Overview

The following is a simple programming exercise we expect you to complete and submit to us. You will be evaluated based on work submitted, and following that you will be called upon to an interview where we will ask questions based on your work.

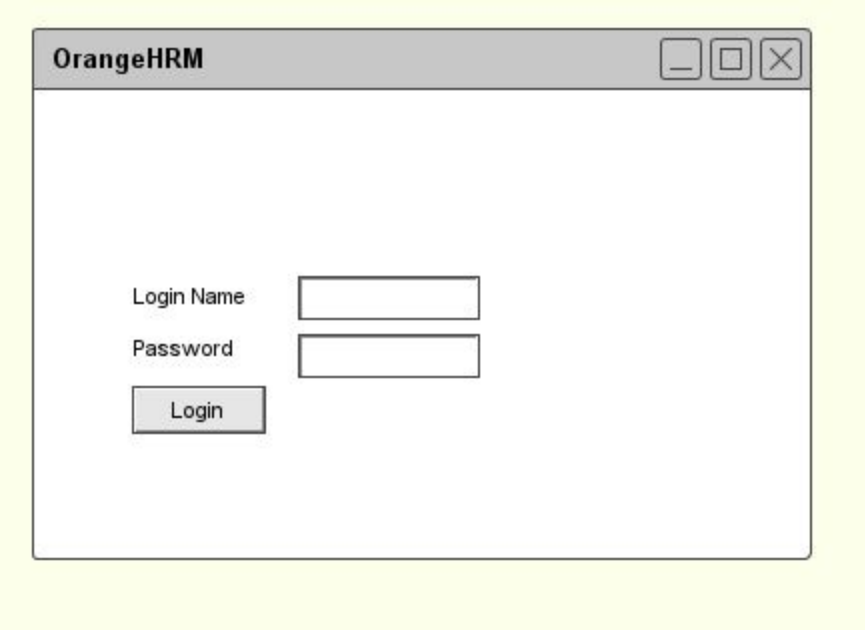
You're not expected to know the technologies mentioned below in depth, but we expect you to learn something new quickly, and apply that knowledge successfully to your work. We also expect you to be thorough with Object Oriented Programming concepts and Software Engineering best practices.

You can take any time to complete and submit the assignment. However we prefer to receive within 2 weeks, and we would be processing the candidates who submit completed assignments as and when we receive them. So the earlier you submit the completed assignment, the more of a competitive edge you will have over other candidates.

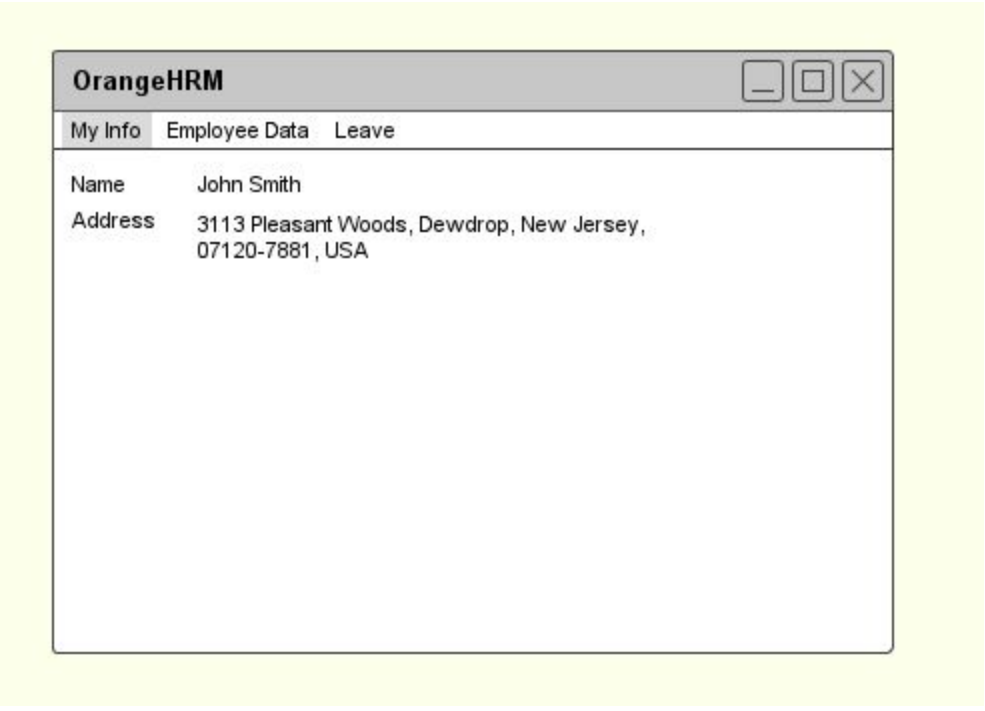
Assignment

Functional Requirements

- System should be able to store the following data for employees (No user interface is needed to enter these data. Test data can be entered directly to database):
    - Employee Table
        - Employee ID
        - Employees' first name and last name separately
        - Employees' address
    - User Table
        - User name (login name)
        - Encrypted password
        - Reference to employee (a user of system is also an employee) record in employee table
- Employee should be able to access a login page (using a browser), provide a valid user ID and password and log on to the system:



- System should prevent employees with invalid user id and passwords from logging in to the system.
- Upon successfully logging in, system should display the following screen to a user:



- My Info, Employee Data, and Leave should be separately clickable menu items on a top menu. “My Info” should be selected by default.
- My Info page should be automatically loaded upon a successful login.
- This page should show the full name of the logged in employee and the employee’s address as shown on the screen.
- If the user clicks “Employee Data” or “Leave” menus, load an empty page.

- If the user clicks “My Info” again, load My Info page again.

Technical Requirements

- Use MariaDB as the database
- Use Apache as the Web Server
- Backend App
    - Implement Restful APIs using Symfony PHP, MVC framework (http://symfony.com/). We don’t expect you to know Symfony. This is a test about how quickly you can learn something new and apply the knowledge to a simple problem.
    - Use Doctrine for accessing database: http://symfony.com/doc/current/book/doctrine.html
- Frontend App
    - Use Angular or React for the frontend application
- Include Unit tests written using PHPUnit. Include test cases for test scenarios:
    - Successful login
    - Failed login
    - Access of employee data for a logged in user
- Deliverables (in a .zip file or in github) should include:
    - Complete application and unit test source code
    - SQL for creating database and populating test data
    - Guide with instructions for:
        - installing
        - running unit tests
        - accessing system using browser

Submit completed assignments to: thilanka@orangehrm.us.com, buddikag@orangehrm.us.com with subject “SE - Programming Assignment”