Functional requirement document

Knowledge Matrix

To-Be

# Content

[Content 1](#_Toc494428958)

[1 Introduction 2](#_Toc494428959)

[1.1 Context 2](#_Toc494428960)

[1.2 Purpose 2](#_Toc494428961)

[2 General Requirements 3](#_Toc494428962)

[3 Overview Use Cases 4](#_Toc494428963)

[3.1 Manage Employees 4](#_Toc494428964)

[3.2 Manage Skill Sets 4](#_Toc494428965)

[3.3 Manage Knowledge Areas 5](#_Toc494428966)

[3.4 Manage Level 5](#_Toc494428967)

[3.5 Bulk Upload 5](#_Toc494428968)

[3.6 Generate Reports 5](#_Toc494428969)

[3.7 Version Management 7](#_Toc494428970)

[4 Use Case Descriptions 8](#_Toc494428971)

[4.1 Manage Re-counting 8](#_Toc494428972)

[4.2 Use case 2 9](#_Toc494428973)

# Introduction

## Context

Currently we are managing the knowledge matrix for all Java developers in an excel. This is a result of one of the initiatives started with SJES and currently being focused by craftsmanship track of strategic Java project.

Below is the high level overview of the activities involved:

1. CoE team prepared a template which is shared with all Java developers along with their team managers, with an expectation that the filled-in skill matrix will be shared back with COE team
2. The developers have filled-in the excel template (sometimes with discussions with team managers or colleagues). This is more like a self-declaration of their skills
3. The filled-in information from all developers is consolidated by COE and created an analysis report on the current skill levels, suggestions & recommendations.
4. This is shared with strategic Java project track owner and to management team.

Craftsmanship track of strategic Java project has a goal to ensure all our Java developers can work independently with a year.

The rest of the document tries to list the requirements assuming we are to create a web application which can fill this need.

## Purpose

This document is intended to provide the high level requirements for Java Skill Management system.

# General Requirements

Below are few general requirements to note:

1. The target system is assumed to be a Web Application which will use the following technologies
   1. Angular
   2. NewStack’s backend with REST service layer
   3. We can consider any Database, to make things simple can consider Oracle or H2 or Derby kind of databases as well
2. There will be different types of users / roles required
   1. Java Developers
      1. Developers will be filling-in their skill levels
   2. Team Managers
      1. Should be able to view the skill levels of all his/her developers
   3. COE
      1. Should be able to upload the existing java developer skill information through an excel
      2. Should be able to view the skill levels of all the developers
   4. Track Owner
      1. Should be able to upload additional information about all java developers like Team Manager, HOD, Project and/or Program, Location and so no
      2. Should be able to view and download advanced reports (descriptive analytics)
         1. Average Developer Skill scores against
            1. Team Managers
            2. Locations
            3. Programs / Projects
            4. Old Frames / New Stack / Angular (Categories / Classifications)
         2. Upload format can be CSV / Excel
         3. Report download format can be Excel

# Overview Use Cases

## Manage Employees

* Should be able to manage employees (developers) in the organization
  + Create, Modify new employees
  + Make employees active & inactive
* Details of employees:

|  |
| --- |
| **ID** |
| **Name** |
| **Team Manager** |
| **Head Of Department** |
| **Team /Service** |
| **Active** |

## Manage Skill Sets

* Should be able to create new & different knowledge skill sets.
  + For example: One skill set for Java and another for Mainframe
* Skill set will have following fields:

|  |
| --- |
| **ID** |
| **Name** |
| **Team Manager** |
| **Head Of Department** |
| **Team / Service** |
| **Framework / Tools** |
| **Domain** |
| **Knowledge Area** |
| **Level** |
| **Relevant Experience  (in months)** |
| **External Experience** |
| **Current Program (If Any)** |

* Each developer should be able to fill or upload the skill sets.
* A template of such a skill set is:
* 
* The form for capturing this skill set (the number of rows) is dependent of knowledge areas.

## Manage Knowledge Areas

* Knowledge Area is one of the fields in the skill set.
* Based on the technology (Java, Mainframe…etc), each employee should be able to provide skill set for each knowledge area.
* Based on these knowledge areas and technology (Java, MF…), there will be one row to capture skill of developer.
* Dynamically “Admin” should be able to add or remove knowledge areas.

