

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

アプリケーション名(プロジェクト名)

employeeAdmin

パッケージ構成

■model パッケージ

src/model

•Common.java

共通部品 Model パッケージ

共通部品のクラス

src/employee/model

•EmployeeBean.java

•EmployeeSystemLogic.java

社員情報システム用 Model パッケージ

JavaBeans クラス

Logic クラス

■dao パッケージ

src/dao/employee

•EmployeeSystemDAO.java

社員情報システム用 DAO パッケージ

データベース操作クラス

■servlet パッケージ

src/employee/servlet

•EmployeeSystem.java

社員情報システム用 servlet パッケージ

Controller クラス

■JSP ファイル

WebContent/WEB-INF/employee

•login.jsp

•employeeSystemMain.jsp

•employeeViewer.jsp

•registerUser.jsp

•confirmRegisterUser.jsp

•updateUser.jsp

•confirmUpdateUser.jsp

•confirmDeleteUser.jsp

社員情報システム用 View モジュール

社員情報ログイン画面

社員情報メイン画面

社員情報詳細画面

社員情報登録画面

社員情報登録内容確認画面

社員情報修正画面

社員情報修正内容確認画面

社員情報削除内容確認画面

■image ファイル

WebContent/img

社員情報システム用社員画像ファイル

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 (action=null かつ page=null)
 - ・ログイン済みの場合 (セッションスコープにログイン情報あり)
 - ・社員一覧取得処理
 - ・「EmployeeBean」のインスタンスを生成する
 - ・「EmployeeBean」の検索条件に初期値を入れる
 - ・「EmployeeBean」のオブジェクトをリクエストスコープの”employeeBean”に入れる
 - ・「EmployeeBean」を EmployeeSystemLogic の (search メソッド) に入れて、呼び出す
 - ・EmployeeSystemLogic の (search メソッド) のHTMLをもらう
 - HTML は空文字””の場合
 - ・エラーメッセージをリクエストスコープに入れる
 - HTML は空文字””じゃない場合
 - ・このHTMLをリクエストスコープの”html”にいれる
 - ・employeeSystemMain.jsp へフォワード転送
 - ・未ログインの場合 (セッションスコープにログイン情報なし)
 - ・パラメータ「user」を取得
 - ・リクエストパラメータに「user」をセット
 - ・login.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」を EmployeeSystemLogic の (register メソッド) に入れて、呼び出す
 - ・EmployeeSystemLogic の (register メソッド) のメッセージをもらう
 - ・このメッセージをリクエストスコープの”message”にいれる
 - ・セッションスコープのアトリビュートを削除:
 - ・「EmployeeBean」

action=update の場合

- セッションスコープにある「EmployeeBean」を EmployeeSystemLogic の (update メソッド)に入れて、呼び出す
- EmployeeSystemLogic の (update メソッド) のメッセージをもらう
- このメッセージをリクエストスコープの”message”にいれる
- セッションスコープの属性を削除:
 - 「EmployeeBean」
 - 「selectedUser」

action=delete の場合

- セッションスコープにある「selectedUser」を EmployeeSystemLogic の (delete メソッド)に入れて、呼び出す
- EmployeeSystemLogic の (delete メソッド) のメッセージをもらう
- このメッセージをリクエストスコープの”message”にいれる
- セッションスコープの属性を削除:
 - 「EmployeeBean」
 - 「selectedUser」

上記の各プロセスが終了後

- employeeSystemMain.jsp へフォワード転送

action=null 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
 - 「EmployeeBean」のインスタンスを宣言する
 - 「selectedUser」の値を「EmployeeBean」に入れる
 - 「EmployeeBean」をEmployeeSystemLogic の (viewUser メソッド)に入れて、呼び出す
 - EmployeeSystemLogic の (viewUser メソッド) のHTMLをもらう
- HTML は空文字“”の場合
 - エラーメッセージをリクエストスコープに入れる
 - employeeSystemMain.jsp へフォワード転送
- HTMLは空文字“”じゃない場合
 - 「EmployeeBean」をセッションスコープの” EmployeeBean”にセットする。
 - このHTMLをリクエストスコープの”html”にいれる
 - employeeViewer.jsp へフォワード転送

