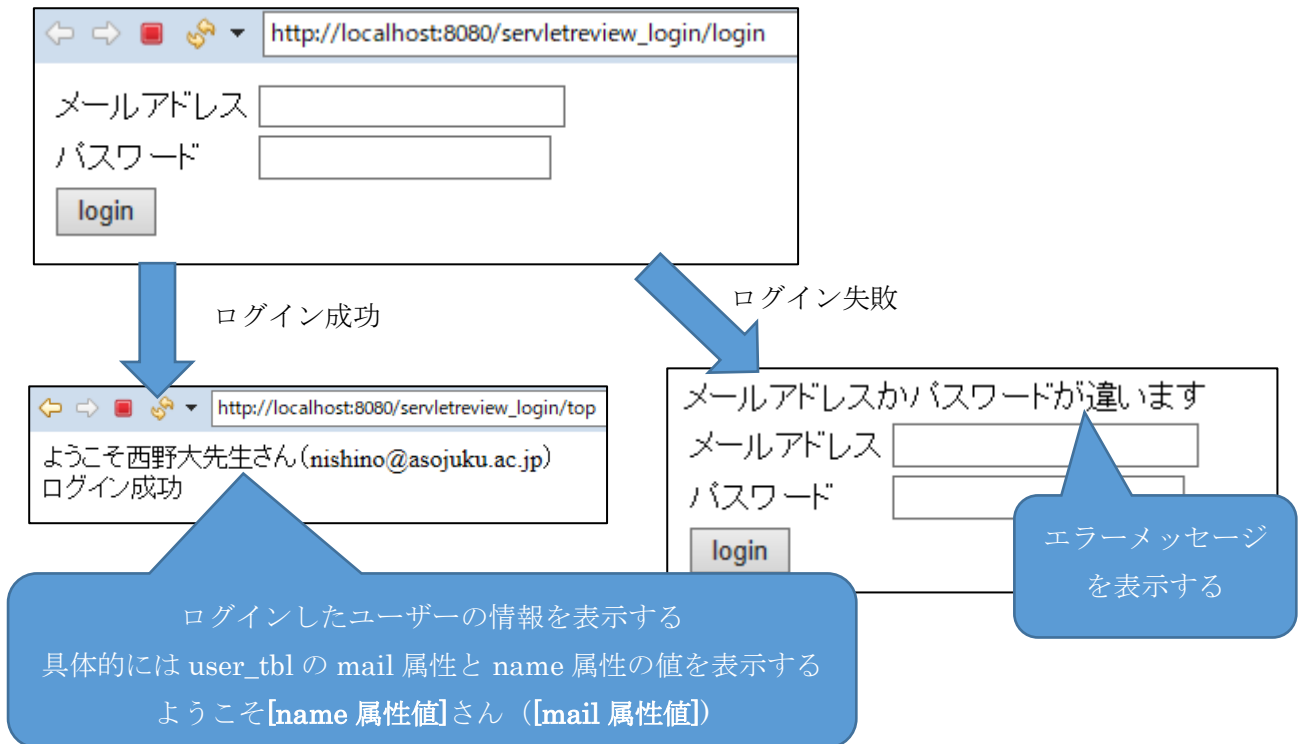


タイトル	
DB を使ったログイン	
レベル	★★★★

以下のように DB を使ったログイン処理を作ってみよう



ワークスペース名	c:¥work¥review
プロジェクト名	servletreview_login

#### ★DB 情報★

DB 名	webtestdb
テーブル名	user_tbl

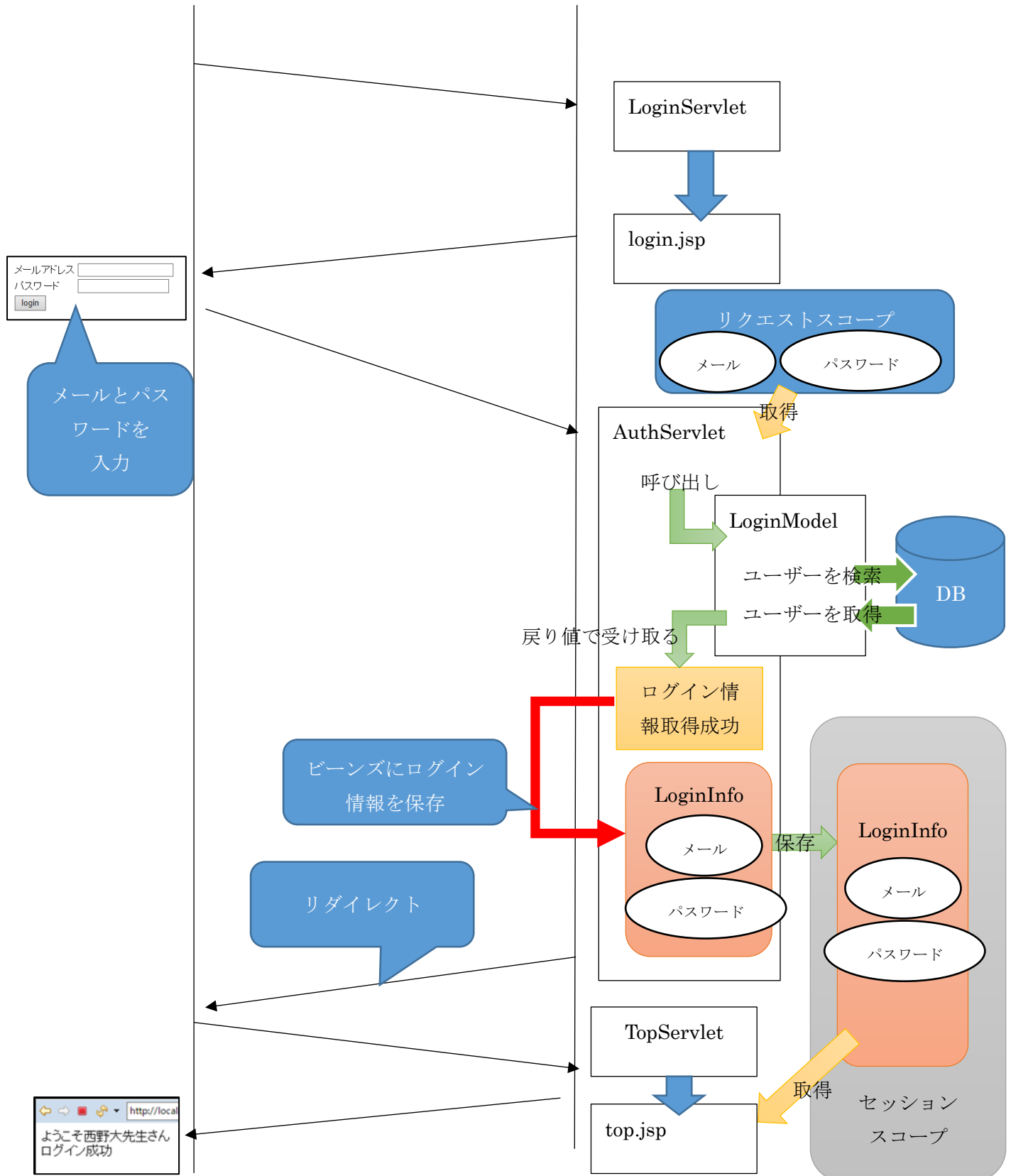
#### ★user\_tbl の属性★

mail	文字列 (主キー)	メールアドレス
password	文字列	パスワード
name	文字列	名前

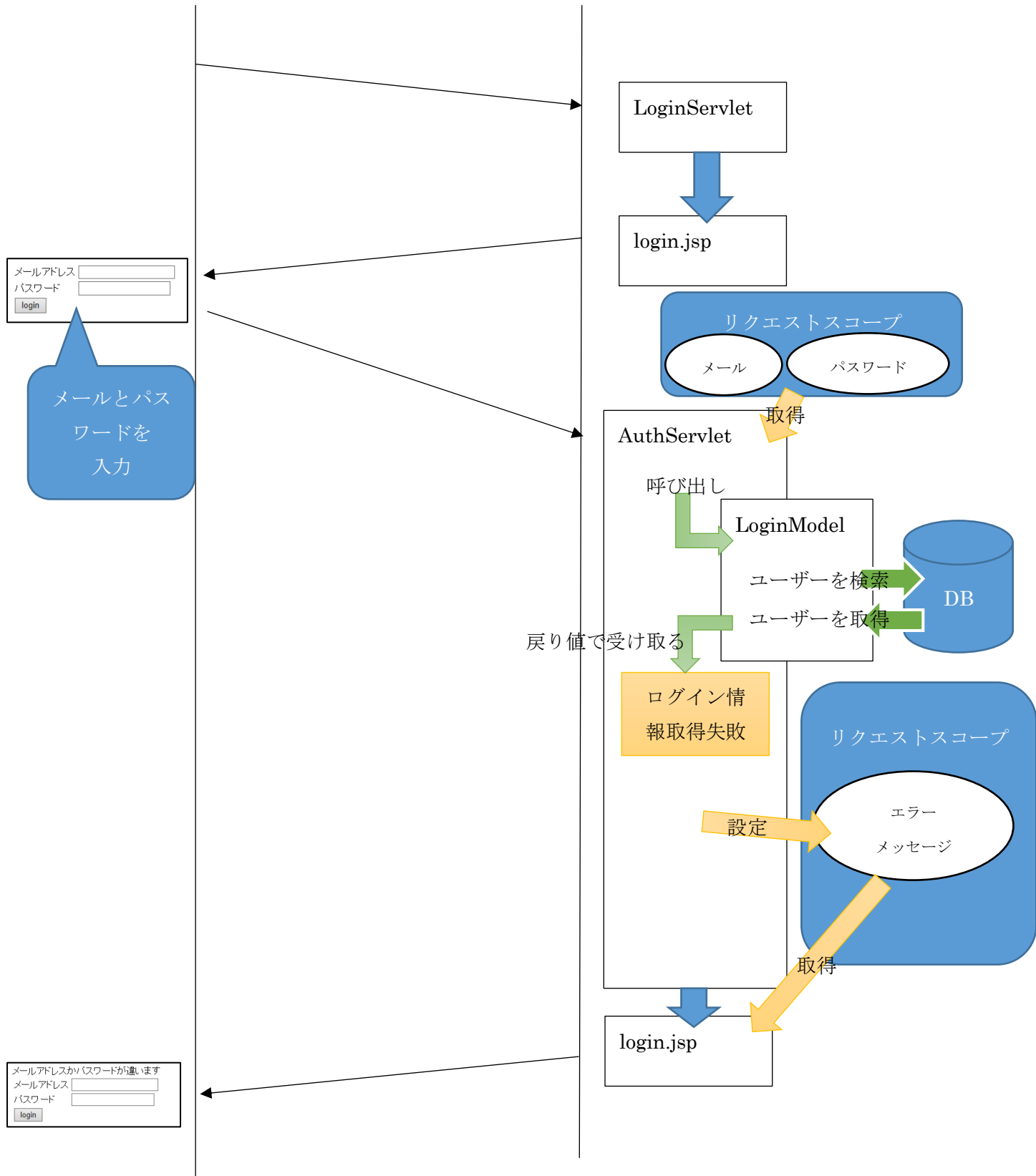
※前期に作成済みのはず

## 処理のフロー

### (1) ログイン成功時

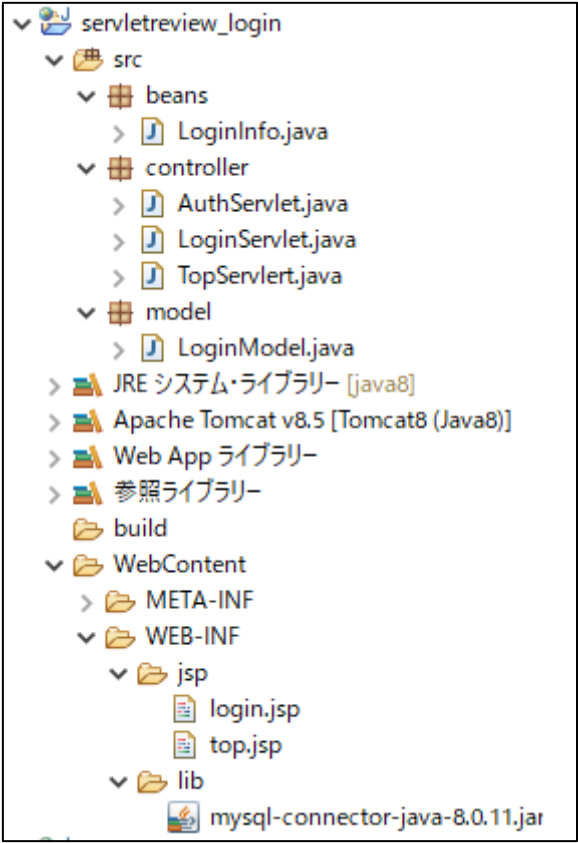


(1) ログイン成功時



ファイル構成

以下のようなファイル構成で作成しましょう



ファイル名	概要	MVC
LoginInfo	ログイン情報を保持するビーンズ	-
AuthServlet	ログイン認証を行うサーブレット	C
LoginServlet	ログイン画面を表示するサーブレット	C
TopServlet	トップページを表示するサーブレット	C
LoginModel	ログイン処理の中核部分を行うモデル（DBに接続して検索する）	M
login.jsp	ログイン画面	V
top.jsp	トップ画面	V
mysql-connector-java	JDBC ドライバ	

早く終わった人は↓もチャレンジ！

内容	難易度
CSS などを使ってログイン画面をカッコよくする	★★★★
エラーメッセージを赤色で表示する	★★

どうしてもわからない人は次ページからのヒントを見よう！

## スペシャルヒント

### login.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2 pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8"%>
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <%
