Customer Problem Statements and System Requirements

Noah Smith

Problem Statement

HR employees currently have a system that provides features for them to search for employees and change data, but the system does not provide the ability for the database to change the table that the employee is in if the employee job location changes. It also doesn't provide the employees to add new employees to the system or see who has made any changes to the database. Employees have also complained that the current system does not make it easy for them to navigate the current system without needing to navigate through different pages. The company needs a system that provides HR with a way that provides ease of access as well as additional functionality to the current system.

Table of Contents

Records – employee records that contain all information relevant to the company

HR – a human resources personnel who will be using the new system.

Container – used to specify an area in the user interface.

Functionality – adding, deleting, editing, and moving are just a few of the functionalities of the new system

System Requirements

No.	Priority Weight	Description
REQ-1	Low	The system should have the capacity to provide data on up to 10,000 employees
REQ-2	High	The system should allow the user to see who has made changes to the system
REQ-3	High	All necessary actions can be performed from a single toolbar at the top of the system
REQ-4	High	New employees must be able to be added as well as deleted from the system by the user
REQ-5	High	All changes must be documented by the system
REQ-6	Medium	Records must be viewable to employee where changes are being made
REQ-7	Low	Access to system is hassle-free and easy to learn.
REQ-8	High	Different types of data can be added to the system based on the needs of the company (i.e. salary, full-time/part-time etc.).

Nonfunctional Requirements

1) Usability – the system is easy for employees to use and provides users an easy way of manipulating the database and making all necessary changes
2) Functionality – The system provides the user with all necessary tools in order to perform the work necessary to complete their job. Additional functionality is added that allows for quick changes to the database and provides employees with the means to access, view, and change data in the database without needing to navigate through several pages to find the information they need.
3) Reliability - the system is simple and available to all HR employees as well as upper management. With it's simplicity, it allows for the quick recoverability and is accurate when applying the data to the system.
4) Performance – Given the simplicity of the system, the efficiency of the system would be easy to measure. It would also be able to scale easily considering that employees can be

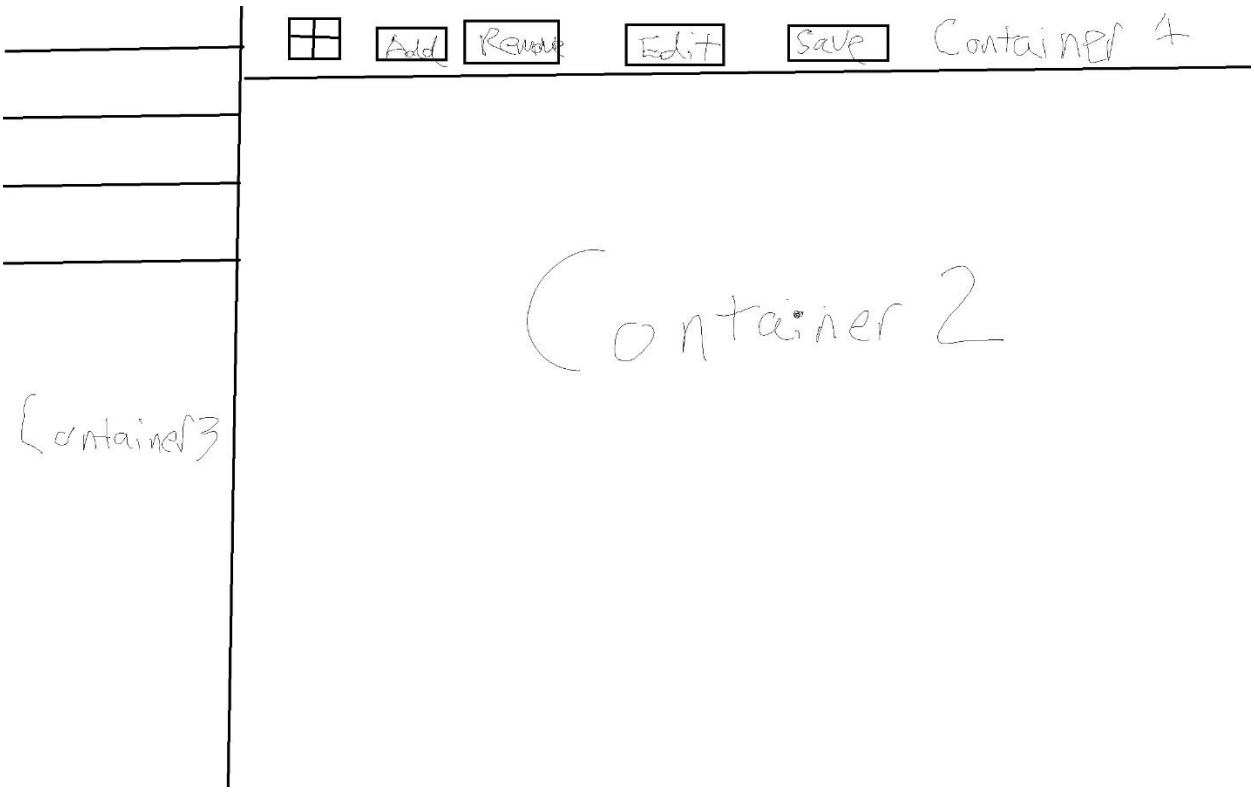
added to the system. The system would also be able to support enterprise level companies with ease yet still provide a simple interface to provide ease of use to the users.

5) Supportability – the system would be easy to test, and any bugs found after implementation would be easy to find and fix given the one page that the system would use.

User Interface Requirements

The user interface will require three main groups that we will call containers. The first container will contain the toolbox that will appear at the top of the screen that will provide additional buttons that will allow the user to perform different actions in the database. The second container will contain the information that needs to be changed and forms that will be used in order to complete these changes. It will also display information about employees in the database. The third container will contain previous searches and saved searches so the user can access the data if they need to go back to it.

In the following sketch container 1 is marked as “A”. Container 2 is marked as “B”. Container 3 is marked as “3”.



Plan of Work

√ Week 1-3: Determine which framework to use and establish front-end, back-end, and database connections.

Week 4: Build front-end of system. Create the system that the customers will see.

Week 5-7: Begin adding functionality through back-end implementation. Add features for HR employees to create reports, add employees, search employees, remove employees, move employees, log in, and view who made changes.

Week 8: Test features and record demo

Week 9-11: Improve features for HR employees based on feedback from customers. Add functionality for managers. View reports created by HR employees as well as make any changes that HR employees can make.

Week 12-14: Begin work on test cases for all features and make necessary changes in order to improve the system based on the result of the test cases.

Week 15: Record final presentation.

Currently I am finishing up setting up the environment I want to use for my system to function and will have this done at the beginning of next week. My focus next week will be using Angular to set up my front end and have that completed by the end of next week and then begin my back end features. Given the simplicity of the front end, I have changed the time frame for how much time I would need in order to complete this task.