page=updateUser の場合

- 「lastpage」のパラメータは null の場合
- 「lastpage」のパラメータはある場合
 - 「lastpage」の値をセッションスコープの”lastpage”に入れる
- 「lastpage」のパラメータは空文字“”の場合
- 共通処理
 - 「EmployeeBean」のインスタンスを宣言する
 - 「selecteduser」の値を「EmployeeBean」に入れる
 - 「EmployeeBean」「lastpage」を EmployeeSystemLogic の (updateUser メソッド)に入れて、呼び出す
 - EmployeeSystemLogic の (updateUser メソッド)のHTMLをもらう
 - HTML は空文字“”の場合
 - エラーメッセージをリクエストスコープに入れる
 - employeeSystemMain.jsp へフォワード転送
 - HTMLは空文字“”じゃない場合
 - このHTMLをリクエストスコープの”html”にいれる
 - 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
 - 「lastpage」の値をセッションスコープの”lastpage”に入れる
- 「lastpage」のパラメータは空文字“”の場合
- 共通処理
 - 「EmployeeBean」のインスタンスを宣言する
 - 「selecteduser」の値を「EmployeeBean」に入れる
 - 「EmployeeBean」を EmployeeSystemLogic の (confirmDeleteUser メソッド)に入れて、呼び出す
 - EmployeeSystemLogic の (confirmDeleteUser メソッド)のHTMLをもらう

HTML は空文字“”の場合

- エラーメッセージをリクエストスコープに入れる
- employeeSystemMain.jsp へフォワード転送

HTMLは空文字“”じゃない場合

- このHTMLをリクエストスコープの”html”にいれる
- confirmDeleteUser.jsp へフォワード転送

page=login の場合

- セッションスコープからログイン情報を削除
「login_employeeId」「login_employeeName」「login_adminFlag」
- login.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, action, user, page

- 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, action

- 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, positionMemo,
naisenNumber, publicCellphoneNumber, action

- 修正フォームの POST での呼び出しに対応

管理用フォームから下記のパラメータを受け取る

String oldPassword, newPassword, employeeName, kana,
departmentName, divisionName, positionName, positionMemo,
naisenNumber, publicCellphoneNumber, adminFlag, action

一般用フォームから下記のパラメータを受け取る

String oldPassword, newPassword, action

•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
- 「EmployeeBean」のインスタンスを宣言する
- パラメータ「employeeId」、「password」の値を「EmployeeBean」に入れる
- 「EmployeeBean」を EmployeeSystemLogic の (login メソッド)に入れて、呼び出す
- 戻り値を「EmployeeBean」にセット
- 戻り値=null の場合

※正しく入力していない場合、エラーメッセー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、名前と権限をセッションスコープに入れる
「login_employeeId」、「login_employeeName」、「login_adminFlag」

※「hidden の user」パラメータが空文字“”の場合:

- 「EmployeeBean」をインスタンス生成。
- 「EmployeeBean」の検索条件に初期値を入れる
- 「EmployeeBean」のオブジェクトをリクエストスコープの”employeeBean”に入れる
- 「EmployeeBean」を EmployeeSystemLogic の (search メソッド)に入れて、呼び出す

- EmployeeSystemLogic の (search メソッド)のHTMLをもらう
HTML は空文字“”の場合

- エラーメッセージをリクエストスコープに入れる

HTML は空文字“”じゃない場合

- このHTMLをリクエストスコープの”html”にいれる

- employeeSystemMain.jsp へフォワード転送

※「hidden の user」パラメータが空文字“”じゃないの場合:

- 「EmployeeBean」のインスタンスを宣言する
- 「user」の値を「EmployeeBean」に入れる
- 「EmployeeBean」を EmployeeSystemLogic の (viewUser メソッド)に入れて、呼び出す

EmployeeSystemLogic の (viewUser メソッド)のHTMLをもらう
HTML は空文字“”の場合

- エラーメッセージをリクエストスコープに入れる
- employeeSystemMain.jsp へフォワード転送

HTMLは空文字“”じゃない場合

- このHTMLをリクエストスコープの”html”にいれる
- 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
- 「EmployeeBean」に検索フォームのパラメータを入れる
- 「EmployeeBean」のオブジェクトをリクエストスコープの”employeeBean”に入れる
- 「EmployeeBean」を EmployeeSystemLogic の (search メソッド)に入れて、呼び出す
- EmployeeSystemLogic の (search メソッド) のHTMLをもらう

HTML は空文字“”の場合

- エラーメッセージをリクエストスコープに入れる

HTML は空文字“”じゃない場合

- このHTMLをリクエストスコープの”html”にいれる
- 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る。
- 「EmployeeBean」を EmployeeSystemLogic の (confirmNewUser メソッド)に入れて、呼び出す
- EmployeeSystemLogic の (confirmNewUser メソッド) のHTMLをもらう

HTML は空文字“”の場合 (エラーあり)

- エラーメッセー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

HTML は空文字“”じゃない場合 (エラーなし)

