遷移図 11](#_Toc88645251)

[7.基本アーキテクチャ図 12](#_Toc88645252)

[8.テーブル定義書 18](#_Toc88645253)

[9.クラス図 20](#_Toc88645254)

[10.クラス定義書 21](#_Toc88645255)

# 改定履歴

|  |  |  |  |
| --- | --- | --- | --- |
| 作成日 | 作成者 | 改定内容 | 版 |
| 2021/11/22 | 片山 |  | 初版 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 2.機能概要

本アプリケーションは、動物の里親情報を閲覧・検索し、コメント機能でやり取りできることを目的としている。

### 機能①　動物閲覧機能

・データベースに登録されている全ての動物の情報「性別」、「年齢」、「(譲渡)場所」、「画像」、「説明文」を表示する。

・動物は登録された順番に表示する。

### 機能②　絞り込み機能

・動物の性別・年齢・場所(譲渡エリア)の条件選択により、画面に表示する動物を絞り込む。

### 機能③　ユーザー登録機能

・ユーザー登録をする。

・入力されたユーザーIDとパスワードと名前をもとにユーザー登録を行う。

・ユーザーIDとパスワードと名前はデータベースに登録される。

・登録するには同意書に同意する必要がある。

### 機能④　ログイン機能

・メッセージ機能を利用するためにアプリケーションにログインする。

・入力されたユーザーIDとパスワードをもとにユーザー認証を行う。

・ユーザーIDとパスワードはデータベースに登録されたものを使う。

### 機能⑤　ログアウト機能

・アプリケーションからログアウトする。

機能概要

### 機能⑥　メッセージ投稿機能

・短い文字列情報を「メッセージ」として投稿できる。

・選択した動物ごとに「メッセージ」を投稿できる。

・投稿が空の場合、投稿を受け付けずエラーメッセージを表示する。

・投稿した「メッセージ」はアプリケーションが終了するまで保持される。

・この機能を使用するにはログインしている必要がある。

### 機能⑦　メッセージ表示機能

・選択した動物ごとの「メッセージ」を表示できる。

### 機能⑧　エラー表示機能

・絞り込み条件に当てはまるものがなかった際、エラーメッセージを表示する。

・メッセージ入力画面上で、テキスト入力せず空の状態で「送信」ボタンを押下した際、

