Case Study

.client overview

The client is a leading U.S community bank serving the under-served and under-represented consumers and small businesses from the Hispanic community, for nearly 100 years in the Central valley & LA areas of California.

The bank has merged with a for-profit social enterprise bank, whose economic interests are owned by a certified B Corporation honored in the top 10% of all B Corps.

The combined entity has over $600 MN in assets and a larger footprint – California, Oregon & Washington.

Business Requirements

The bank required integration of new APIs to the existing Loan decision system that processes consumer loans and extend it to the Cloud, and launch it in Nov 16.

Additionally, the bank was looking to upgrade the entire Loan decision system for richer UI, higher performance and scalability, extensibility, and ease of maintenance.

The upgraded Loan decision system would be rolled out to all branches.

Key Challenges

The existing Loan decision system, built in Dot Net technologies, had minimal functional specification documents (FSD).

The Loan decision system has to be revamped without breaking any existing functionality, while integrating new APIs.
Our Approach

- The team followed agile methodology and worked closely with the client’s business team in sprints, to understand all existing functionality and corresponding business logic.
- The team analyzed all scripts/modules that get triggered at back-end while using the loan decision system, and then confirmed the functionality & corresponding business logic from the client.
- As and when clarity on functionality & business logic was gained, it was coded in UI screens which were built in parallel.

Results

- After UAT was completed for APIs delivered, the bank’s Senior Management highly appreciated the team for the quality of work delivered.
- The bank has awarded Indium another project for a Cloud-based solution for commercial loans.

Our Solutions

- The Loan decision system was rewritten in Java technologies. AngularJS and Spring MVC architecture were leveraged for richer UI, enhanced performance, extensibility and scalability.
- The existing MS SQL Server database was reused to reduce time & effort, which would have otherwise been spent on migration.