|  |  |
| --- | --- |
|  | **Requirements for Small Application Exercise** |
| Author: GTN | Topic: Small Application Exercise |
| Date: 18.05.21 |  |
| Valid: HMT, NQN |  |
| Version: 4.0 |  |

# Introduction

This document describes the requirements for a small project. When implementing this project, the reader is advised to use their knowledge learned during the previous weeks. This small project is applied for newcomers of .NET and Java group.

# Glossary

|  |  |
| --- | --- |
| **Term** | **Description** |
| PIM | Project Information Management |

# Business context

PIM is the tool used to manage administrative documentation of projects.

The tool has a lot of features; however, in this exercise newcomers are required to implement only two functions:

|  |  |
| --- | --- |
| **US #** | **Description** |
| 01 | New and update a project |
| 02 | Project list |

Out of scope:

* Role management
* Login, logout
* Help, New Customer/Supplier.

# User Story 01 – Project creation and updation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US01** | As a Project Manager, I want to create a new project entry in PIM tool so that I can quickly look up its information afterwards. I can also update information of an existing project. | | | |
| **Attributes** | **Feature** | Project Information | **Status** | Draft |
|  |  |  | **Version** | 0.1 |
|  | **Analyst** | ATN | **Sprint** | 1 |
|  | **Complexity point** | 1.5 | **Priority/BV** | High |

## Specifications

1. The Project Manager logs in to the tool and creates new project by clicking on the link “Project” under New menu.
2. He can also search for a project (refer to US02) and update it.