　エラーメッセージを表示する。

# 3.対象ユーザー

動物を譲渡したい、里親になりたい教室内のユーザー（全年齢対象）

# 4.制約事項

特になし

# **5.操作手順**

**URL**

<http://localhost:8080/aniMatch/>

**トップ画面(タイトル画面)**

肉球型のボタンを押すとメイン画面へ遷移する。

**メイン画面**

動物の画像と情報が表示される。

動物ごとに付与されている「メッセージ」ボタンを押すとログイン画面に遷移する。

「絞り込み」ボタンを押すと絞り込み画面に遷移する。

絞り込まれた動物の画像と情報が表示される。

絞り込み条件に当てはまるものがない場合、エラーメッセージが表示される。

「戻る」ボタンを押すと絞り込み画面に遷移する。

**絞り込み画面**

動物の情報「性別」「年齢」「(譲渡)場所」の絞り込み条件を指定する。

「絞り込み」ボタンを押すとメイン画面に遷移する。

操作手順

**ログイン画面**

「ユーザーID」と「パスワード」を入力し、

「ログイン」ボタンを押すとログイン結果画面へ遷移する。

「新規登録」ボタンを押すとユーザー登録画面へ遷移する。

**ログイン結果画面(ログイン成功)**

ユーザーの名前とログイン中であることが表示される。

「メッセージ入力」ボタンを押すとメッセージ入力画面へ遷移する。

「ログアウト」ボタンを押すとログアウト確認画面へ遷移する。

**ログイン結果画面(ログイン失敗)**

ログインが失敗したことが表示される。

「トップへ」ボタンを押すとメイン画面へ遷移する。

**ログアウト確認画面**

ログアウトを確認するメッセージが表示される。

「はい」ボタンを押すとログアウト結果画面へ遷移する。

「いいえ」ボタンを押すと「ログアウト」ボタンを押した画面に遷移する。

**ログアウト結果画面**

ログアウト完了メッセージが表示される。

「TOPへ」ボタンを押すとトップ画面へ遷移する。

操作手順

**メッセージ入力画面**

ユーザー名とログイン中であることが表示される。

「メッセージ」を入力し、「送信」ボタンを押すと「メッセージ」が表示される。

「更新」ボタンを押すと最新のメッセージ投稿の情報に更新される。

「ログアウト」ボタンを押すとログアウト画面に遷移する。

**ユーザー登録画面**

登録するユーザーの「名前」「ユーザー名」「パスワード」を入力し、

「確認」ボタンを押すと登録確認画面に遷移する。

