**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 共通部品 Modelパッケージ

・Common.java 共通部品のクラス

src/employee/model 社員情報システム用Modelパッケージ

・EmployeeBean.java JavaBeansクラス

・EmployeeSystemLogic.java Logicクラス

■daoパッケージ

src/dao/employee 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/employee/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/employee 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

■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」のオブジェッ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アトリビュートを削除：

・「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

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」ペラメータが空文字“”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」のオブジェッ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」ペラメータが空文字“”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」のオブジェッ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 |
| deparetmentCode | 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" |

1. **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 |
| departmentCode | VARCHAR(3) | NOT　NULL 、  PRIMARY KEY | 部署コード |
| departmentName | 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 |

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

・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。