

# Employeess

## Vision

“Employees” is web-application with webservices which allows users to record, change, search and delete information about employees and departments.

Application should provide:

- Storing tables employees and departments with data in a database;
- Departments should store their names;
- Employees should store the following data: related department, employee name, date of birth, salary;
- Display list employees;
- Updating the list of employees (adding, editing, removing);
- Display list of departments;
- Updating the list of departments (adding, editing, removing);
- Populate database with the test data;
- Web service(RESTful) for CRUD operations returns data stored in the database;
- Web application uses web service to fetch the data from database;
- display a list of departments and the average salary (calculated automatically) for these departments;
- display a list of employees in the departments with an indication of the salary for each employee and a search field to search for employees born on a certain date or in the period between dates;



## 1. Employees

### 1.1. Display list employees

The mode is designed to view the list of employees, if it possible to display the list of employees for a specified range of birthdays.

Main scenario:

- User selects item “Employees”;
- Application displays list of Employees.

Employees	Departments																					
date from: <input type="text" value="01/08/1990"/>  to: <input type="text" value="01/09/1992"/>  <input type="button" value="Choose"/> <input type="button" value="Add New Employee"/>																						
<table border="1"><thead><tr><th>id</th><th>name</th><th>birthday</th><th>salary</th><th>department</th><th colspan="2">action</th></tr></thead><tbody><tr><td>1</td><td>Alex</td><td>01.02.1991</td><td>1000</td><td>web</td><td><input type="button" value="Edit"/></td><td><input type="button" value="Delete"/></td></tr><tr><td>2</td><td>Peter</td><td>04.05.1992</td><td>400</td><td>Android</td><td><input type="button" value="Edit"/></td><td><input type="button" value="Delete"/></td></tr></tbody></table>		id	name	birthday	salary	department	action		1	Alex	01.02.1991	1000	web	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>	2	Peter	04.05.1992	400	Android	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
id	name	birthday	salary	department	action																	
1	Alex	01.02.1991	1000	web	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>																
2	Peter	04.05.1992	400	Android	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>																

Pic. 1.1 View the Employees list.

The list displays the following columns:

- id – unique employee number;
- name – employee's name;
- birthday – date of birth of employee;
- salary – salary of employee;
- department – name of department;
- action – 2 buttons to edit the record and delete the record.

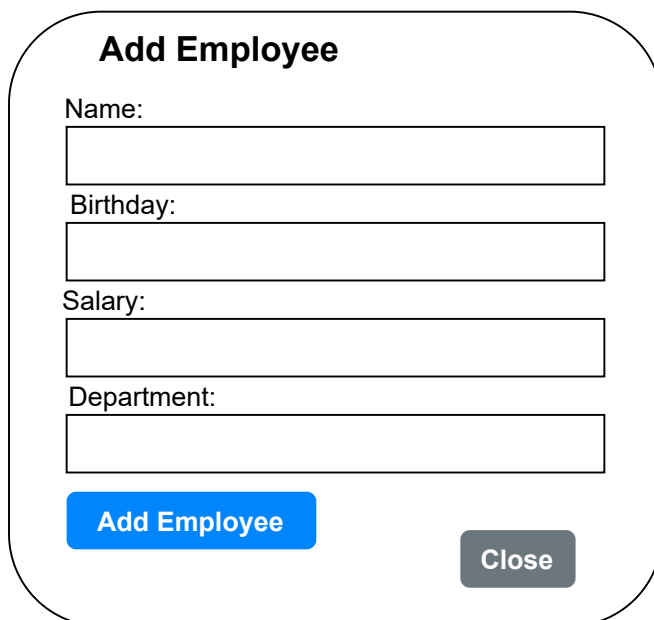
### ***Filtering by date:***

- In the order list view mode, the user sets a date filter and presses the refresh list button (to the right of the date entry field);
- The application will display a form to view the list of employees with updated data.

## **1.2 Add employee**

### ***Main scenario:***

- User clicks the “Add New Employee” button in the employees list view mode;
- Application displays form to enter employee data;
- User enters employee data and presses “Add Employee” button;
- If any data is entered incorrectly, incorrect data messages are displayed;
- If entered data is valid, then record is adding to database;
- If error occurs, then error message is displaying;
- If new employee record is successfully added, then list of employees with added records is displaying.