**登録確認画面**

登録するための「名前」「ユーザー名」「パスワード」が表示される。

同意書が表示される。

「同意して登録する」ボタンを押すとログイン結果画面(登録完了)に遷移する。

「修正」ボタンを押すとユーザー登録画面に遷移する。

**ログイン結果画面(登録完了)**

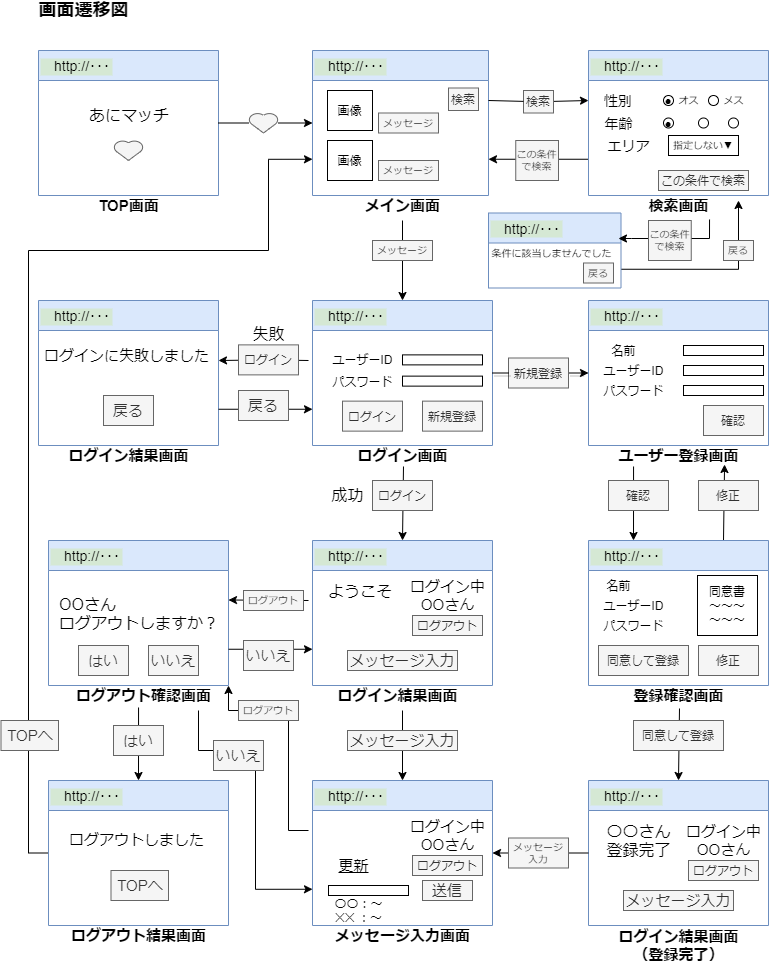
登録完了メッセージが表示される。

ユーザー名とログイン中であることが表示される。

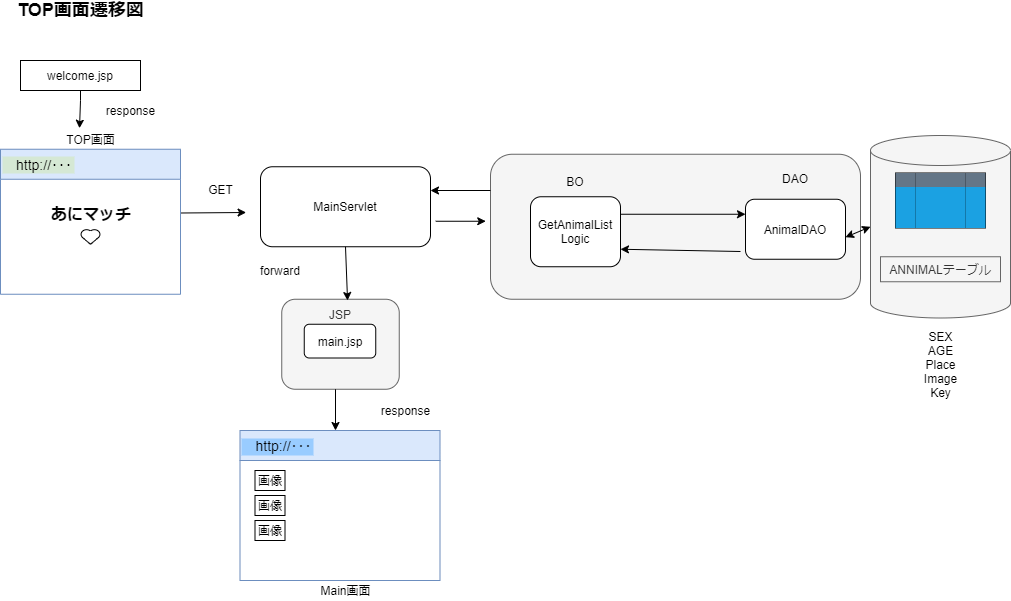
「メッセージ入力」ボタンを押すとメッセージ入力画面へ遷移する。

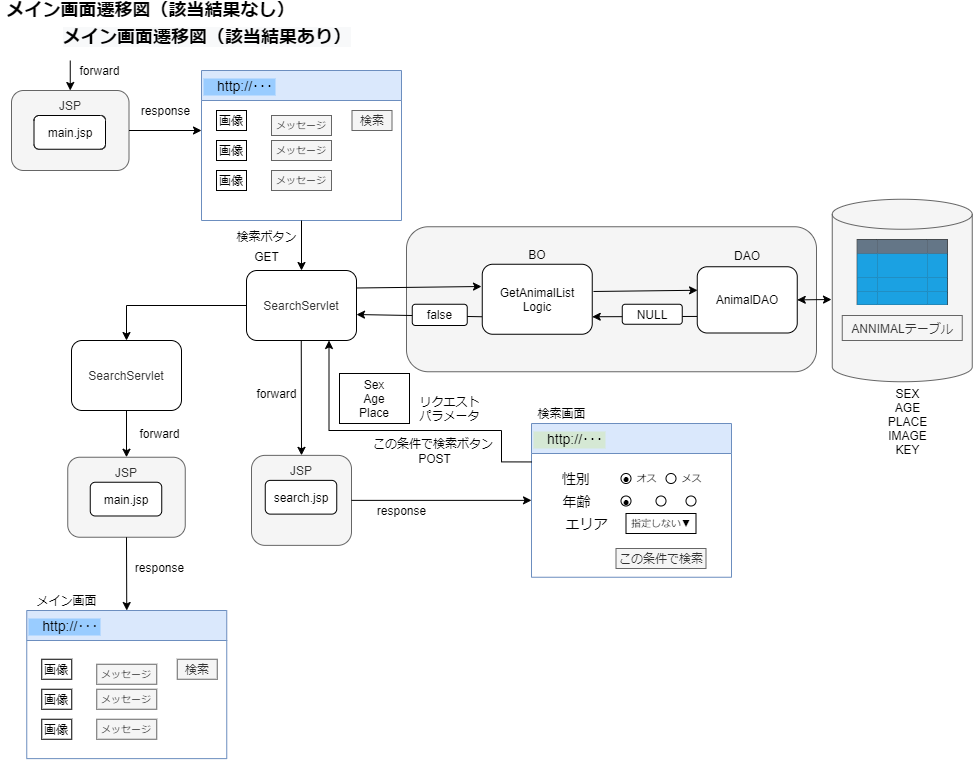
「ログアウト」ボタンを押すとログアウト画面に遷移する。

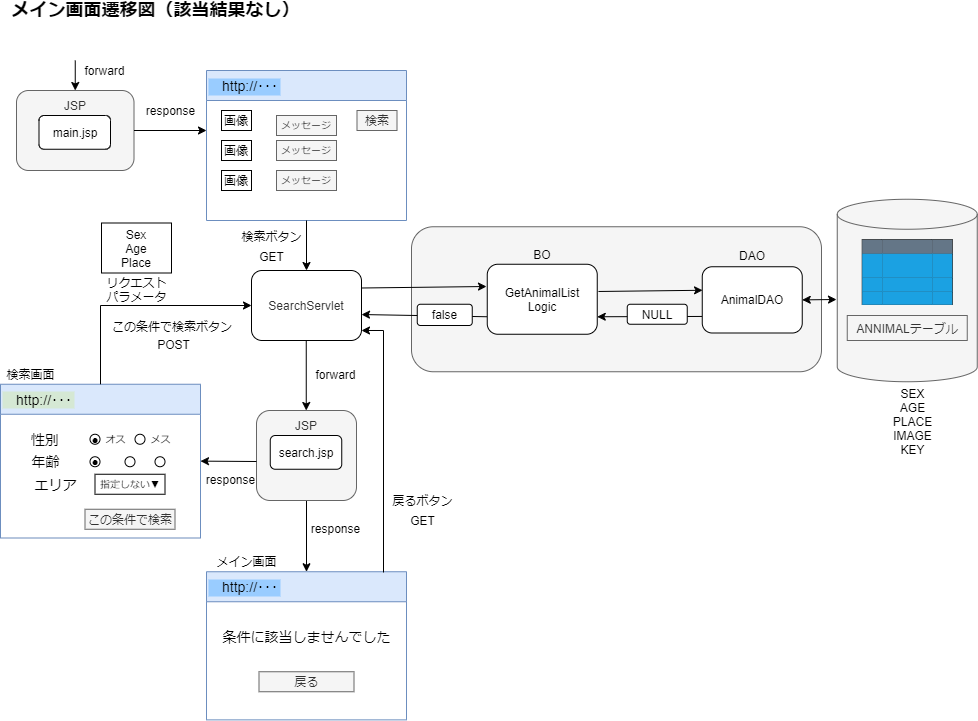
# **6.画面遷移図**

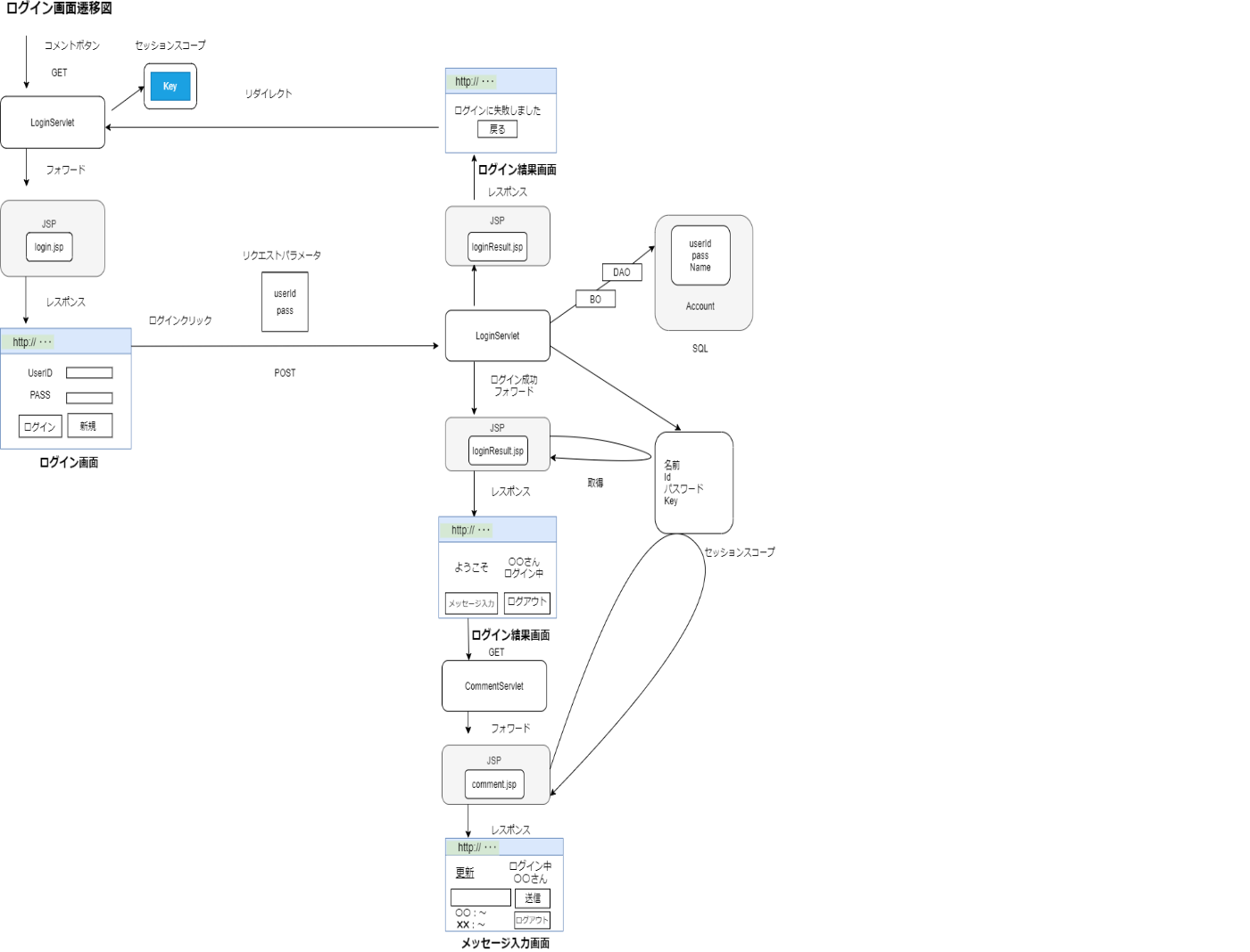


# **7.基本アーキテクチャ図**



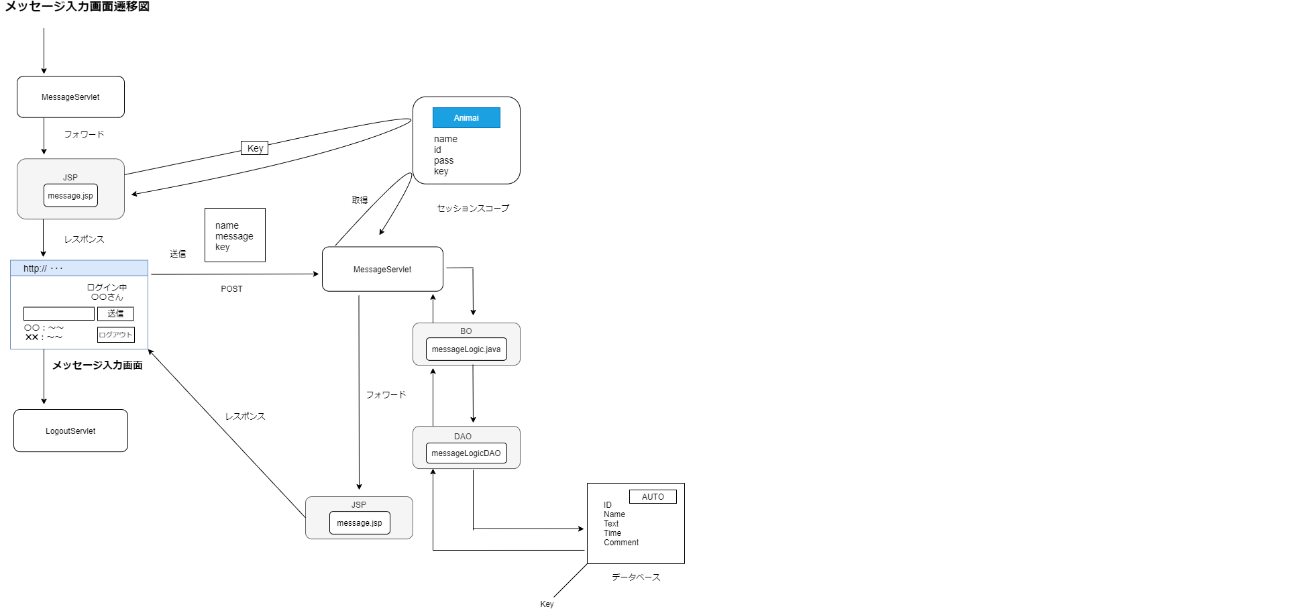






グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明



# **8.テーブル定義書**

①「ACCOUNT」テーブル

**テーブル情報**



**カラム情報**



②「message」テーブル

**テーブル情報**



**カラム情報**



③「Animalテーブル」

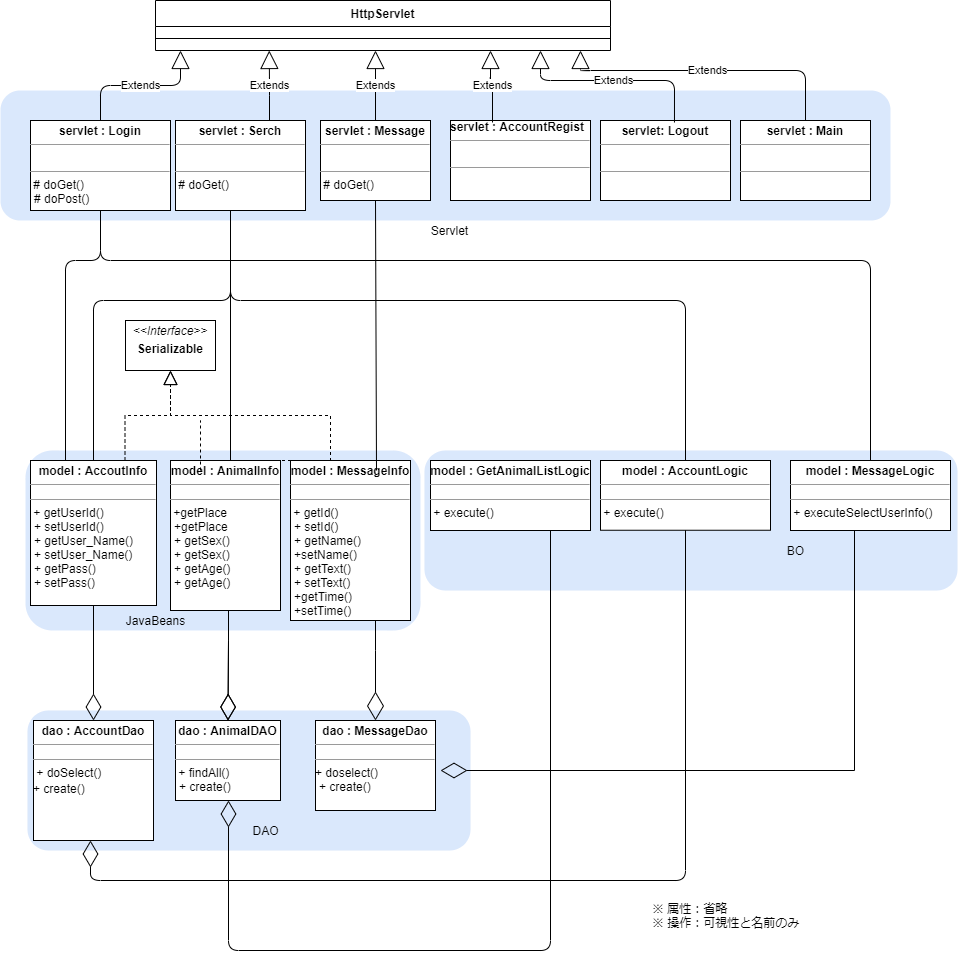
**テーブル情報**



**カラム情報**



# **9.クラス図**



# 10.

クラス定義書

【DAO】

クラス定義書DAO①

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | |
| 機能 | 「ACCOUNT」テーブルを担当する | | | | | | |
| クラス名 | **AccountDAO** | | | | | | |
| パッケージ | dao | | | | | | |
| 可視性 | public | | | | | | |
| 継承クラス |  | | | | | | |
