SMU Classification: Restricted

Multi-tenant Real-time General Ledger (GL) System

Ma Ningzhi

SMU
SINGAPORE MANAGEMENT UNIVERSITY

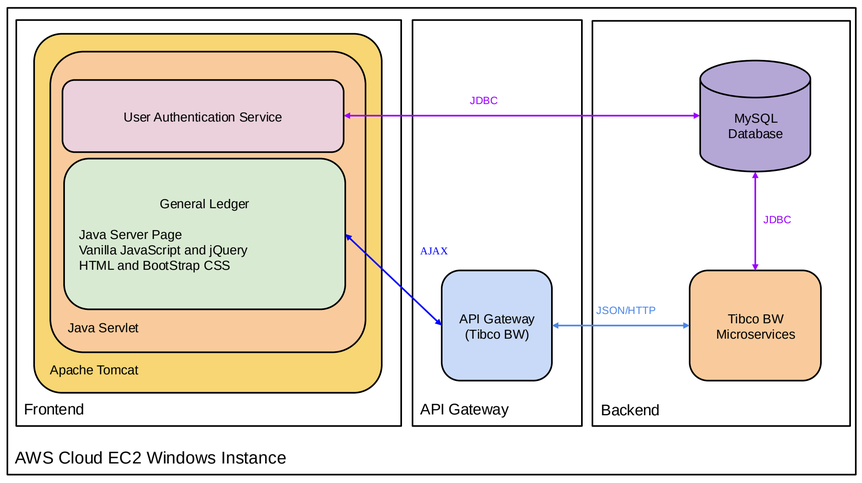
1. Overview

About tBank GL
SMU tBank is a suite of FinTech apps developed to simulate real-world banking operations for hands-on learning. (1) Within the suite, tBank GL consolidates and manages transactions, providing an overview of each bank's financial position.

Goal
This project aims to effectively migrate the current tBank GL Application from the current tech stack of vanilla JavaScript, jQuery, Java Sever Page, TIBCO BusinessWorks and MySQL to Outsystems Cloud.

Motivation
1. Technological advancement
2. Maintainability and cost efficiency
3. Integration with Narfin Platform

2. As-Is

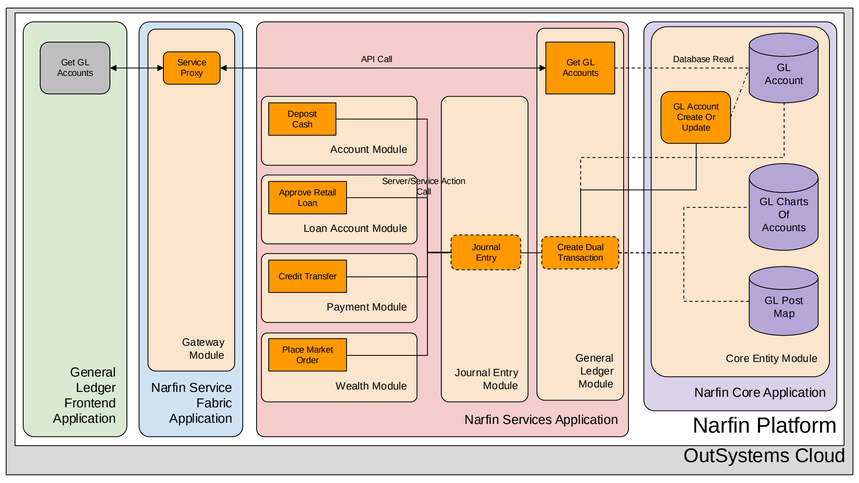


Outdated Technology
Legacy tech stack with limited support, lack of relevance and usability, as well as high maintenance costs.

Limited Maintainability
Lack of skilled student developers due to outdated technologies.

UI & UX Issues
Basic, inconsistent UI with little interactivity and poor user engagement.

3. To-Be

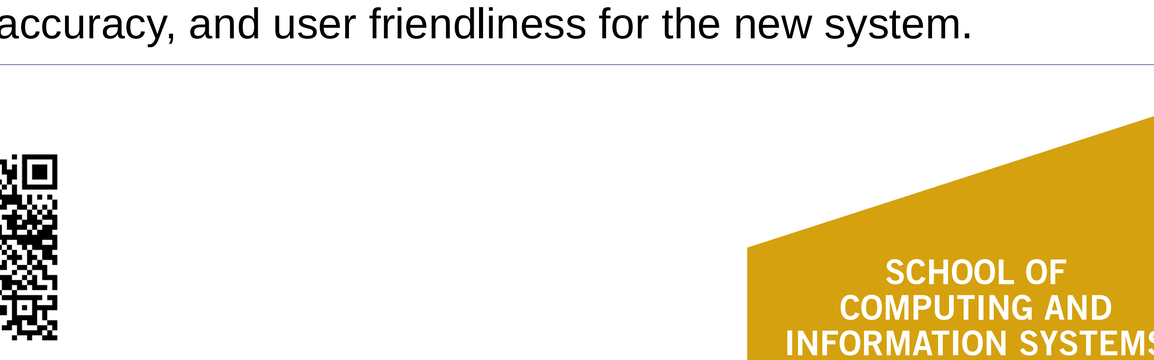


Improved Maintainability
Modern tech stack with OutSystems for reduced maintenance costs and streamlined development process.