A screenshot of a web form titled "Add Employee" inside a rounded rectangle. The form contains four text input fields, each preceded by a label: "Name:", "Birthday:", "Salary:", and "Department:". Below the input fields are two buttons: a blue button labeled "Add Employee" and a grey button labeled "Close".

**Add Employee**

Name:

Birthday:

Salary:

Department:

**Add Employee** **Close**

Pic. 1.2 Add employee.

When adding an employee, the following details are entered:

- name – employee's name;
- birthday – date of employee's birthday;
- salary – employee's salary;
- department – department where is employee;

### ***Cancel operation scenario:***

- User clicks the “Add New Employee” button in the order list view mode;
- Application displays form to enter employee data;
- User enters employee data and presses “Close” button;
- Data don't save in data base, then list of employees records is displaying to user.
- If the user selects the menu item "Employees", or "Departments", the data will not be saved to the database and the corresponding form with updated data will be opened.

## **1.3 Edit employee.**

### **Main scenario:**

- User clicks the “Edit” button in the employees list view mode;
- Application displays form to enter employee data;
- User enters employee data and presses “Update” button;
- If any data is entered incorrectly, incorrect data messages are displayed;
- If entered data is valid, then edited data is added to database;
- If error occurs, then error message is displaying;
- If employee record is successfully edited, then list of employees with added records is displaying.

### **Cancel operation scenario:**

- User clicks the “Edit” button in the employees list view mode;
- Application displays form to enter employee data;
- User enters employee data and presses “Close” button;
- Data don’t save in data base, then list of employees records is displaying to user.
- If the user selects the menu item "Employees" or "Departments", the data will not be saved to the database and the corresponding form with updated data will be opened.

**Update information**

Name:

Birthday:

Salary:

Department:

**Update** **Close**

Pic. 1.3 Edit employee.

When editing an employee, the following details are entered:

- name – employee's name;
- birthday – date of employee's birthday;
- salary – employee's salary;
- department – department where is employee;

Constraints for data validation:

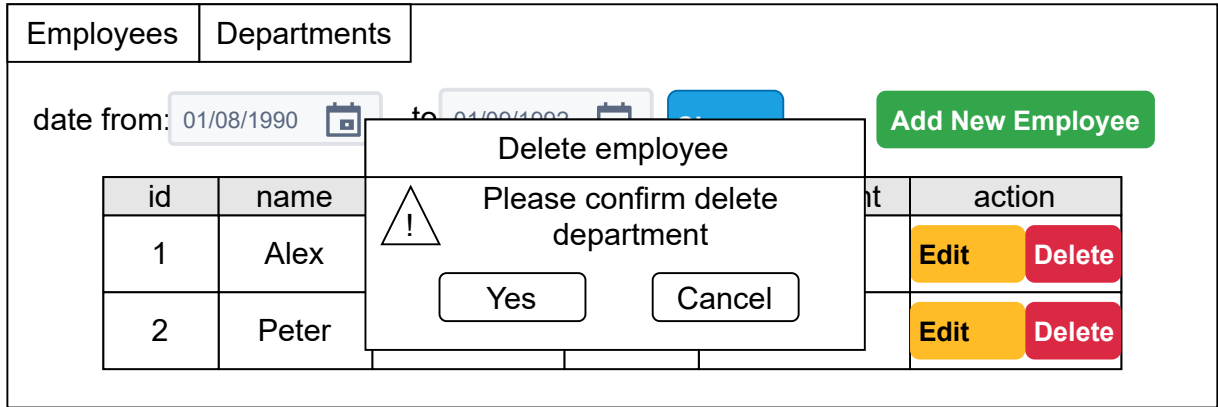
- name – maximum length of 100 characters;
- birthday – order add date in format yyyy-mm-dd;
- salary – integer type;
- department – maximum length of 100 characters;

## **1.4 Removing the order**

### **Main scenario:**

- The user, while in the list of employees, presses the "Delete" button in the selected employee line;

- If the employee can be removed, a confirmation dialog is displayed;
- The user confirms the removal of the employee;
- Record is deleted from database;
- If error occurs, then error message displays;
- If employee record is successfully deleted, then list of employees without deleted records is displaying.