- このHTMLをリクエストスコープの”html”にいれる
- confirmRegisterUser.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
- 「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る
- 「EmployeeBean」を EmployeeSystemLogic の (confirmUpdateUser メソッド)に入れて、呼び出す

- EmployeeSystemLogic の(confirmUpdateUser **メソッド**)のHTMLをもらう
HTML は空文字“”の場合 (エラーあり)
 - エラーメッセー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

- HTML は空文字“”じゃない場合 (エラーなし)
- このHTMLをリクエストスコープの”html”にいれる
 - confirmUpdateUser.jsp へフォワード転送

■model(Bean)

【 EmployeeBean.java 】

フィールド仕様

| 名前 | データ型 | 備考 |
|-----------------------|--------|-----------|
| employeeId | Int | 社員 ID |
| password | String | パスワード |
| employeeName | String | 名前 |
| kana | String | ふりがな |
| gender | String | 性別 |
| baseCode | String | 拠点コード |
| baseName | String | 拠点名 |
| departementCode | String | 部署コード |
| departmentName | String | 部署名 |
| divisionCode | String | 課コード |
| divisionName | String | 課名 |
| positionCode | String | 役職コード |
| positionName | String | 役職名 |
| positionMemo | String | 役職詳細 |
| naisenNumber | String | 内線番号 |
| publicCellphoneNumber | String | 業務携帯番号 |
| adminFlag | Byte | 管理者フラグ |
| fromDate | Int | 入社年月 (開始) |
| toDate | Int | 入社年月 (終了) |

※すべてのフィールドに getter/setter を用意する。

■model(Logic)

【 EmployeeSystemLogic.java 】

•register メソッド

引数:EmployeeBean EmployeeBean

戻り値:ArrayList<String> message

処理詳細:

- 同一社員 ID の存在チェック
 - 引数 EmployeeBean を元に取得 SQL 文の編集
 - EmployeeDaoLogic の (findEmployee メソッド) の引数にセットして呼び出す。
戻り値 ArrayList<EmployeeBean>の件数が≠0 の場合
対応したエラーメッセージを「message」に入れて返す。
- 引数 EmployeeBean を元に登録 SQL 文の編集
- EmployeeDaoLogic の (updateEmployee メソッド) の引数にセットして呼び出す。
戻り値 >0 の場合
対応した完了メッセージを「message」に入れて返す。
戻り値 ≤0 の場合
対応したエラーメッセージを「message」に入れて返す。

•update メソッド

引数:EmployeeBean EmployeeBean

戻り値:ArrayList<String> message

処理詳細:

- 引数 EmployeeBean を元に更新 SQL 文の編集
- EmployeeDaoLogic の (updateEmployee メソッド) の引数にセットして呼び出す。
戻り値 >0 の場合
対応した完了メッセージを「message」に入れて返す。
戻り値 ≤0 の場合
対応したエラーメッセージを「message」に入れて返す。

•delete メソッド

引数:EmployeeBean EmployeeBean

戻り値:ArrayList<String> message

処理詳細:

- 引数 EmployeeBean を元に更新 SQL 文の編集
- EmployeeDaoLogic の (updateEmployee メソッド) の引数にセットして呼び出す。
戻り値 >0 の場合

対応した完了メッセージを「message」に入れて返す。
戻り値 ≤ 0 の場合
対応したエラーメッセージを「message」に入れて返す。

•viewUser メソッド

引数:EmployeeBean EmployeeBean
戻り値:ArrayList<String> message

処理詳細:

- 引数 EmployeeBean を元に取得 SQL 文の編集
 - EmployeeDaoLogic の (findEmployee メソッド) の引数にセットして呼び出す。
戻り値 ArrayList<EmployeeBean>の件数が1件の場合
戻り値の 1 要素目:HTML
戻り値 ArrayList<EmployeeBean>が null または件数が 0 件の場合
戻り値の 1 要素目:空文字("")
戻り値の 2 要素目:対応したエラーメッセージを「message」に入れて返す。

•updateUser メソッド

引数:EmployeeBean EmployeeBean, String lastpage , byte adminFlag
戻り値:ArrayList<String> message

処理詳細:

- 引数 lastpage=null の場合
戻り値の 1 要素目:HTML(引数 EmployeeBean を元に作成)
- 引数 lastpage≠null の場合
引数 EmployeeBean を元に取得 SQL 文の編集
EmployeeDaoLogic の (findEmployee メソッド) の引数にセットして呼び出す。
戻り値 >0 の場合
戻り値の 1 要素目:HTML
戻り値 ≤ 0 の場合
戻り値の 1 要素目:空文字("")
戻り値の 2 要素目:対応したエラーメッセージを「message」に入れて返す。