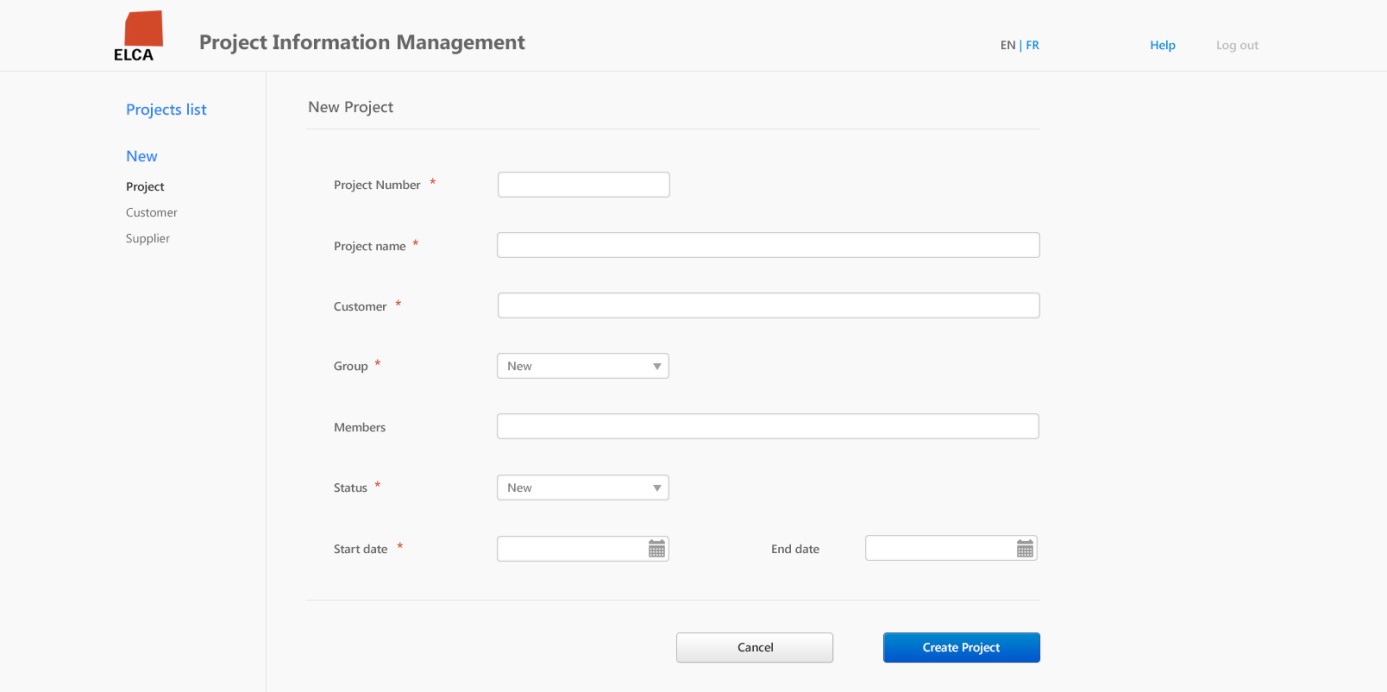
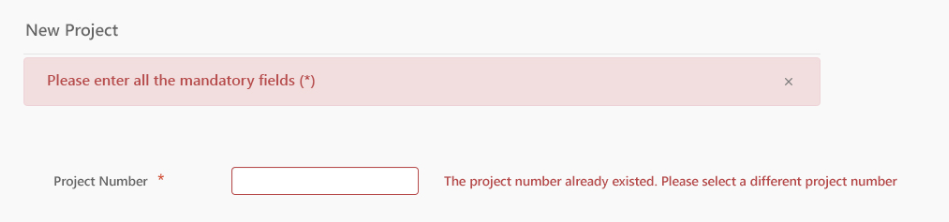
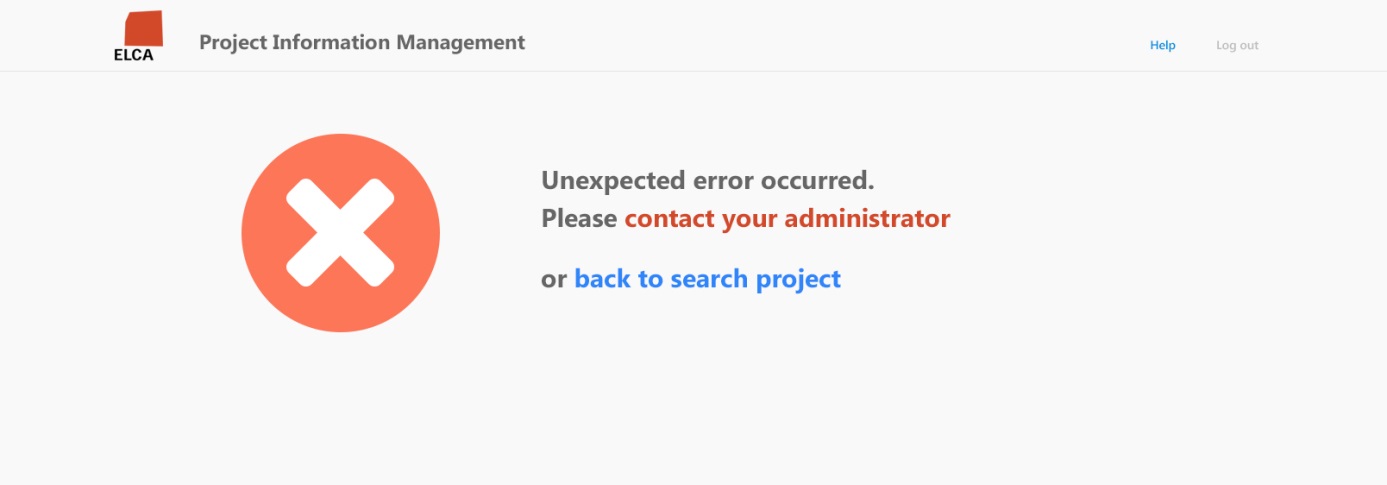


Figure .1 New project screen

1. Title “New Project” is changed to “Edit Project information” when user edits an existing project.
2. A project has following information: Number, Name, Customer, Group, Members, Status, Start Date, End date. Refer to Database section below for more information.
3. In Edit mode, Project Number cannot be changed, existing data must be filled in the form.
4. With new projects, status “New” is selected by default.
5. “Members” field can contain single or multiple visas, separated by comma (,). E.g: DTH, BHU, JHV.
6. Any fields that have validation errors are highlighted.



1. If some mandatory data is missing then display the following error message: “Please enter all the mandatory fields (\*).”
2. If the user creates a new project having number already existed, display the following error message: “The project number already existed. Please select a different project number”.
3. If a visa is not valid, i.e. does not exist, display the following error message: “The following visas do not exist: {visa list}.”
4. If there is unexpected error (e.g. network error, database connection error, etc.), redirect user to a dedicated error page and display the following error message: “Unexpected error occurred [<detail of the error>]. Please contact your administrator”.



