

Randori Care Vertical Demo

This document provides the initial project plans for the healthcare vertical demo, for a fictitious healthcare association, Randori Care. Unlike product focused demos, a vertical demo is focused on a particular business requirement – in this case, the scenario is medical information gathering/processing for the healthcare industry.

Randori Care is a private hospital located in the United States. The US Government recently initiated an effort to digitize the medical record systems nationwide, as covered on [ArsTechnica](#). As the setting for our demo, a patient will register to make an appointment with a doctor via an online form, presented using Flex, rather than with the traditional paper-based forms usually seen in doctor's offices. The workflow and back-end will be handled by LiveCycle.

Project Vision

The goal of this demo is to seamlessly integrate Flex and LiveCycle by leveraging the strengths of each product. Flex will provide the customer with a rich and dynamic interface, whereas LiveCycle will offer offline editing, barcode for paper processing, document security and backend workflow management.

Project Description

We are building a throw-away prototype. The front-end has two heads – a Flex interface and a PDF document interface. Both can be used interchangeably by the customer to input data to register for an appointment with a doctor. The user session however will be faked to reduce the project complexity.

Both interfaces connect to a custom written Java Servlet via HTTP GET/POST. The Servlet in turn invokes the Form Server or WorkFlow Server via the Java API. There is no database. All persistent data is stored as XML.

The workflow should be kept simple (no customer QPACS). It should have at least two sequential actions with at least one decision branch. The Policy Server will be integrated with the workflow if time permits.

Project Requirements

Front-end

- Since there will be two front-ends that the client can use, they should both be available at any time; the client can switch from one to the other with a single click
- Flex front-end has to be dynamic, featuring Flex's strengths with intuitive and engaging user interfaces; must look nice ☺
- PDF front-end must feature 2D barcode for paper form processing, if client decides to take the form offline

Custom Servlet

- Communicate with front-end via HTTP GET/POST
- Talks to back-end servers via the Java API
- Store data in XML
- No session support

Workflow

- At least two sequential actions
- At least one branch decision
- Avoid custom QPACs
- Policy Server optional