| インポート | java.sql.Connection  java.sql.DriverManager  java.sql.PreparedStatement  java.sql.ResultSet  java.sql.SQLException  java.util.ArrayList  java.util.List  model.AccountInfo | | | | | | |
| サブクラス |  | | | | | | |
| フィールド | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | JDBC\_URL | private | String | jdbc:mysql://localhost/test\_db?characterEncoding=UTF-8 | データベース  （以下DB）のURL |
| 2 | DB\_USER | private | String | test\_user | DBのユーザーID |
| 3 | DB\_PASS | private | String | test\_user | DBのパスワード |
| 4 | accountInfoList | public | List<AccountInfo> | newArrayList<>(); | ArrayListインスタンスの生成 |
| 5 | conn | public | Connection | DriverManager.getConnection(JDBC\_URL, DB\_USER, DB\_PASS) | DBへの接続情報 |

**次ページへ続く**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | | aniMatch | | | | | | | | |
| 機能 | | 「ACCOUNT」テーブルを担当する | | | | | | | | |
| クラス名 | | **AccountDAO** | | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | | 型 | | 初期値 | 備考 | |
| 6 | sql | public | | String | | "SELECTUSER\_ID,  PASS,USER\_NAME FROM ACCOUNT" | ユーザー情報を問い合わせるSQL文 | |
| 7 | pStmt | public | | PreparedStatement | | conn.prepareStatement(sql) | SQL発行用オブジェクト | |