1. If there is no error during saving the project, the users are navigated back to Projects list page.

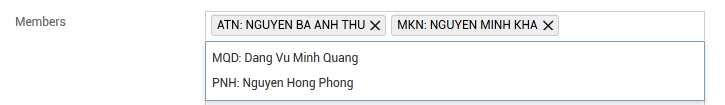
## Done Tests

|  |  |  |
| --- | --- | --- |
|  | **Test case description** | Result |
|  | When I create a new project and does not fill any values,  Then I receive error messages informing about required fields. |  |
|  | When I create a new project and fill values for all fields,  Then I can save the project successfully. The application is navigate to the Projects list screen. |  |
|  | When I update an existing project, filling all fields with new information,  Then the update information is successfully saved. |  |

## Advanced task

1. List of members is a suggestion box for multiple visas:

* When typing on the field, list of visas and corresponding names are listed.
* User selects a visa, the visa is added to the text box, as below.



# User Story 02 – Project list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **US02** | As a Project Manager, I want see list of projects. I can search for a project in order to look up its information and update it if necessary. | | | |
| **Attributes** | **Feature** | Project Information | **Status** | Draft |
|  |  |  | **Version** | 0.1 |
|  | **Analyst** | ATN | **Sprint** | 1 |
|  | **Complexity point** | 2 | **Priority/BV** | High |

## Specifications

1. When the user opens the application, Project list is displayed.
2. User can filter the project with search criteria: Name, Number, Customer, and Status.

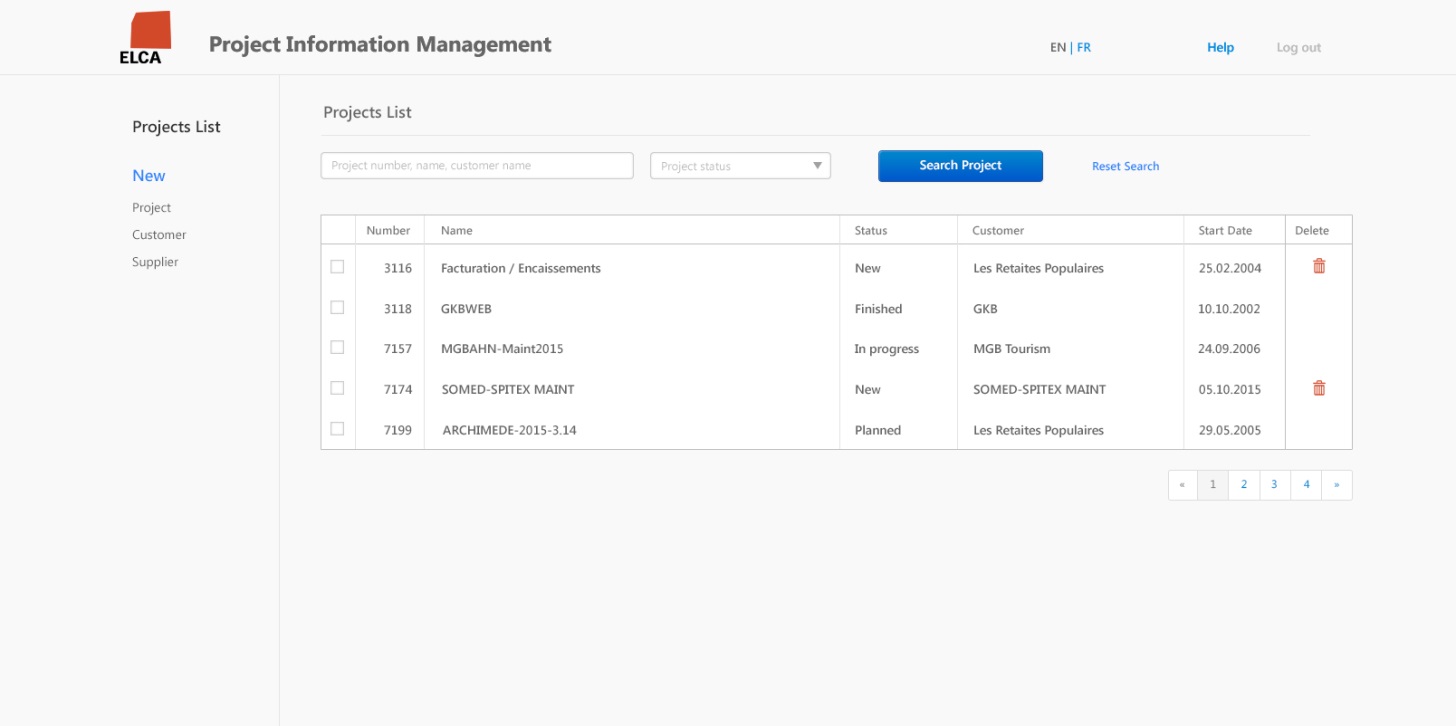
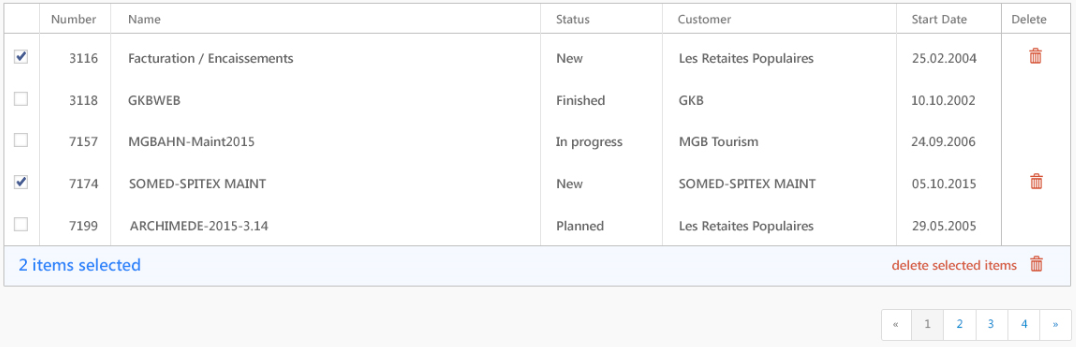