•search メソッド

引数:EmployeeBean EmployeeBean, byte adminFlag
戻り値:ArrayList<String> message

処理詳細:

引数 EmployeeBean を元に取得 SQL 文の編集

EmployeeDaoLogic の (findEmployee メソッド) の引数にセットして呼び出す。

戻り値 ArrayList<EmployeeBean>の件数が 0 件以上の場合

戻り値の 1 要素目:HTML

戻り値 ArrayList<EmployeeBean>が null の場合

戻り値の 1 要素目:空文字("")

戻り値の 2 要素目:対応したエラーメッセージを「message」に入れて返す。

•confirmNewUser メソッド

引数:EmployeeBean EmployeeBean

戻り値:ArrayList<String> message

処理詳細:

- 入力内容のエラーチェック

- 入力内容にエラーありの場合

戻り値の 1 要素目:空文字("")

戻り値の 2 要素目:対応したエラーメッセージを「message」に入れて返す。

- 入力内容にエラーなしの場合

- 同一社員 ID の存在チェック

- 引数 EmployeeBean を元に取得 SQL 文の編集

- EmployeeDaoLogic の (findEmployee メソッド) の引数にセットして呼び出す。

戻り値 ArrayList<EmployeeBean>の件数が≠0 の場合

戻り値の 1 要素目:空文字("")

戻り値の 2 要素目:対応したエラーメッセージを「message」に入れて返す。

戻り値 ArrayList<EmployeeBean>の件数が=0 の場合

戻り値の 1 要素目:HTML(引数 EmployeeBean を元に作成)

•confirmUpdateUser メソッド

引数:EmployeeBean EmployeeBean 、String oldPassword

戻り値:ArrayList<String> message

処理詳細:

- 入力内容のエラーチェック

- 入力内容にエラーありの場合

戻り値の 1 要素目:空文字("")

戻り値の 2 要素目:対応したエラーメッセージを「message」に入れて返す。

- 入力内容にエラーなしの場合

戻り値の 1 要素目:HTML(引数 EmployeeBean を元に作成)

•confirmDeleteUser メソッド

引数:EmployeeBean EmployeeBean

戻り値:ArrayList<String> message

処理詳細:

引数 EmployeeBean を元に取得 SQL 文の編集

EmployeeDaoLogic (findEmployee メソッド) の引数にセットして呼び出す。

戻り値 > 0 の場合

戻り値の 1 要素目:HTML

戻り値 ≤ 0 の場合

戻り値の 1 要素目:空文字(””)

戻り値の 2 要素目:対応したエラーメッセージを「message」に入れて返す。

■model(Dao)

【 EmployeeSystemDAO.java 】

•updateEmployee メソッド

引数:String sql

戻り値:int result

処理詳細:

•SQL の実行

•引数 sql を元に SQL 文の実行 (executeUpdate())

処理が正常に完了した場合

戻り値 result に実行結果を入れて返す。

処理中に例外が発生した場合

戻り値 result に「-1」を入れて返す。

•findEmployee メソッド

引数:String sql

戻り値:ArrayList<EmployeeBean> employeeList

処理詳細:

•SQL の実行

•ArrayList<EmployeeBean>のインスタンス employeeList を作成

•引数 sql を元に SQL 文の実行 (executeQuery()) し、社員情報を取得

•取得した社員情報を EmployeeBean に格納し、employeeList に追加

•処理が正常に完了した場合

戻り値として employeeList を返す。

•処理中に例外が発生した場合

戻り値として null を返す。

■view(JSP)

【 login.jsp 】

- ・ログイン入力フォーム

<form action="/employeeAdmin/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" |
| user | hidden | value=社員ID |

【 employeeSystemMain.jsp 】

- ・検索入力フォーム

<form action="/employeeAdmin/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(開始) |
| toDate | select box | 入社年月(終了) |
| action | hidden | value="search" |

【 registerUser.jsp 】

- ・新規登録入力フォーム

<form action="/employeeAdmin/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 | 部署名 |
| divisionName | select | 課名 |
| positionName | select | 役職名 |
| positionMemo | text | 役職詳細 |
| naisenNumber | text | 内線番号 |
| publicCellphoneNumber | text | 業務携帯番号 |
| action | hidden | value="confirmNewUser" |

【updateUser.jsp】

・修正入力フォーム(管理用)

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

| パラメータ名 | Type | 備考 |
|-----------------------|--------|---------------------------|
| 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 | 部署名 |
| divisionName | select | 課名 |
| positionName | select | 役職名 |
| positionMemo | 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" |

・修正入力フォーム(一般用)

