QBurst | A High AI-Q™ Company

Driving Rapid Application Development for the Banking and Financial Services Industry

Strengthening a core banking software leader's development platform with a modular architecture, Vaadin 24, and enhanced tooling to enable rapid, scalable, and compliant financial application development.

Overview

A North American leader in banking CRM software required a major upgrade to their signature application development platform to address scalability and efficiency issues stemming from legacy architecture and components.

- Strengthened the platform's architectural foundation, making it modular and composable using Java 21/Spring Boot 3 to enhance usability, scalability, and maintainability.
- Refined the Integrated Development Environment (IDE) to support smoother domain modeling and scripting workflows, accelerating internal team adoption.
- Achieved 55% reduction in time-to-market for new financial applications and 40% faster queries after load testing scalability with over one million records.
- Ensured future compliance by upgrading the UI framework to Vaadin 24 for ADA & WCAG compliance and implementing audit and access logs.

Client Profile

Based in North America, the client is a global leader specializing in SaaS solutions for the banking and financial sector. Their software is used to streamline operations, enhance customer experiences, and serve as the core-banking system foundation for numerous financial institutions.

Challenges: Architectural Debt and Sluggish Development

The existing development framework was holding back the client's ability to innovate and scale:

- **Scalability Limitations:** The architecture lacked the necessary capacity and efficiency to handle increasing transaction volumes and workloads.
- **Developer Velocity:** Low developer bandwidth and the risk of disrupting existing features made delivering new financial solutions a slow, costly process.
- **Technological Obsolescence:** Legacy development components required urgent technological upgrades to meet current enterprise-grade performance and compliance requirements.
- **Legacy Complexity:** Years of custom widgets, extensions, and configuration quirks in the legacy system made maintenance and upgrades highly complex.

Composable, Future-Ready FinTech Platform

QBurst strengthened the architectural foundation of the client's Rapid Application Development (RAD) framework, transforming it into a composable and modular platform using modern technologies like Java 21 and Spring Boot 3.

The platform was designed to provide consistent, reusable components across the client's entire suite of banking applications.

- **Architectural Modularity:** The framework was refactored for improved usability, scalability, and maintainability, specifically tackling and untangling years of accumulated legacy adaptations.
- **IDE & Workflow Refinement:** The Integrated Development Environment was enhanced to better support domain modeling (creating relationships, attributes, and constraints) and scripting workflows, enabling smoother adoption and faster development by internal teams.
- **Optimized Rule Execution:** The rule execution engine was optimized to efficiently handle both batch and real-time scenarios, with seamless integration of scripting capabilities into the data repository.
- **Compliance & Performance:** Upgraded the UI framework to Vaadin 24 to achieve ADA and WCAG compliance. Performance profiling and indexing were implemented to meet enterprise-grade responsiveness and throughput.

Technical Highlights

- **Core Upgrade:** Modernized the platform to Java 21 and Spring Boot 3 for enhanced performance and security.
- **Compliance Framework:** Upgraded to Vaadin 24 to immediately address strict ADA & WCAG accessibility standards.
- **Versioned Data Repository:** Upgraded UI with editing and rollback capabilities on the Postgres DB, ensuring data integrity across different versions.
- **Performance Engineering:** Utilized tools like YourKit Java Profiler to optimize the rule engine with faster queries via indexing and horizontal scaling (supported by Hazelcast and Kafka).
- **Scripting Runtime:** Introduced a flexible scripting language to define and execute complex business logic dynamically.

Impact

The modernization of the RAD framework resulted in direct, measurable improvements across the client's product development lifecycle:

- **55% Reduction in Time-to-Market:** The use of reusable components, upgraded tooling, and streamlined workflows significantly accelerated the delivery of new financial applications.
- **Improved Scalability and Speed:** Load testing validated scalability with over one million records, achieving 40% faster queries and horizontal scaling.
- **Operational Efficiency:** Automation of rule-based processes via the optimized execution engine reduced manual effort and minimized the potential for human errors.
- **Future-Ready and Compliant:** The framework's extensibility, along with the Vaadin 24 upgrade and logging capabilities, ensures the platform meets evolving financial industry compliance and accessibility needs.