Requirement Documentation
Employee Skill Tracker
Senior Project
Ben O'Truk
James C. Wolf
Mike Fleming
Table of Contents

1. Introduction
   1.1 Purpose
   1.2 Scope of Project
   1.3 Features
   1.4 Resources

2. Overview
   2.1 System Environment
   2.2 Accounts
      2.2.1 User
      2.2.2 Manager
      2.2.3 Admin
   2.3 Profile
      2.3.1 Team
      2.3.2 Skills
      2.3.3 Achievements
      2.3.4 Quests
   2.4 Teams
      2.4.1 Members
      2.4.2 Skills
      2.4.3 Awards
      2.4.4 Projects

3. Requirements
   3.1 Functional Requirements
      3.1.1 Add Employee
      3.1.2 Delete Employee
3.1.3 Edit Employee
3.1.4 Add Skill
3.1.5 Delete Skill
3.1.6 Find Employee
3.1.7 View Employee Information

3.2 Non-Functional Requirements
3.2.1 Gamification
3.2.1.1 Leaderboards
3.2.1.2 Achievements
3.2.2 Single Sign-on
3.2.3 Give Admin
3.2.4 Remove Admin
1. Introduction

1.1 Purpose

The purpose of this document is to describe the Employee Skill Tracker in detail. This document will cover the features, user interfaces, functional requirements, non-function requirements, and the overall purpose of the skill tracker. The project is being developed for Liaison Technologies, so that they may more effectively and easily track their employee's skills while letting said employees see how they have been rated, and how they stand up to the average employee on the team. Both the developers and clients should be able to read this document and get a decent idea of what this project entails. Once complete, it will be proposed to Liaison Technologies for their approval.

1.2 Scope of Project

This project will be for the employees at Liaison Technologies. This system will contain a database of skills that are associated with the employees. It will be designed to provide a way for management to track employees experience with certain skills. The skill sets associated with the employees will give management a way to assess a team member based on their experience level with certain programming languages and such. It will also allow the employee to complete "tasks" that will give him the ability to raise a certain skill set according to the specific challenge of that task. There will also be some other functions added to the system such as awards, quests, and leaderboards, so that the employee can see benefit after completing tasks or projects.

1.3 Features
1. Employee Skills Spreadsheet - This is only viewable by admins, and allows them to see all of the skills of their employees within their department.

2. Skills Database - database that tracks the skills of employees which is used for evaluation purposes and to learn what types of training needs to be provided.

3. Employee Profiles - A profile based system that employees can look up other employees.

1.4 Resources

   Node.js is an open source server-side development platform that is written in JavaScript.

   MongoDB is an open source NoSQL database software. Users skills and profile information will be kept inside a mongo database.

   Cloud 9 is an online integrated development environment that allows multiple users to edit the project at a single time. Cloud 9 also acts as a repository, since this project is stored on their servers.

   EditableGrid is an open source spreadsheet that will be used to represent the employees skills to the admins.

2. Overview

2.1 System Environment
The Liaison Sill Tracker has three sets of data: admins, employees, and teams. Admins are the ones with the most power within the system, and also belong to the Employee and Team data set. Employees also belong to the Team data set, and the Team data set only belongs to its own data set.

2.2 Accounts

2.2.1 User

The user account of the system would be associated with a normal employee that isn't in a management position. This account would have access to the basic functions of the system such as tasks, projects, and skill tracking of the users account. It would not allow the user to be able to track the skills of any other user.
2.2.3 Manager

The manager account of the system would be associated with an employee that is in a management position. This account would have access to certain employee's information depending on the team they belong to.

2.2.2 Admin

The admin account of the system would be associated with employees that are not only in a position status that is considered above management. This account would have access to every basic users account so that it could track, monitor, and assess each employee.

2.3 Profile
The profile will consist on non-sensitive information i.e. information that employees should be fine with sharing. A picture of the user, the total number of tasks that the employee has completed, which department/team the employee is on, a short about section that the employee can edit. Certain sections of the profile will be editable by the employee, where as other parts, such as the number of tasks complete, will not be editable by the user. Other employees should be able to click on their co-workers profiles and see the basic information about them. The management rating of the employee's skills will not be in the profile section.

