**社員情報システム　内部設計仕様書**

作成日 ： 2017年10月26日

作成者 ： 向山、香川、ミッチェル

1. **システム機能詳細**

**パッケージ構成**

■modelパッケージ

src/model 共通部品 Modelパッケージ Do I need to seperate this?

・Common.java 共通部品のクラス

src/model 社員情報システム用Modelパッケージ

・EmployeeBean.java JavaBeansクラス

・SystemLogic.java Logicクラス

■daoパッケージ

src/dao 社員情報システム用DAOパッケージ

・EmployeeSystemDAO.java データベース操作クラス

■servletパッケージ

src/servlet 社員情報システム用servletパッケージ

・EmployeeSystem.java Controllerクラス

■JSPファイル

WebContent/WEB-INF/jsp 社員情報システム用Viewモジュール

・login.jsp 社員情報ログイン画面

・employeeSystemMain.jsp 社員情報メイン画面

・employeeViewer.jsp 社員情報詳細画面

・registerUser.jsp 社員情報登録画面

・confirmRegisterUser.jsp 社員情報登録内容確認画面

・updateUser.jsp 社員情報修正画面

・confirmUpdateUser.jsp 社員情報修正内容確認画面

・confirmDeleteUser.jsp 社員情報削除内容確認画面

**クラス・モジュール構成**

■controller(servlet)

【 EmployeeSystem.java 】

・doGetメソッド

引数：HttpServletRequest request, HttpServletResponse response

戻り値：void

処理詳細：

・URL、ハイパーリンクでの呼び出しに対応

・パラメータは無しの場合

・employeeSystemMain.jspへフォワード転送

・actionのパラメータによって異なる処理を行う

action=registerの場合

・セッションスコープにある「EmployeeBean」のデータをデータベースに入れる

・セッションスコープのアトリビュートを削除：

・「EmployeeBean」

・「selectedUser」

action=updateUserの場合

・セッションスコープにある「EmployeeBean」のデータを使って、データベースの中にある同じ「employeeId」のデータを更新する

・セッションスコープのアトリビュートを削除：

・「EmployeeBean」

・「selectedUser」

action=deleteUserの場合

・セッションスコープにある「EmployeeBean」のデータをデータベースに入れる

・セッションスコープのアトリビュートを削除：

・「EmployeeBean」

・「selectedUser」

上記の各プロセスが終了後

・employeeSystemMain.jspへフォワード転送

※問題があるなら、エラーメッセジーをリクエストスコープにいれる

action=””の場合

pageのパラメータによって異なる処理を行う

page=registerUserの場合

・registerUser.jspへフォワード転送

page=viewUserの場合

・「selectedUser」のパラメータはある場合

・「selecteUser」の値を使って、データベースからデータを引く

・「EmployeeBean」を作る

・セッションスコープのアトリビュートを作る：

「EmployeeBean」=新EmployeeBean

「selectedUser」=EmployeeBeanのemployeeId ・employeeViewer.jspへフォワード転送

page=updateUserの場合

・「lastpage」のパラメータはある場合

・セッションスコープのアトリビュートを作る：

「lastpage」=「lastpage」のパラメータの値

・「lastpage」のパラメータは空文字“”の場合

・セッションスコープのアトリビュートを作る：

「lastpage」=employeeSystemMain.jsp

・「selectedUser」のパラメータはある場合

・「selecteUser」の値を使って、データベースからデータを引く

・「EmployeeBean」を作る

・セッションスコープのアトリビュートを作る：

「EmployeeBean」=新EmployeeBean

「selectedUser」=EmployeeBeanのemployeeId

・updateUser.jspへフォワード転送

page=deleteUserの場合

・「lastpage」のパラメータはある場合

・セッションスコープのアトリビュートを作る：

「lastpage」=「lastpage」のパラメータの値

・「lastpage」のパラメータは空文字“”の場合

・セッションスコープのアトリビュートを作る：

「lastpage」=employeeSystemMain.jsp

・「selectedUser」のパラメータはある場合

・「selecteUser」の値を使って、データベースからデータを引く

・「EmployeeBean」を作る

・セッションスコープのアトリビュートを作る：

「EmployeeBean」=新EmployeeBean

「selectedUser」=EmployeeBeanのemployeeId

・confirmDeleteUser.jspへフォワード転送

・doPostメソッド

引数：HttpServletRequest request, HttpServletResponse response

戻り値：void

処理詳細：

・ログインフォームのPOSTでの呼び出しに対応

下記のパラメータを受け取る

String employeeId, password

・検索フォームのPOSTでの呼び出しに対応

下記のパラメータを受け取る

String baseName, departmentName, divisionName, positionName, fromDate, toDate

・新規登録フォームのPOSTでの呼び出しに対応

下記のパラメータを受け取る

String employeeId, password, employeeName, kana, gender, departmentName, divisionName, positionName, officePositionMemo, naisenNumber, publicCellphoneNumber

・修正フォームのPOSTでの呼び出しに対応

下記のパラメータを受け取る

String employeeId, oldPassword, newPassword, employeeName, kana, departmentName, divisionName, positionName, officePositionMemo, naisenNumber, publicCellphoneNumber, adminFlag