| 8 | rs | public | | ResultSet | | pStmt.executeQuery | SQL抽出結果 | |
| 9 | userId | public | | String | | rs.getString("USER\_ID") | DBのUSER\_ID（ユーザーID） | |
| 10 | pass | public | | String | | rs.getString("PASS") | DBのPASS（パスワード） | |
| 11 | userName | public | | String | | rs.getString("NAME”) | DBのNAME(ユーザー名) | |
| 12 | accountInfo | public | | AccountInfo | | new AccountInfo(userId, pass, userName) | 上記9～11の属性を持つ  AccountInfoインスタンスの生成 | |
| 13 | con | public | | Connection | | DriverManager.getConnection(JDBC\_URL, DB\_USER, DB\_PASS) | DBへの接続情報 | |
| 14 | ps | public | | PreparedStatement | | null | SQL発行用オブジェクト  格納用 | |
| 15 | rs | public | | ResultSet | | null | SQL抽出結果格納用 | |
| 16 | buf | public | | StringBuffer | | new StringBuffer() | SQL文の生成 | |
| メソッド | No | | 名称 | | 属性  （可視性） | | 型 | 初期値 | | 備考 | |
| 1 | | findAll | | public | | void | List<AccountInfo> | | DB接続、動物の主キーを指定した全レコードの取得 | |
| 2 | | doSelect | | public | | String inputUserId  String inputPass | AccountInfo | | DB接続、対象のレコード  （ユーザーデータ）を取得 | |
| 3 | | create | | public | | AccountInfo accountInfo | boolean | | レコードの追加 | |