## Manage Level

* While filling out the skill sets, developer has to fill the level.

|  |  |
| --- | --- |
| **How to fill in level ?** | |
|  |  |
| ​0 | ​No knowledge at all. |
| ​1 | ​You have been trained, but no hands on, so you definitively need coaching. |
| ​2 | ​You have been trained, little hands on, so you probably still need coaching. |
| ​3 | ​You have been trained, have hands on and can work independently |
| ​4 | ​You have been trained, have hands on, can work independently and feels comfortable to coach other developers |

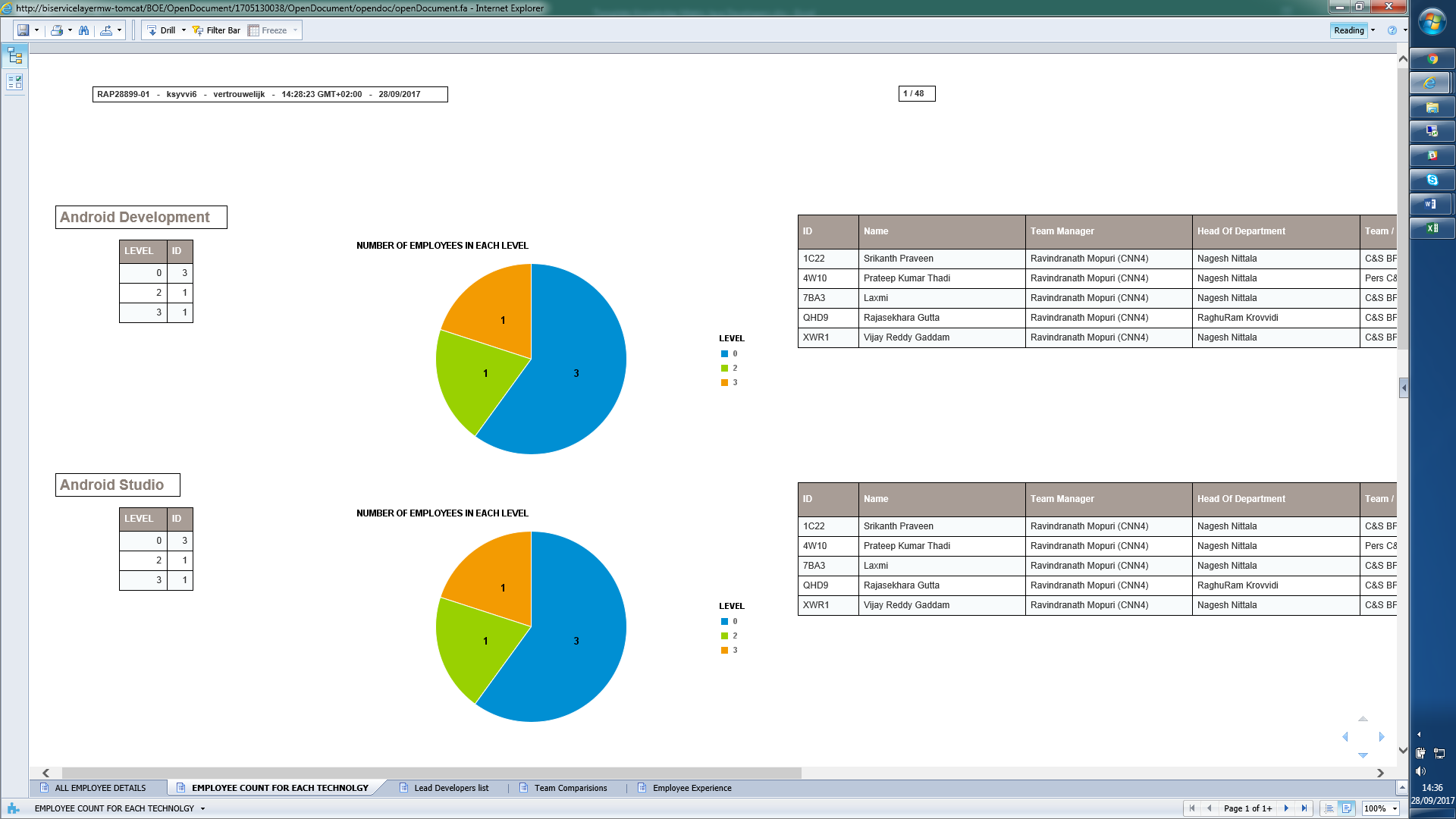
* Following is the definition of a level:

## Bulk Upload

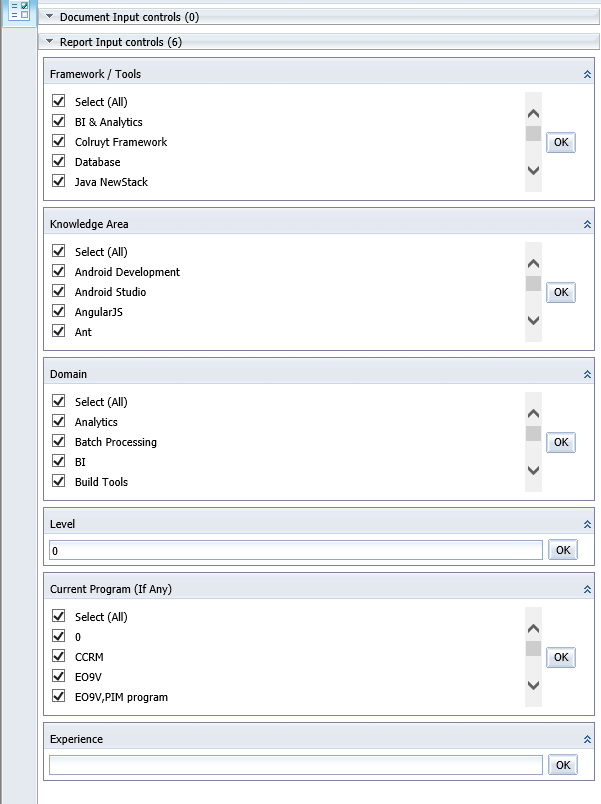
* It should be possible for CoE, Team manager or any other person who has access to upload the files for different employee (multiple files).
* One file per one developer.
* This file should be in the same format “Template Knowledge Matrix” for that domain (Java, Mainframe…).
* It is not required that this file should mandatorily have all the knowledge areas filled. If any knowledge area is not filled, by default for that knowledge area, level should be considered as “0”.

## Generate Reports

* It should be possible for CoE, Team Managers or other who have access to should be able to generate following reports:
  1. All Employee Details based on technology/Domain (for Java, MF,…)
  2. Count of employee for each level (as a pie graph) and for each knowledge area. Beside the graph, the table should be displayed with details.



* 1. Report based on the selection of knowledge area, framework tool, domain, level, experience and current program.



* 1. Criteria based Report:
     + Should be able to create a criteria on knowledge area/domain with level.
     + Based on this criteria report should be generated.
     + See an example below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Technical Criteria for Lead Developer NewStack** | | | |
| Java NewStack | JEE Technologies | JPA | 1. Atleast in 1 area level should be 4   2. Atleast in 1 area level should be minimum 3.   3. In rest of the areas level should be minimum 2. |
| Java NewStack | JEE Technologies | EJB |
| Java NewStack | JEE Technologies | JSF |
| Java NewStack | JEE Technologies | PrimeFaces |
| Java NewStack | JEE Technologies | Design Java NewStack Applications | 4. Level should be minimum 3 |
| Java NewStack | Integration Technologies | SOAP WebServices | 5. Atleast in 2 areas level should be 3. |
| Java NewStack | Integration Technologies | REST WebServices |
| Java NewStack | Integration Technologies | Design SOAP WebServices |
| Java NewStack | Integration Technologies | Design REST WebServices |

## Version Management

* **We plan to update the knowledge matrix every 6 months.**
* **So the previous 2 versions of each developer should be maintained to see the progress in learning.**
* **CoE, Team managers & others who have access to view the history should be able to view following information:**
  1. **Comparison of previous data and current data in the form of graphs.**
  2. **Highlight the major areas where knowledge level was increased or require to focus on.**

# Use Case Descriptions

|  |  |
| --- | --- |
| Use Case | Manage Employees |
| Brief Description and context |  |
| Primary Actor |  |
| Secondary Actor(s) |  |
| **Main Success Scenario** |  |
| **Main Success Pre Condition** |  |
| Condition Statement(s) |  |
| **Main Success Post Condition** |  |
| Condition Statement(s) |  |
| **Basic Path** |  |
| Path Description |  |
| **Alternate Path** |  |
| Path Description |  |

## Manage Re-counting

## Use case 2

|  |  |
| --- | --- |
| Use Case | Manage Skills Set |
| Brief Description and context |  |
| Primary Actor |  |
| Secondary Actor(s) |  |
| Functional Process/Activity |  |
| Logical Data Model |  |
| **Main Success Scenario** | MainScen\_[Scenario Name] |
| **Main Success Pre Condition** |  |
| Condition Statement(s) |  |
| **Main Success Post Condition** |  |
| Condition Statement(s) |  |
| **Basic Path** |  |
| Path Description |  |
| **Alternate Path** |  |
| Path Description |  |
| **Alternate Path** |  |
| Path Description |  |