Figure .1 Project list screen

Note: Pagination is not required in basic level

1. When clicking on the Search button, the matching projects are displayed in a grid view as Figure 6.1. The search result is sorted by project number ascending.
2. If no criteria are entered, all projects are listed.
3. “Reset Search” button is used to clear the search criteria and return all projects in search result.
4. In the search result grid, the project number is displayed as a hyperlink. This link is used to navigate to Edit Project page (refer to US01) for updating the selected project information.
5. User can perform single deletion by delete icon or multiple projects deletion by selecting multiple projects and click on “Delete selected item”.
   1. A project can be deleted only its status is “New”.



1. When navigating to other screens from Project list page, current search criteria must be preserved. When user comes back to Project list, the current criteria are applied on both the search form and search results.

## Done Tests

|  |  |  |
| --- | --- | --- |
|  | **Test case description** | Result |
|  | When I enter the application,  Then the project list is displayed, sorted by Project number ascending. |  |
|  | When I enter the criterion into Search field (for Name, Number, Customer)  Then the list shows the projects of which names/number match the keyword. |  |
|  | Test the same for Status field. |  |
|  | When I choose a project and then click on button “Delete”.  Then the application just allows me to delete only the project has status “New”. |  |
|  | When I come to New/Edit screen perform the modification and then click on the button “Cancel” then application should navigate back to Project list screen with previously entered criterion is kept |  |

## Advanced

1. Pagination
2. Table with sorting headers
3. A button to show / hide advanced filter section. Advanced filter has more criteria: Project leader, containing member, >= Start date, <= Start date, >= End date, <= End date

