

Background

You will be developing a solution to a mock business need. The goal of this exercise is for you to demonstrate your skills in collaborating with a client, designing a solution, and presenting the result for review by FPX Solution Development team members. You have the freedom to make this as feature rich as you wish, however your solution must be functionally viable. We realize real-world time limitations and will consider them if the resulting application is not functioning end-to-end.

In keeping with the scenario, we'll ask you to accept some constraints. An actual solution would execute within our existing environments, as such yours such execute within the Java Virtual Machine and function within a Tomcat server. In researching, you may find an out-of-the-box or pre-existing solution to this business need; we ask that you disregard them and consider the scenario as one where those are not available.

Business Need

We've noticed that we have a problem with some of the opportunities we've captured in Salesforce "getting stale". So we're thinking we need a real-time notification; something that keeps us informed before they become stale. In asking around there is some disagreement on when something becomes stale, so that should be something our managers can easily change.

Assignment

Having considered the problem space, your first task is to provide us, as the mock the client, a clearly stated solution proposal that defines your approach and enumerates your deliverables. We will review your proposal, perhaps redefining or clarifying some elements, with the goal of arriving at an agreed upon statement of work. Throughout the process we will be available to answer any questions or provide additional information.

Some topics you may wish to discuss in your solution:

- Restful vs. SOAP system exchange
- Your choice of libraries to reduce custom code
- Build systems
 - Automated testing
 - Code coverage
 - Dependency management
- Web frameworks of choice
- Knowledge of Salesforce
- Design Patterns
- Software architecture

Statement of Work for FPX Solution Development

Problem

Opportunities that are captured in Salesforce are becoming stale. A notification system is required to notify the interested members before this happens. The criteria for determining when an opportunity is about to become stale must be one that is configurable.

Context

Recognizing opportunities before they become stale allows the organization to take some action to remedy the situation. Lost opportunities result in lost revenue to the organization. There could be several reasons for why an opportunity is going to be stale. For example a sales representative is busy pursuing several other opportunities that they have neglected others. If the organization has the ability to detect when these opportunities are about to be stale, the organization could take some action like reassigning these neglected opportunities to some other sales representative, resulting in more wins for the organization.

Proposed Solution

This solution will allow the members of the organization to get notifications when opportunities within the system are about to be stale. The notification messages will be delivered to the owner of the opportunity and their managers through a web based interface. The criteria for determining when an opportunity is about to be stale can also be set using a web based interface. The following sections will describe the solution in two parts the web based user interface portion followed by the backend services.

User Interface

Authentication Interface

In order to access the notification system, the user must first get authenticated using their Salesforce account for their organization. A mockup of this interface is presented below. Authentication is done using the Salesforce [OAuth2](#) based authorization service. If the authentication is successful the notification service will obtain an access token and validate the user.

Stale Opportunity Notification
<Organization>

Salesforce Login

Email Address

Password

Login

Figure 1 Mockup Login Interface

Stale Opportunities Notification View

Once the user is authenticated they are presented with a list of the opportunities that are qualified as about-to-be stale. A mock up view is shown below. Using this view the user can dig into each item in the table, where they are presented with a detailed opportunity view. The detailed view will enable the user to find out the reasons (criteria), which caused the notification.

Stale Opportunity Notification

[Logout](#)

Opportunities

Criteria

▼ Opportunity	▼ Account Name	▼ Owner
Sale Services to Boeing	Boeing	John Doe
Sale Services to IBM	IBM	John Doe
Sale Services to Apple	Apple	John Doe

Figure 2 Opportunities List view

Criteria View

By selecting on the Criteria tab the user can also view the list of the criteria that they have set or those they have inherited from their managers. Using this view the user can edit, delete and add new criterion. In order to delete and edit a criterion the user must also be the owner. The Edit and Delete buttons on this view are only enabled when an item in the table is selected.

Stale Opportunity Notification			Logout
<div> <div>Opportunities</div> <div><i>Criteria</i></div> </div>			
<div> <div>New</div> <div>Edit</div> <div>Delete</div> </div>			
▼ Criteria	▼ Conditions	▼ Owner	
Approaching Closing	(Opportunity.CloseDate - CurrentDate) < 5	John Doe	
No Activity In More than 7 days	(CurrentDate - Opportunity.LastModifiedDate) > 7	Manager Richard S.	

Figure 3 Criteria List View

Logout

The users can logout of the notification system at any time, from any of the main views. A Logout link will be located at the right hand side of the page.

Stale Notification Criteria

The criteria are a set of conditions, which the opportunities are evaluated against to determine if they are about to be stale. These expressions must result in a Boolean value; a true value indicates that a given opportunity has qualified to be put into the about-to-be stale list. Once these conditions are true for a given opportunity, the system triggers a notification. These expressions can be a function of:

- Any property of the Opportunity object.
- The *CurrentDate* variable, which is an internal variable that the backend system monitors.

System View

A high-level system view is shown in the figure below. There are three key components to this solution:

- The front-end thin client web based user interface, which is described in the section User Interface.
- The service-side web server, which is used to serve the user interface.
- The streaming client that interfaces to the [SalesForce streaming service](#) and the actual notification processing service, which evaluates if an opportunity has failed a criteria.

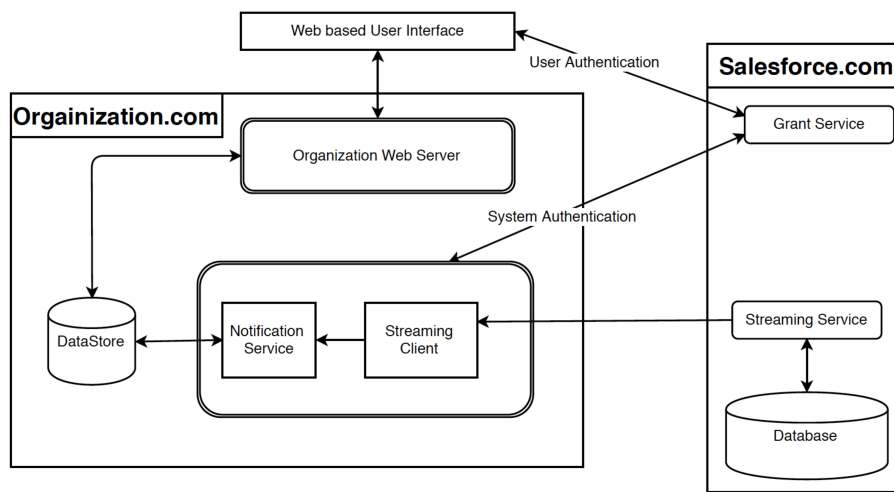


Figure 4 System View

The streaming client service subscribes to the SalesForce streaming service and any changes to the following SalesForce objects are received:

1. Opportunity
2. User
3. UserRole

The notification processing service is triggered by the following events:

- When an object we are tracking has been modified, i.e. a new streaming message has been received from SalesForce.
- A user adds or modifies a criterion. All opportunities that depend on the criterion are evaluated.
- At regular intervals, all opportunities that have their criteria expression depend on a time property are evaluated.