・actionのパラメータによって異なる処理を行う

・action=loginの場合

・ログインフォームのパラメータを受け取る

・入力したパラメータとデータベースの一致を探す

※正しく入力してない場合、エラーメッセジ―をリクエストスコープに入れて、login.jspへフォワード転送

・ログインした社員の社員ＩＤと権限にセッションスコープに入れる

※「hiddenのselectedUser」ペラメータが空文字“”の場合：

・employeeSystemMain.jspへフォワード転送

※「hiddenのselectedUser」ペラメータが空文字“”じゃないの場合：

・selectedUserの値をセッションスコープに入れる

・employeeViewer.jspへフォワード転送

・action=searchの場合

・検索フォームのパラメータを受け取る

・受け取ったパラメータをリクエストスコープに入れて、employeeSystemMain.jspへフォワード転送

・action=confirmNewUserの場合

・新規登録フォームのパラメータを受け取る

・「EmployeeBean」のインスタンスを作って、受け取ったパラメータをインスタンスに入れて、内容を確認する

※正しく入力する場合：

・confirmRegisterUser.jspへフォワード転送

※正しく入力しない場合：

・エラーメッセジ―をリクエストスコープに入れる

・registerUser.jspへフォワード転送

・action=confirmUpdateUserの場合

・修正フォームのパラメータを受け取る

・セッションスコープにある「EmployeeBean」の内容を変更して、確認する

※正しく入力する場合：

・confirmUpdateUser.jspへフォワード転送

※正しく入力しない場合：

・エラーメッセジ―をリクエストスコープに入れる

・updateUser.jspへフォワード転送

■view(JSP)

【 login.jsp 】

・ログイン入力フォーム

<form action="/…/EmployeeSystem" method="post">

|  |  |  |
| --- | --- | --- |
| パラメータ名 | Type | 備考 |
| employeeID | text | 社員ＩＤ |
| password | password | 社員パスワード |
| action | hidden | value=”login” |
| action | hidden | value=”viewEmployee” |

【 employeeSystemMain.jsp 】

・検索入力フォーム

<form action="/…/EmployeeSystem" method="post">

|  |  |  |
| --- | --- | --- |
| パラメータ名 | Type | 備考 |
| baseName | select box | 拠点名 |
| departmentName | select box | 部署名 |
| divisionName | select box | 課名 |
| positionName | select box | 役職名 |
| fromDate | select box | 入社年月 |
| toDate | select box | 入社年月 |
| action | hidden | value="search" |

【registerUser.jsp 】

・新規登録入力フォーム

<form action="/…/EmployeeSystem" method="post">

|  |  |  |
| --- | --- | --- |
| パラメータ名 | Type | 備考 |
| employeeId | number | 社員ＩＤ |
| ｐassword | text | 社員パスワード |
| employeeName | text | 名前 |
| ｋana | text | ふりがな |
| ｇender | select | 性別 |
| departmentName | select | 課名 |
| divisionName | select | 課名 |
| positionName | select | 役職名 |
| officePositionMemo | text | 役職詳細 |
| naisenNumber | text | 内線番号 |
| publicCellphoneNumber | text | 業務携帯番号 |
| action | hidden | value="confirmNewUser" |

【registerUser.jsp 】

・管理者パスワード入力フォーム

<form action="/…/EmployeeSystem" method="post">

|  |  |  |
| --- | --- | --- |
| パラメータ名 | Type | 備考 |
| employeeId | number | 社員ＩＤ |
| oldPassword | text | 旧社員パスワード |
| newPassword | text | 新社員パスワード |
| employeeName | text | 名前 |
| ｋana | text | ふりがな |
| departmentName | select | 課名 |
| divisionName | select | 課名 |
| positionName | select | 役職名 |
| officePositionMemo | select | 役職詳細 |
| naisenNumber | text | 内線番号 |
| publicCellphoneNumber | text | 業務携帯番号 |
| adminFlag | select | 管理者フラグ |
| action | hidden | value="confirmUpdateUser" |

1. **データベース詳細設計**

データベース名：boarddb

ユーザー名：root

パスワード：root00

テーブル名：board\_db

|  |  |  |  |
| --- | --- | --- | --- |
| フィールド名 | データ型 | 制約 | 備考 |
| id | INT | NOT　NULL 、AUTO\_INCREMENT 、  PRIMARY KEY | 投稿番号 |
| name | VARCHAR(12) |  | 名前 |
| email | VARCHAR(20) |  | e-mail |
| comment | VARCHAR(100) | NOT NULL | コメント |
| dateTime | DATETIME |  | 投稿日時  yyyy-MM-dd hh:mm:ss |

1. **コーディング規約**

命名規則

・Javaの命名規則に準ずる

コメント

・引数と戻り値の仕様を明記する

・クラス名の上に担当者と制作開始日時を記入

・修正が発生した場合、メソッド単位(HTML・JSPは修正部分)で担当と日時を記載

・

その他

・