# Database

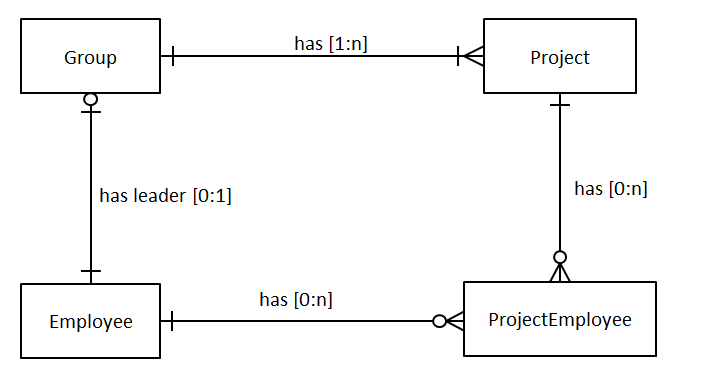


Table: PROJECT

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Mandatory** | **Description** |
| ID | NUMBER(19,0) | Yes | Surrogate key of the project table. |
| GROUP\_ID | NUMBER(19,0) | Yes | Foreign key refer to table Group that this project will belong to. |
| PROJECT\_NUMBER | NUMBER(4,0) | Yes | Project number. Unique |
| NAME | VARCHAR(50) | Yes | Project name |
| CUSTOMER | VARCHAR(50) | Yes | Customer name of the project |
| STATUS | CHAR(3) | Yes | Project status, possible values are: NEW, PLA, INP, FIN corresponding to  New, Planned, In progress, Finished. |
| START\_DATE | Date | Yes | Start date of the project. |
| END\_DATE | Date | No | End date of the project. If this value is not null, it must be later than the project start date. |
| VERSION | NUMBER(10,0) | Yes | Version control column |

Table: PROJECT\_EMPLOYEE

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Mandatory** | **Description** |
| PROJECT\_ID | NUMBER(19,0) | Yes | Foreign key refer to table Project |
| EMPLOYEE\_ID | NUMBER(19,0) | Yes | Foreign key refer to table Employee |

Table: EMPLOYEE

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Mandatory** | **Description** |
| ID | NUMBER(19,0) | Yes | Surrogate key of the Employee table. |
| VISA | CHAR(3) | Yes | Visa of the employee |
| FIRST\_NAME | VARCHAR(50) | Yes | First name of the employee |
| LAST\_NAME | VARCHAR(50) | Yes | Last name of the employee |
| BIRTH\_DATE | DATE | Yes | Birthdate of the employee |
| VERSION | NUMBER(10,0) | Yes | Version control column |

Table: GROUP

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Mandatory** | **Description** |
| ID | NUMBER(19,0) | Yes | Surrogate key of the Group table. |
| GROUP\_LEADER\_ID | NUMBER(19,0) | Yes | Foreign key refer to table Employee who is leader of this Group |
| VERSION | NUMBER(10,0) | Yes | Version control column |

# Technical notes .NET

1. Develop a 3-tier application using ASP.NET, Entity Framework, SQL Server
2. Use exceptions to handle errors in the application. For example, define a business exception ProjectNumberAlreadyExistsException to handle duplicated project number error.
3. Unexpected technical errors should be globally handled once for the whole application with global error handlers.
4. The application must handle concurrent update of Project information when updating and deleting.
5. The database transaction is handled at business layer.
6. Apply multilingual for all labels, messages inside the application.

# Technical notes Java

1. Develop a 3-tier application using Spring MVC, Spring, JPA / Hibernate.
2. Apply multilingual for all labels, messages inside the application.
3. Use exceptions to handle errors in the application. For example, define a business exception ProjectNumberAlreadyExistsException to handle duplicated project number error. Exceptions must be handled by **Spring’s exception resolver mechanism**.
4. Technical errors should be globally handled once for the whole application. The application must log errors with enough information for the developer to trace.
5. Changes in Spring configurations are not recommended, unless you have a good reason to.
6. The application must handle concurrent update of Project information when updating and deleting.
7. The database transaction is handled at business layer.
8. Apply QueryDSL to create database queries.
9. Entity mapping should share the id and version column from superclass.
10. Pay attention to Coding Best Practices section 2 (General programming) and 3 (Exception handling).

# General guidelines

1. All basic features, non-functional requirements and technical notes must be complete. Advanced features are nice to have.
2. Non-functional requirements in S25.2-ScreenLiveTestsWebGUI must pass for all applicable test cases.
3. Database tables must contain the exact columns in section 6, no more no less.
4. Aside from technical constraints, business validations must be implemented on backend.
5. Unused code from previous exercises must be removed.