Pic 1.4 Delete employee dialog

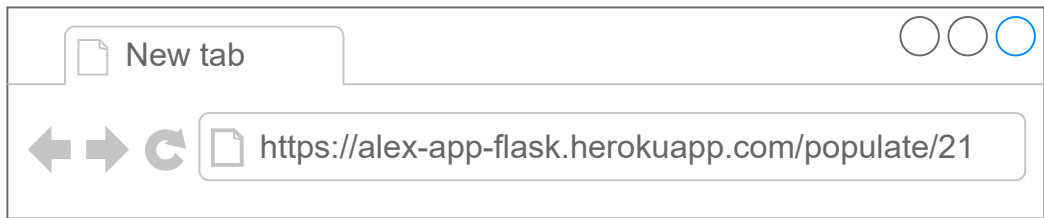
Cancel operation scenario:

- User is in display mode of employees list and press "Delete" button;
- Application displays confirmation dialog “Please confirm delete employee?”;
- User press “Cancel” button;
- List of departments without changes is displaying.

### 1.5 Populating DataBase with new employees

**Main scenario:**

- The user adds uri '/populate/[number]' to the host url of application, where number is any positive digit from 1 to 999 and push enter
- Application displays a List of new employees populated DB. Its random fake records

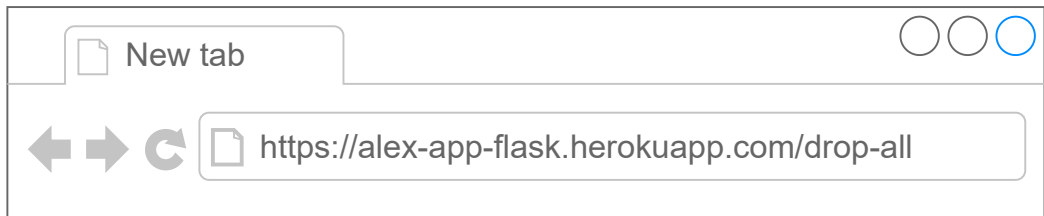


Pic 1.5 Populate DB example

### 1.6 Erase or drop DataBase

**Main scenario:**

- The user adds uri '/drop-all' to the host url of application and push enter
- Application displays a List of new employees populated DB. Its random fake records



Pic 1.6 Drop DB example

## 2. Departments

### 2.1 Display list of Departments

This mode is intended for viewing and editing the departments list.

#### **Main scenario:**

- User selects item “Departments”;
- Application displays list of departments.

Employees

Departments

Add New Department

id	name	Average Salary	action	
1	web	450	Edit	Delete
2	backend	670	Edit	Delete

The list displays the following columns:

- id – department's id;
- name – departments name;
- Average Salary – an average salary of department;
- action – buttons Edit and Delete to manipulate record.

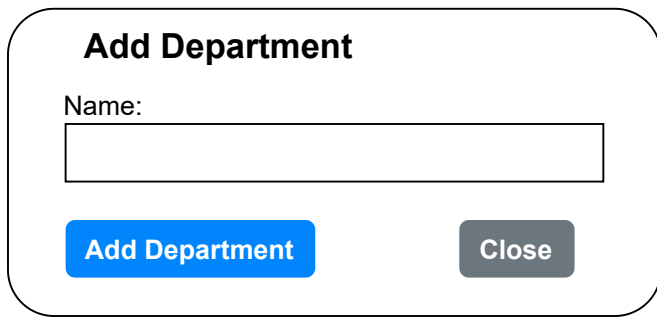
### 2.2 Add client

#### **Main scenario:**

- User clicks the “Add New Department” button in the departments list view mode;
- Application displays form to enter department data;
- User enters department’s data and presses “Update” button;
- If any data is entered incorrectly, incorrect data messages are displayed;
- If entered data is valid, then record is adding to database;
- If error occurs, then error message is displaying;
- If new deparment record is successfully added, then list of departments with added records is displaying.

#### **Cancel operation scenario:**

- User clicks the “Add New Department” button in the departments list view mode;
- Application displays form to enter department’s data;
- User enters department’s data and presses “Close” button;
- Data don’t save in data base, then list of departments records is displaying to user.
- If the user selects the menu item "Employees" or "Departments", the data will not be saved to the database and the corresponding form with updated data will be opened.

A screenshot of a web form titled "Add Department". It features a label "Name:" followed by a text input field. Below the input field are two buttons: a blue "Add Department" button and a grey "Close" button.