クラス定義書DAO②

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | |
| 機能 | 「ANNIMAL」テーブルを担当する | | | | | | |
| クラス名 | **AnimalDAO** | | | | | | |
| パッケージ | dao | | | | | | |
| 可視性 | public | | | | | | |
| 継承クラス |  | | | | | | |
| インポート | java.sql.Connection  java.sql.DriverManager  java.sql.PreparedStatement  java.sql.ResultSet  java.sql.SQLException  java.util.ArrayList  java.util.List  model.AnimalInfo | | | | | | |
| サブクラス |  | | | | | | |
| フィールド | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | JDBC\_URL | private | String | jdbc:mysql://localhost/test\_db?characterEncoding=UTF-8 | データベース  （以下DB）のURL |
| 2 | DB\_USER | private | String | test\_user | DBのユーザーID |
| 3 | DB\_PASS | private | String | test\_user | DBのパスワード |
| 4 | animalInfoList | public | List<AnimalInfo> | newArrayList<>(); | ArrayListインスタンスの生成 |
| 5 | conn | public | Connection | DriverManager.getConnection(JDBC\_URL, DB\_USER, DB\_PASS) | DBへの接続情報 |

**次ページへ続く**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | | aniMatch | | | | | | | | |
| 機能 | | 「Animal」テーブルを担当する | | | | | | | | |
| クラス名 | | **AnimalDAO** | | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | | 型 | | 初期値 | 備考 | |
| 6 | sql | public | | String | | "SELECT ANIMAL\_KEY, SEX,  AGE, PLACE" | 動物情報を問い合わせるSQL文 | |
| 7 | pStmt | public | | PreparedStatement | | conn.prepareStatement(sql) | SQL発行用オブジェクト | |
| 8 | rs | public | | ResultSet | | pStmt.executeQuery | SQL抽出結果 | |
| 9 | animalKey | public | | String | | rs.getString("ANIMAL\_KEY") | DBのANIMAL\_KEY（アニマルKEY） | |
| 10 | sex | public | | String | | rs.getString("SEX") | DBのSEX（パスワード） | |
| 11 | age | public | | String | | rs.getString("AGE") | DBのAGE(年齢) | |
| 12 | place | public | | Place | | rs.get("PLACE") | DBのPLACE(場所) | |
| 13 | animalInfo | public | | AnimalInfo | | new AnimalInfo(animalKey, sex, age,place) | 上記9～12の属性を持つ  Animalンスタンスの生成 | |
| メソッド | No | | 名称 | | 属性  （可視性） | | 型 | 初期値 | | 備考 | |
| 1 | |  | |  | |  |  | |  | |
| 2 | |  | |  | |  |  | |  | |
| 3 | |  | |  | |  |  | |  | |

クラス定義書DAO③

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | |
| 機能 | 「Message」テーブルを担当する | | | | | | |
| クラス名 | **MessageDAO** | | | | | | |
| パッケージ | dao | | | | | | |
| 可視性 | public | | | | | | |
| 継承クラス |  | | | | | | |
| インポート | java.sql.Connection;  java.sql.DriverManager;  java.sql.PreparedStatement;  java.sql.ResultSet;  java.sql.SQLException;  java.util.ArrayList;  java.util.List;  model.MessageInfo; | | | | | | |
| サブクラス |  | | | | | | |
| フィールド | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | JDBC\_URL | private | String | jdbc:mysql://localhost/test\_db?characterEncoding=UTF-8 | データベース（以下DB)に関する情報 |
| 2 | DB\_USER | private | String | test\_user | DBのユーザーに関する情報 |
| 3 | DB\_PASS | private | String | test\_pass | DBのパスワードに関する情報 |
| 4 | messageInfoList | public | List<MessageInfo> | newArrayList<>(); | ArrayListインスタンスの生成 |

**次ページへ続く**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | |
| 機能 | 「Message」テーブルを担当する | | | | | | |
| クラス名 | **MessageDAO** | | | | | | |
| フィールド | 5 | conn | public | Connection | DriverManager.getConnection(JDBC\_URL,DB\_USER,DB\_PASS) | DBへの接続情報 |
| 6 | sql | public | String | "SELECT ID,NAME, TEXT,MESSAGE\_TIME, ANIMAL\_KEY  FROM MESSAGE " | メッセージ情報を問い合わせSQL文 |
| 7 | rs | public | ResultSet | pStmt.executeQuery | SQL文を実行、結果を取得 |
| 8 | id | public | String | rs.getString("ID") | DBのID（ID） |
| 9 | name | public | String | rs.getString("NAME") | DBのNAME（名前） |
| 10 | text | public | String | rs.getString("TEXT”) | DBのTEXT(テキスト) |
| 11 | message\_time | public | String | rs.getString("MESSAGE\_TIME") | DBのMESSAGE\_TIME(メッセージの時刻) |
| 12 | animal\_key | public | int | rs.getIng("ANIMAL\_KEY") | DBのANIMAL\_KEY(動物のキー) |
| 13 | messageInfo | public | MessageInfo | new MessageInfo(id,name,text,message\_time,animal\_key | 上記8～12の属性を持つ  MessageInfoインスタンスの生成 |
| 14 | con | public | Connection | DriverManager.getConnection(JDBC\_URL,DB\_USER,DB\_PASS) | DBへの接続情報 |
| 15 | ps | public | PreparedStatement | conn.prepareStatement(sql) | SQL発行用オブジェクト |
| 16 | rs | public | ResultSet | pStmt.executeUpdate | SQL抽出結果 |

クラス定義書

【Model】

クラス定義書Model①

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | | |
| 機能 | ログインユーザー情報を担当する | | | | | | | |
| クラス名 | **AccountInfo** | | | | | | | |
| パッケージ | model | | | | | | | |
| 可視性 | public | | | | | | | |
| 継承クラス | implements Serializable | | | | | | | |
| インポート | java.Serializable | | | | | | | |
| サブクラス |  | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | userId | private | String |  | ユーザーID |
| 2 | pass | private | String |  | ユーザーパスワード |
| 3 | userName | private | String |  | ユーザー名 |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| メソッド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | getUserId() | public | String | void | userIdを外部クラスで取得（getter） |
| 2 | setUserId() | public | void | String userId | フィールドにユーザーIDを代入 |
| 3 | getPass() | public | String | void | passを外部クラスで取得（getter） |
| 4 | setPass() | public | void | String passWord | フィールドにパスワードを代入 |
| 5 | getUserName() | public | String | void | userNameを外部クラスで取得（getter） |
| 6 | setUserName() | public | void | String userName | フィールドにユーザー名を代入 |

