**クラス毎のフィールドとメソッド**

**1. dao.MuttersDAO**

* **フィールド:**
  + JDBC\_URL: データベース接続URL
  + DB\_USER: データベース接続ユーザー名
  + DB\_PASS: データベース接続パスワード
* **メソッド:**
  + findAll(): 全ての有効なつぶやきを新しい順にリストで取得します。
  + insert(Mutter mutter): 画像付きのつぶやきをデータベースに登録します。
  + create(Mutter mutter): 画像なしのつぶやきをデータベースに登録します（insertと一部重複あり）。
  + deleteMutterById(int id): 指定されたIDのつぶやきの削除フラグを立てます。
  + updateMutterText(int id, String newText): 指定されたIDのつぶやきのテキスト内容を更新します。
  + findById(int id): 指定されたIDのつぶやき情報を取得します。
  + getImageById(int id): 指定されたIDのつぶやきの画像データを取得します。

**2. model.Mutter**

* **フィールド:**
  + id: つぶやきのID
  + userName: 投稿ユーザー名
  + text: つぶやき内容
  + createdAt: 投稿日時
  + image: 添付画像データ
* **メソッド:**
  + 各フィールドのgetterメソッドとsetterメソッド
  + 複数のコンストラクタ

**3. model.DeleteMutterLogic**

* **フィールド:** なし
* **メソッド:**
  + execute(int id): 指定されたIDのつぶやきを削除（フラグ更新）する処理をMuttersDAOに依頼します。

**4. model.GetMutterListLogic**

* **フィールド:** なし
* **メソッド:**
  + execute(): 全てのつぶやきリストを取得する処理をMuttersDAOに依頼し、その結果を返します。

**5. model.LoginLogic**

* **フィールド:** なし
* **メソッド:**
  + execute(User user): 受け取ったユーザー情報が固定のパスワード("1234")と一致するかを判定します。

**6. model.User**

* **フィールド:**
  + name: ユーザー名
  + pass: パスワード
* **メソッド:**
  + 各フィールドのgetterメソッド

**7. model.PostMutterLogic**

* **フィールド:** なし
* **メソッド:**
  + execute(Mutter mutter): つぶやき情報をMuttersDAOに登録する処理を依頼します。

**8. servlet.EditFormServlet**

* **フィールド:** serialVersionUID
* **メソッド:**
  + doGet(HttpServletRequest request, HttpServletResponse response): 編集対象のつぶやきIDを取得し、その情報をリクエストスコープに格納して編集画面へフォワードします。
  + doPost(HttpServletRequest request, HttpServletResponse response): doGetを呼び出します。

**9. servlet.DeleteMutterServlet**

* **フィールド:** serialVersionUID
* **メソッド:**
  + doGet(HttpServletRequest request, HttpServletResponse response): 削除対象のつぶやきIDを取得し、削除処理のロジックを実行後、メイン画面へリダイレクトします。
  + doPost(HttpServletRequest request, HttpServletResponse response): doGetを呼び出します。

**10. servlet.EditMutterServlet**

* **フィールド:** serialVersionUID
* **メソッド:**
  + doPost(HttpServletRequest request, HttpServletResponse response): 編集されたつぶやきIDと内容を受け取り、更新処理を実行後、メイン画面へリダイレクトします。
  + doGet(HttpServletRequest request, HttpServletResponse response): doPostを呼び出します。

**11. servlet.ImageServlet**

* **フィールド:** serialVersionUID
* **メソッド:**
  + doGet(HttpServletRequest request, HttpServletResponse response): リクエストされたつぶやきIDの画像データを取得し、レスポンスとして出力します。
  + doPost(HttpServletRequest request, HttpServletResponse response): doGetを呼び出します。

**12. servlet.Main**

* **フィールド:** serialVersionUID
* **メソッド:**
  + doGet(HttpServletRequest request, HttpServletResponse response): つぶやきリストを取得してリクエストスコープに格納し、ログイン状態を確認してメイン画面へフォワードまたはログイン画面へリダイレクトします。
  + doPost(HttpServletRequest request, HttpServletResponse response): 受け取ったつぶやき内容と画像をPostMutterLogicに渡し、投稿処理を実行後、メイン画面へフォワードします。

**13. servlet.Logout**

* **フィールド:** serialVersionUID
* **メソッド:**
  + doGet(HttpServletRequest request, HttpServletResponse response): セッションを無効化し、ログアウト画面へフォワードします。

**14. servlet.Login**

* **フィールド:** serialVersionUID
* **メソッド:**
  + doPost(HttpServletRequest request, HttpServletResponse response): ユーザー名とパスワードを受け取り、LoginLogicで認証処理を行い、成功すればユーザー情報をセッションスコープに保存してログイン結果画面へフォワードします。