**Add Department**

Name:

**Add Department** **Close**

Pic. 2.2 Add department

When adding a department, the following details are entered:

- name - department's name

Constraints for data validation:

- name - maximum length of 100 characters;

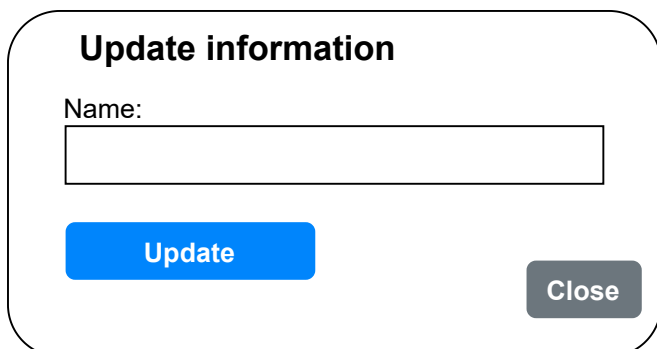
## 2.3 Edit department

### **Main scenario:**

- User clicks the “Edit” button in the departments list view mode;
- Application displays form to enter department data;
- User enters department’s data and presses “Update” button;
- If any data is entered incorrectly, incorrect data messages are displayed;
- If entered data is valid, then edited data is added to database;
- If error occurs, then error message is displaying;
- If department’s record is successfully edited, then list of departments with added records is displaying.

### **Cancel operation scenario:**

- User clicks the “Edit” button in the departments list view mode;
- Application displays form to enter department data;
- User enters department data and presses “Close” button;
- Data don’t save in data base, then list of departments records is displaying to user.
- If the user selects the menu item "Employees" or "Departments", the data will not be saved to the database and the corresponding form with updated data will be opened

A screenshot of a web form titled "Update information". It features a label "Name:" followed by a text input field. Below the input field are two buttons: a blue "Update" button and a grey "Close" button.

**Update information**

Name:

**Update** **Close**

Pic. 2.3 Edit department.

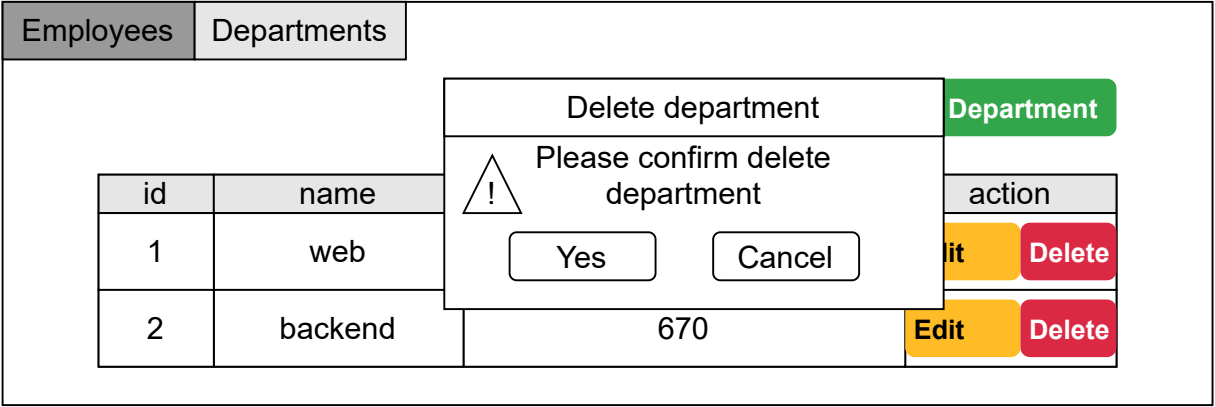
## 2.4 Removing client

### Main scenario:

- The user, while in the list of departments mode, presses the "Delete" button in the selected client line;
- Application displays confirmation dialog “Please confirm delete deparment?”;
- The user confirms the removal of the department;
- Record is deleted from database;
- If error occurs, then error message displays;
- If department record is successfully deleted, then list of departments without deleted records is displaying.

### Cancel operation scenario:

- User is in display mode of departments list and press “Delete” button;
- Application displays confirmation dialog “Please confirm delete department?”;
- User press “Cancel” button;
- List of departments without changes is displaying.



Pic. 2.4 Delete department dialog.