2.4 Teams

Employees will also be broken up into teams. As of right now, there will be three teams; Support, Mapping, and Integration. Each team will have its respective employees. This encourages friendly competition between teams, which creates a better working environment. Each team will have a manager that can view/edit the employees that associated with it. Admins can also promote members to higher ranks inside the teams.

3. Requirements

3.1 Functional Requirements

3.1.1 Add Employee

The admin enters a new employee to the database.

1. The admin/manager selects Add employee.
2. The system presents a form.
3. The admin/manager fills in that form and hit "submit".
4. The system verifies the information and returns admin/manager to the main page.

3.1.2 Delete Employee
The admin/manager deletes a current employee from the database

1. The admin/manager selects Delete employee.
2. The system provides a simple form for search criteria to narrow down the list of employees.
3. The admin/manager selects the appropriate employee from the list and hits "submit".
4. The system removes the employee from the database and return the admin/manager to the main page.

3.1.3 Edit Employee

The admin/manager edits skill of a current employee

1. The admin/manager selects Edit employee.
2. The system provides a simple form for search criteria to narrow down the list of employees.
3. The admin/manager selects the appropriate employee from the list.
4. The system provide the admin/manager with access to the employees information.
5. The admin/manager edits the information then selects "submit".
6. The system returns admin/manager to the main page.

3.1.4 Add Skill

The admin/manager adds skill to the database

1. The admin/manager selects Add skill.
2. The system provides a form to fill out that will provide the information associated with that skill such as range of experience.
3. The admin/manager then adds a skill to the list and hits "submit".
4. The system returns admin/manager to the main page.

3.1.5 Delete Skill

The admin/manager deletes skill from the database

1. The admin/manager selects delete skill.

2. The system provides a list of skills associated with user accounts.

3. The admin/manager then selects and deletes the appropriate skills.

4. The system return admin/manager to the main page.

3.1.6 Find Employee

The admin/manager uses a search function find employees by name, team, or skills

1. Search box with keywords for employee names or teams.

2. Filter options allows the admin/manager to find employees with skills of a certain value.

3. Clicking the employee's name brings up the employee's information.

4. Icon available to edit an employee.

3.1.7 View Employee Information

The admin/manager views information of a selected employee

1. The admin/manager selects view information.

2. The system provides a search box to locate a specific employee.

3. The admin/manager selects appropriate employee from list.

4. The system provides admin/manager with employees information.

5. The admin/manager clicks "done" and the system returns admin to main screen.
3.2 Non Functional Requirements

3.2.1 Gamification

This feature deals with the gamification of the skill tracker. Currently, the developers have some free-reign on what the actual gamification will consist of. However, the gamification should encourage use of the skill tracker, and motivate employees to increase their skills.

3.2.1.1 Leaderboards

Leaderboards add a competitive aspect to the work area and will provide a fun way for employees to work harder. Leaderboards will be based off monthly progression between teams and individual members. These leaderboards can be based off tickets created or resolved, as well as projects completed. An XP system can be used with Admins adding bonus XP for tremendous accomplishments.

3.2.1.2 Achievements
Achievement systems are common systems used in games of all sorts. They allow the user to feel awarded by accomplishing some feat or reaching a milestone. Users will be able to receive badges and possible rewards for completing achievements. They will also be able to viewer their progress on an Achievement. Some examples of an Achievement would be reaching 100 tickets completed or work at Liaison for 1 year.

3.2.2 Single Sign-on
Allows users to login to the skill tracker with credentials that they should already have.

1. The user is presented with a sign-on page (If the user is already logged in, the sign-on page is skipped)
2. The system takes the user to their profile after a successful login.
3. The system takes admins/managers to their profile, but they can also go to another page that displays the appropriate employees attributes.

3.2.3 Give Admin
Lets Admins give administrative privileges to other users.

1. The admin goes to the desired users profile.
2. The admin chooses to add the user to the admin data set.
3. The admin is prompted again, to make sure that they want to add a new user to the admin data set.
4. The desired user is transferred to the admin dataset.

3.2.4 Remove Admin
Lets Admins remove administrative status from other admins.

1. The admin goes to the desired admins profile.
2. The admin chooses to remove admin status from the admin.
3. The admin is prompted again, to make sure that they want to move the admin back into the user data set.
4. The desired admin is transferred to the user dataset.