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

| パラメータ名 | Type | 備考 |
|-------------|--------|---------------------------|
| oldPassword | text | 旧社員パスワード |
| newPassword | text | 新社員パスワード |
| action | hidden | value="confirmUpdateUser" |

2、データベース詳細設計

データベース名:employee_admin

ユーザー名:root

パスワード:root00

・社員情報テーブル

テーブル名:employee

| フィールド名 | データ型 | 制約 | 備考 |
|-----------------------|-------------|---------------------------|-----------------------------|
| employeeId | INT | NOT NULL 、 PRIMARY KEY | 社員 ID |
| password | VARCHAR(20) | NOT NULL | パスワード |
| employeeName | VARCHAR(30) | NOT NULL | 名前 |
| kana | VARCHAR(50) | | ふりがな |
| gender | VARCHAR(1) | NOT NULL | 性別 |
| departmentCode | VARCHAR(3) | NOT NULL | 部署コード |
| divisionCode | VARCHAR(2) | | 課コード |
| positionCode | VARCHAR(1) | | 役職コード |
| positionMemo | VARCHAR(20) | | 役職詳細 |
| naisenNumber | VARCHAR(10) | | 内線番号 |
| publicCellphoneNumber | VARCHAR(13) | | 業務携帯番号 |
| adminFlag | TINYINT(1) | DEFAULT 0 | 管理者フラグ |
| registrationDateTime | DATETIME | | 登録日時 yyyy-MM-dd hh:mm:ss |

・拠点マスタ

テーブル名:base

| フィールド名 | データ型 | 制約 | 備考 |
|----------|-------------|---------------------------|-------|
| baseCode | VARCHAR(1) | NOT NULL 、 PRIMARY KEY | 拠点コード |
| baseName | VARCHAR(10) | NOT NULL | 拠点名 |

・部署マスタ

テーブル名:department

| フィールド名 | データ型 | 制約 | 備考 |
|-----------------|-------------|---------------------------|-------|
| department Code | VARCHAR(3) | NOT NULL 、 PRIMARY KEY | 部署コード |
| department Name | VARCHAR(20) | NOT NULL | 部署名 |
| baseCode | VARCHAR(1) | NOT NULL | 拠点コード |

・課マスタ

テーブル名:division

| フィールド名 | データ型 | 制約 | 備考 |
|--------------|-------------|---------------------------|------|
| divisionCode | VARCHAR(2) | NOT NULL 、 PRIMARY KEY | 課コード |
| divisionName | VARCHAR(20) | NOT NULL | 課名 |

・役職マスタ

テーブル名:position_table

| フィールド名 | データ型 | 制約 | 備考 |
|--------------|-------------|---------------------------|-------|
| positionCode | VARCHAR(1) | NOT NULL 、 PRIMARY KEY | 役職コード |
| positionName | VARCHAR(20) | NOT NULL | 役職名 |

・社員全情報ビューテーブル

ビュー名:employee_view

| フィールド名 | データ型 | 制約 | 備考 |
|-----------------------|-------------|---------------------------|-----------------------------|
| employeeId | INT | NOT NULL 、 PRIMARY KEY | 社員 ID |
| password | VARCHAR(20) | NOT NULL | パスワード |
| employeeName | VARCHAR(30) | NOT NULL | 名前 |
| kana | VARCHAR(50) | | ふりがな |
| gender | VARCHAR(1) | NOT NULL | 性別 |
| baseCode | VARCHAR(1) | NOT NULL | 拠点コード |
| baseName | VARCHAR(10) | NOT NULL | 拠点名 |
| departmentCode | VARCHAR(3) | NOT NULL | 部署コード |
| departmentName | VARCHAR(20) | NOT NULL | 部署名 |
| divisionCode | VARCHAR(2) | | 課コード |
| divisionName | VARCHAR(20) | NOT NULL | 課名 |
| positionCode | VARCHAR(1) | | 役職コード |
| positionName | VARCHAR(20) | NOT NULL | 役職名 |
| positionMemo | VARCHAR(20) | | 役職詳細 |
| naisenNumber | VARCHAR(10) | | 内線番号 |
| publicCellphoneNumber | VARCHAR(13) | | 業務携帯番号 |
| adminFlag | TINYINT(1) | DEFAULT 0 | 管理者フラグ |
| registrationDateTime | DATETIME | | 登録日時 yyyy-MM-dd hh:mm:ss |

3、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

- 仕様以外のフィールドは追加しないこと。
- private メソッドに関しては、仕様書に盛り込む必要はないが、必ずコメントで処理内容と引数と戻り値と呼び出し元を明記する。