クラス定義書Model②

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | | |
| 機能 | ユーザー情報の仕組み | | | | | | | |
| クラス名 | **AccountLogic** | | | | | | | |
| パッケージ | model | | | | | | | |
| 可視性 | public | | | | | | | |
| 継承クラス | executeSelectAccountInfo | | | | | | | |
| インポート | java.util.List;  dao.AccountDAO; | | | | | | | |
| サブクラス |  | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | dao | public | AccountDAO | new AccountDAO() | アカウントDAOのインスタンス生成 |
| 2 | AccountInfoList | public | List<AcountInfo> | dao.doSelect | 「AccountInfo」テーブルの全レコード取得 |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| メソッド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | execute | public | void | List<AccountInfo> | 「AccountInfo」テーブルの全レコード取得 |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

クラス定義書Model③

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | | |
| 機能 | 動物の情報を担当する | | | | | | | |
| クラス名 | **AnimalInfo** | | | | | | | |
| パッケージ | model | | | | | | | |
| 可視性 | public | | | | | | | |
| 継承クラス |  | | | | | | | |
| インポート | java.io.Serializable | | | | | | | |
| サブクラス |  | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | animalKey | private | int |  | 動物のインデックス |
| 2 | sex | private | String |  | 性別 |
| 3 | age | private | String |  | 年齢 |
| 4 | place | private | String |  | 現在地 |
| 5 | image | private | Long |  | 画像 |
| メソッド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | getAnimalKey() | public | int | void | AnimalKeyを外部クラスで取得（getter） |
| 2 | setAnimalKey() | public | void | int AnimalKey | フィールドにアニマルキーを代入 |
| 3 | getSex() | public | String | void | Sexを外部クラスで取得（getter） |
| 4 | setSex() | public | void | String Sex | フィールドに性別を代入 |
| 5 | getAge() | public | String | void | Ageを外部クラスで取得（getter） |
| 6 | setAge() | public | void | String Age | フィールドに年齢を代入 |
| 7 | getPlace() | public | String | void | Placeを外部クラスで取得（getter） |
| 8 | setPlace () | public | void | String Place | フィールドに場所を代入 |
| 9 | getImage() | public | String | void | Imageを外部クラスで取得（getter） |
| 10 | setImage() | public | void | Long Image | フィールドに写真を代入 |

クラス定義書Model④

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | | |
| 機能 | 動物の情報取得の仕組み | | | | | | | |
| クラス名 | **GetAnimalListLogic** | | | | | | | |
| パッケージ | model | | | | | | | |
| 可視性 | public | | | | | | | |
| 継承クラス |  | | | | | | | |
| インポート | java.util.List  dao.AnimalDAO | | | | | | | |
| サブクラス |  | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | dao | public | AnimalDAO | new AnimalDAO() | AnimalDAOのインスタンス生成 |
| 2 | animalList | public | List<AnimalInfo> | dao.findoAll() | 「AnimalInfo」テーブルの全レコード取得 |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| メソッド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | execute | public | void | List<AnimalInfo> | 「AnimalInfo」テーブルの全レコード取得 |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

クラス定義書Model⑤

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | | |
| 機能 | メッセージの情報のJava Beans「Message」テーブルのレコードを表す | | | | | | | |
| クラス名 | **MessageInfo** | | | | | | | |
| パッケージ | model | | | | | | | |
| 可視性 | public | | | | | | | |
| 継承クラス | Serializable | | | | | | | |
| インポート | java.io.Serializable | | | | | | | |
| サブクラス |  | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | id | private | int |  | ユーザーID |
| 2 | name | private | String |  | ユーザーの名前 |
| 3 | text | private | String |  | メッセージの内容 |
| 4 | message\_time | private | String |  | 投稿日時 |
| 5 | animal\_key | private | int |  | 動物の主キー |
| メソッド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | MessageInfro | public | void | void | コンストラクタ |
| 2 | MessageInfro | public | String name,String text,String message\_time | void | ユーザーの名前、メッセージ内容、投稿日時に値を代入するコンストラクタ |
| 3 | MessageInfo | public | int id,int animal\_Key, String name,String text,String message\_time | void | ユーザーID、動物の主キー、ユーザーの名前、メッセージ内容、投稿日時に値を代入するコンストラクタ |

**次ページへ続く**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | |
| 機能 | メッセージの情報のJava Beans「Message」テーブルのレコードを表す | | | | | | |
| クラス名 | **MessageInfo** | | | | | | |
| メソッド | 4 | getId | public | void | int | ユーザーIDの取得 |
|  | 5 | getName | public | void | String | ユーザーの名前の取得 |
|  | 6 | getText | public | void | String | メッセージ内容の名前の取得 |
|  | 7 | get Message\_time | public | void | String | 動物の主キーの取得 |
|  | 9 | get Animal\_key | public | void | int | DBのNAME（名前） |

クラス定義書Model⑥

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | | | |
| 機能 | メッセージの情報を取得する | | | | | | | |
| クラス名 | **MessageLogic** | | | | | | | |
| パッケージ | model | | | | | | | |
| 可視性 | public | | | | | | | |
| 継承クラス |  | | | | | | | |
| インポート | java.util.List  dao.MessageDAO | | | | | | | |
| サブクラス |  | | | | | | | |
| フィールド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | dao | public | MessageDAO | new MessageDAO() | MessageDAOのインスタンスの生成 |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| メソッド | | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | execute | public | Message message | void | 投稿されたメッセージ情報をDBに追加 |
| 2 | findAll | public | void | List<AMessageInfo> | DB接続、動物の主キーを指定したメッセージの取得 |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