Integration with Narfin Platform
Seamless integration with multi-tenant support, and future expansion options.

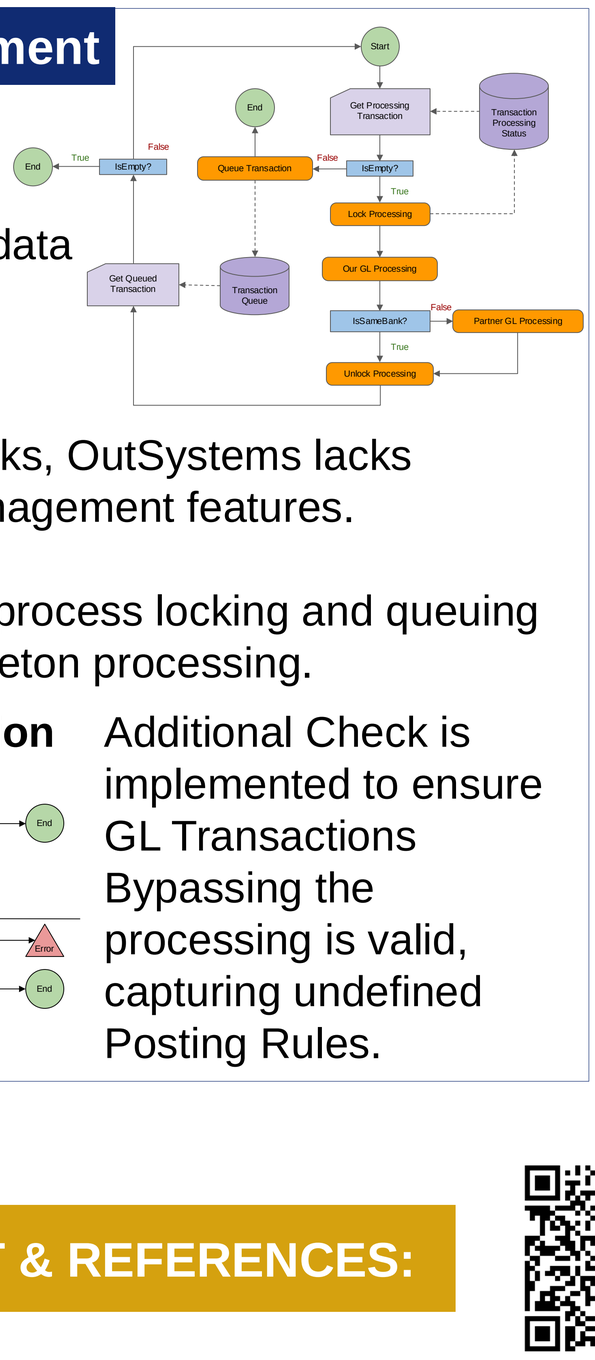
Security and Real-Time Processing
User authentication, secured APIs (OAuth 2.0), role-based access, and efficient real-time data processing.

Enhanced UI & UX
User-friendly design, easy navigation, responsive controls, and automated data loading for better UX.



4. Challenge & Improvement

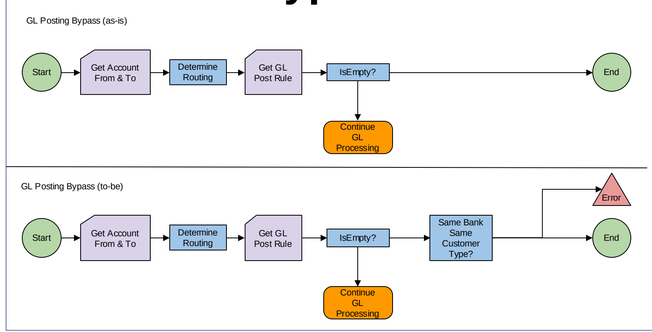
Real-Time Processing
Handling concurrent GL transactions poses risks of data inconsistencies and wrong balance calculations.



OutSystems' Limitation
Unlike TIBCO BusinessWorks, OutSystems lacks deliberate, strict thread management features.

Custom Solution
Implemented a transaction process locking and queuing mechanism to enforce singleton processing.

Enhanced Bypass Validation
Additional Check is implemented to ensure GL Transactions Bypassing the processing is valid, capturing undefined Posting Rules.



5. Learning Points

Effective Migration
Migration from legacy to modern platform requires thoughtful handling of outdated tech, data and UIUX to mitigate the differences and be well-integrated into the new platform.

Concurrency & Security
Real-time financial transactions processing required robust, deliberate controls for accuracy and secure access.

Improvements & Enhancement
Leverage new platform's features to enhance the reliability, accuracy, and user friendliness for the new system.

ACKNOWLEDGEMENT & REFERENCES:

SCHOOL OF COMPUTING AND INFORMATION SYSTEMS