



Development of Automated Billing & BI Reporting for a Real Estate Law Firm

Business:

Product Development & Analytics

Domain:

Law

Tools:

Java, Pentaho, Apache POI, Alfresco, Angular 4

Key Highlights

Key Success:

- » Hundreds of hours of manual effort savings due to the automation of the system, leading to lower costs and higher productivity.
- » Lower TCO, owing to open source technologies being used.

Technology Stack:

Java, Pentaho, Apache POI, FreeMarker, Docx4j, Alfresco, AngularJS

Client

The client is a US based boutique law firm. They were embarking on a technology modernization program of their property tax appeals application which entailed building a billing system and other complementary functionalities.

Overview

A prominent Law firm in the American Midwest wanted to embark on a technology modernization program that involved automating their entire billing system along with its complementary functionalities. Also to be incorporated was BI reporting and security features for a seamless workflow. Indium Software's solution involved modernization of some legacy FoxPro components and complete automation of all workflows in the billing system.

1 Business Requirements

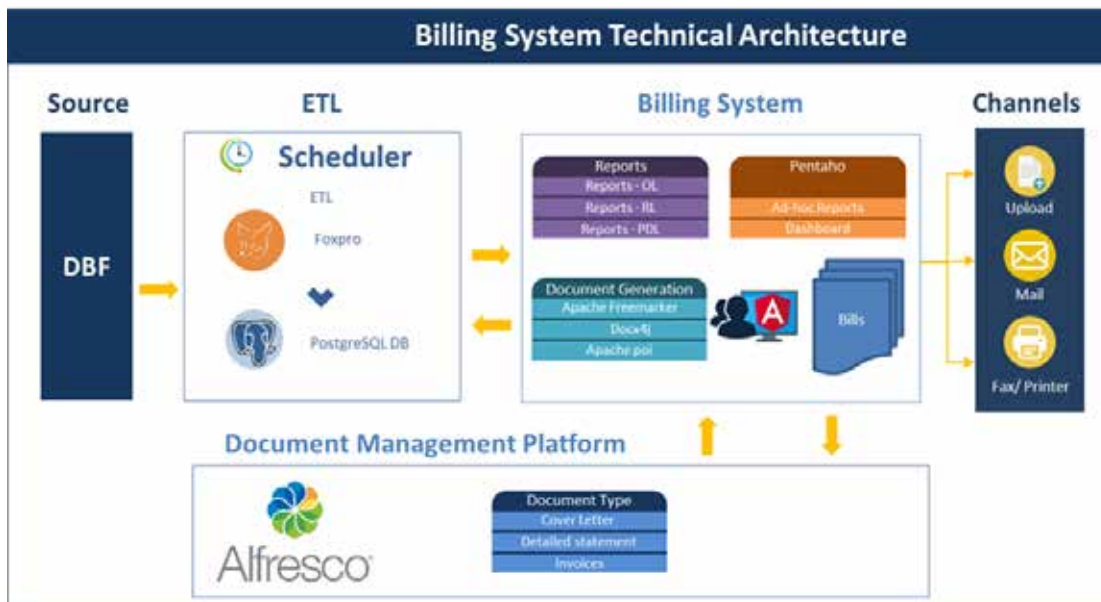
- » End-end billing infrastructure to enable full automation in the billing and appeals process.
- » Solution should offer flexibility to support customizations on various levels, a wide range of billing modules and manual intervention when required.
- » Billing suite should integrate seamlessly into other software programs (open and closed source), BI tools and security infrastructure for a seamless and streamlined workflow and a unified database from which comprehensive analytics and reporting is possible.

2 Solution

Indium Software's technical solution included:

- » After setting up the infrastructure, the first step was to extract data from the existing FoxPro database into PostgreSQL for historical and incremental data. Pentaho was chosen for the ETL work and the following guidelines were observed during the process:
 - » Input/output errors occasionally occurred when scheduled jobs were executing. Consequently a 'Check DB Connection' was inserted to delay and re-attempt to establish the connection 2 times before reporting an error and aborting the transformation.
 - » To maximize performance, 'Database Lookup' was used as opposed to 'Database Join' and the usage of 'Modified Java Script' & 'Select Values' steps in a transformation were kept to a minimum.
 - » Since the volume of data was huge, 'Unique Rows Has-set' and 'Memory Group By' steps were avoided in favour of 'Sort Rows', 'Group By' and 'Unique Rows' to achieve high speed and performance.
 - » By using 'Modify Java Script', intelligently and as sparingly as possible, 'Concat Fields', 'Replace String' etc. (which are even more damaging to performance) were neglected.
 - » Due to a glitch in one of the client's applications, the database was populated with millions of duplications. The result was that in the database, certain names were repeated a random number of times (in the millions each) but each with a different identification number in the Dimension Table. A separate ETL had to be written to transfer all non-unique names from the Dimension table into a temporary table. The unique codes of the temporary table were then compared with those in the Fact Tables and matches were deleted, thereby removing all duplicates.
 - » Detailed logs for each ETL were maintained to ensure quality and future error rectification.
 - » Check Points of jobs were properly maintained to ensure encountering an error does not cause the data to need to be re-loaded again from the beginning but instead will continue from where it stopped.
- » A Java API scheduler was used to run data extraction jobs automatically.
- » A data model for the billing system was developed and implemented.
- » The team then designed and developed a complete Billing system with all workflows.
- » All billing process like intimation letter emails, reminders, billing calculations etc. and other processes, letters were automated with options to manually review before automated dispatch.
- » Once billing data was generated, the system sent the cover letters, charge details, and invoices through email, with an option to download statements. An open source Java stack was utilized for this development.
- » A User interface that allows users to create various or custom cover letter, invoice, charge details templates and store it into Alfresco was then developed.
- » Angular 6 was used for front-end and backend development web services using spring boot / NodeJS.
- » BI reports were generated using Pentaho BI and the facility for the client's employees to generate ad hoc reports to the client's specifications was set up.





Billing System's high-level Technical Architecture

3 Business Impact

Our technology choices for this solution will address the following aspects for the client:

- » The automation the new billing system saved hundreds of hours of employee's work time over the course of a year.
- » Significant cost savings achieved due to the boost to employee productivity.
- » Freeing up employee's time and the scalable nature of the billing solution equipped the client could pursue growth unhindered.
- » Lower TCO – Many of the technologies chosen are open source which gives a cost advantage.
- » Cloud Ready – the system could be switched to cloud infra with limited effort and in future if required.
- » Mobile Ready – the UI and reports could be ported to Mobile with limited effort and cost in future if required.



INDIA
 Chennai
 +91 44 6606 9100
 Bengaluru
 +91 80 4645 7777
 Mumbai
 +91 022 6215 4028

USA
 Cupertino | Princeton
 Toll-free: 1 888 207 5969
SINGAPORE
 +65 9630 7959

UK
 London
 +44 773 653 9098

General Inquiries : info@indiumsoftware.com
 Sales Inquiries : sales@indiumsoftware.com