11 String errMsg = (String)
12 %>
13 <% if( errMsg != null ){ %>
14 <%= errMsg %>
15 <% } %>
16 <form action="auth" method="GET">
17 <table>
18 <tr>
19 <td>メールアドレス</td>
20 <td><input type="text" name="mail"></td>
21 </tr>
22 <tr>
23 <td>パスワード</td>
24 <td><input type="password" name="password"></td>
25 </tr>
26 <tr>
27 <td colspan="2"><input type="submit" value="login"></td>
28 </tr>
29 </table>
30 </form>
31 </body>
32 </html>
```

### AuthServlet.java

```
@WebServlet("/auth")
public class AuthServlet extends HttpServlet {

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        //////////////////////////////////////
        //ログイン画面（JSP）で入力された値の取得
        String mail =
        String password =

        //////////////////////////////////////
        //モデルのインスタンスを作成し、ログインメソッドを実行
        LoginModel loginModel =
        LoginInfo loginInfo =

        //////////////////////////////////////
        //ログインメソッドの結果、ログイン情報が取得できかどうかで画面遷移先を変える
        if( loginInfo != null ) {
            //ログイン情報をセッションに保存
            HttpSession session =
            //TOP画面へリダイレクト
            response.sendRedirect("top");
        } else {
            //エラーメッセージをセットして、login.jspへフォワード
            request.setAttribute("errMsg", "メールアドレスかパスワードが違います");
            RequestDispatcher dispatcher =
            dispatcher.forward(request, response);
        }
    }
}
```

## LoginModel.java

```
public class LoginModel {  
    /**  
     * ログイン処理を実行する  
     * @param mail  
     * @param password  
     * @return  
     */  
    public LoginInfo login(String mail,String password) {  
        LoginInfo loginInfo = null;  
  
        Connection con = null;  
        PreparedStatement stmt = null;  
        ResultSet rs = null;  
  
        try{  
            ///////////////////////////////////  
            //DBの接続  
            con = DriverManager.getConnection(  
                "jdbc:mysql://localhost:3306/webtestdb?characterEncoding=UTF-8&serverTimezone=JST",  
                "root","");  
  
            ///////////////////////////////////  
            //SELECT文の発行  
            stmt =   
            stmt.  
            stmt.  
            rs =   
  
            ///////////////////////////////////  
            //DBから値を取得  
            while( rs.next() ){  
                loginInfo = new LoginInfo();  
  
                loginInfo.setMail(  
                loginInfo.setName(  
            }  
  
        }catch(ClassNotFoundException e) {  
            //エラー発生した場合にコンソールにログを出力する  
            e.printStackTrace();  
        }catch(SQLException e) {  
            //エラー発生した場合にコンソールにログを出力する  
            e.printStackTrace();  
        }  
        finally {  
            //接続（コネクション）を閉じる  
            if(con!=null) {  
                try {  
                    con.close();  
                }  
                catch(SQLException e) {  
                    e.printStackTrace();  
                }  
            }  
        }  
  
        return loginInfo;  
    }  
}
```

使用可i

17/16