クラス定義書

【Servlet】

クラス定義書Servlet①

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| システム名 | aniMatch | | | | | |
| 機能 | 新規登録に関するリクエストを処理するコントローラ | | | | | |
| クラス名 | AccountRegist | | | | | |
| パッケージ | servlet | | | | | |
| 可視性 | public | | | | | |
| 継承クラス | HttpServlet | | | | | |
| インポート | java.io.IOException  java.util.regex.Pattern  javax.servlet.RequestDispatcher  javax.servlet.ServletException  javax.servlet.annotation.WebServlet  javax.servlet.http.HttpServlet  javax.servlet.http.HttpServletRequest  javax.servlet.http.HttpServletResponse  javax.servlet.http.HttpSession  model.AccountInfo  model.AccountLogic | | | | | |
| サブクラス |  | | | | | |
| フィールド | No | 名称 | 属性  （可視性） | 型 | 初期値 | 備考 |
| 1 | serialVersionUID | private | long | 1L | Final宣言文 |
| 2 | getAccountLogic | public | GetAccountLogic | new GetAccountLogic() | GetAccountLogicのインスタンス生成 |
| 3 | session | public | HttpSession | request.getSssion() | セッションからアカウントデータを取得 |
| 4 | accountInfoOnSession | public | AccountInfoDto | (AccountInfoDto)session.getAttribute(“ACCOUNT\_INFO”) | セッションからアカウントデータを取得 |

**次ページへ続く**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| フィールド | 5 | dispatcher | public | RequestDispatcher | request.getRequestDispatcher (“/WEB-INF/jsp/accountConfirm.jsp”) | フォワード先（登録確認画面）の情報を取得 |
| 6 | userName | public | String | request.getParametar  (“userName”) | リクエストパラメータ |
| 7 | usreId | public | String | request.getParametar  (“userId”) | リクエストパラメータ |
| 8 | pass | public | String | request.getParametar(“pass”) | リクエストパラメータ |
| 9 | regist | public | String | request.getParametar(“regist”) | リクエストパラメータ |
| 10 | account | public | Account | new AccountLogic(accountInfoOnSession,getUserName(),userId,pass,regist | AccountInfoインスタンスの生成 |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| メソッド |  | 名称 | 属性  （可視性） | 引数リスト | 戻り値 | 備考 |
| 1 | doGet() | public | HttpServletRequest request,HttpServletResponse response | void | アカウントデータを取得して、リクエストスコープに保存。  登録を確認するため、セッションスコープからユーザー情報を取得。 |
| 2 | doPost() | public | HttpServletRequest request,HttpServletResponse response | void | 入力値チェック  アカウントリストを取得して、リクエストスコープに保存 |

クラス定義書Servlet②

|  |  |
| --- | --- |
| システム名 | aniMatch |
| 機能 | ログインに関するリクエストを処理するコントローラ |
| クラス名 | Login |
| パッケージ | servlet |
| 可視性 | public |
| 継承クラス | HttpServlet |
| インポート | java.io.IOException  javax.servlet.RequestDispatcher  javax.servlet.ServletException  javax.servlet.annotation.WebServlet  javax.servlet.http.HttpServlet  javax.servlet.http.HttpServletRequest  javax.servlet.http.HttpServletResponse  javax.servlet.http.HttpSession  model.AccountInfo  model.AccountLogic  model.MessageInfo |
| サブクラス |  |

クラス定義書Servlet③

|  |  |
| --- | --- |
| システム名 | aniMatch |
| 機能 | ログアウトへの遷移 |
| クラス名 | Logout |
| パッケージ | servlet |
| 可視性 | public |
| 継承クラス | HttpServlet |
| インポート | java.io.IOException  javax.servlet.RequestDispatcher  javax.servlet.ServletException  javax.servlet.annotation.WebServlet  javax.servlet.http.HttpServlet  javax.servlet.http.HttpServletRequest  javax.servlet.http.HttpServletResponse  javax.servlet.http.HttpSession |
| サブクラス |  |

クラス定義書Servlet④

|  |  |
| --- | --- |
| システム名 | aniMatch |
| 機能 | メインへの遷移 |
| クラス名 | Main |
| パッケージ | servlet |
| 可視性 | public |
| 継承クラス | HttpServlet |
| インポート | java.io.IOException  java.util.List  javax.servlet.RequestDispatcher  javax.servlet.ServletException  javax.servlet.annotation.WebServlet  javax.servlet.http.HttpServlet  javax.servlet.http.HttpServletRequest  javax.servlet.http.HttpServletResponse  model.AnimalInfo  model.GetAnimalListLogic |
| サブクラス |  |

クラス定義書Servlet⑤

|  |  |
| --- | --- |
| システム名 | aniMatch |
| 機能 | メッセージへの遷移 |
| クラス名 | Message |
| パッケージ | servlet |
| 可視性 | public |
| 継承クラス | HttpServlet |
| インポート | java.io.IOException  java.time.LocalDateTime  java.time.format.DateTimeFormatter  java.util.List  javax.servlet.RequestDispatcher  javax.servlet.ServletException  javax.servlet.annotation.WebServlet  javax.servlet.http.HttpServlet  javax.servlet.http.HttpServletRequest  javax.servlet.http.HttpServletResponse  javax.servlet.http.HttpSession  model.AccountInfo  model.MessageInfo  model.MessageLogic |
| サブクラス |  |