Loan Management System

Contents

[1.0 Problem statement 2](#_Toc44598507)

[2.0 Technical Skills 2](#_Toc44598508)

[3.0 Architecture Diagram 3](#_Toc44598509)

[4.0 User Stories 3](#_Toc44598510)

[5.0 Expected Deliverables 4](#_Toc44598511)

[6.0 Milestone and duration 5](#_Toc44598512)

[7.0 Implementation Notes 5](#_Toc44598513)

# Problem statement

Loan management system manages the process of originating Loan orders, modify/update and cancellation of loan related information. The Loan information accounts for the loan number, loan amount, loan term, borrower information, property information, status, loan management fees, origination date, origination account, lien information, legal documents and loan history.

The main components that make up the loan are as follows:

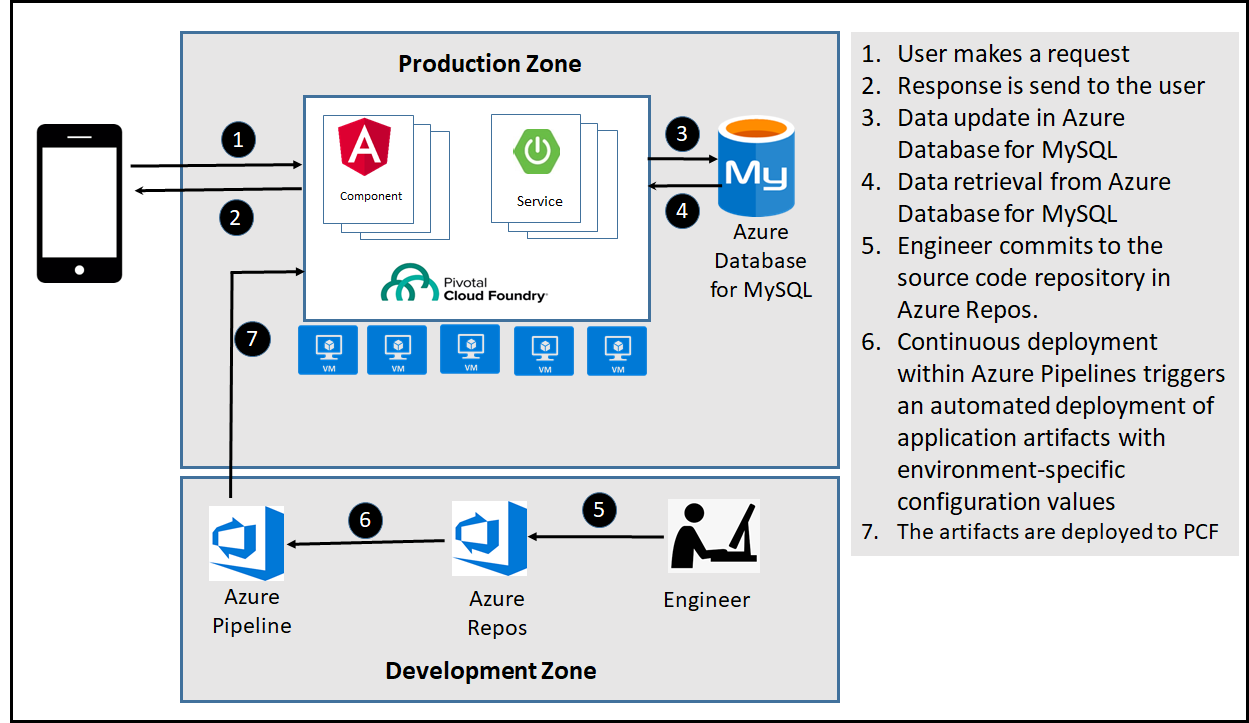
1. Borrower information: details of the person or party borrowing the loan amount
2. Property information: details of the property for which loan is originated
3. Lien information: information related to the taxes and bill amounts liable to be paid to various taxing authorities
4. Legal documents: legal documents related to the property and ownership related information

The scope of this project is to deploy an application using the latest state-of-the-art technology stack on Pivotal Cloud platform and Microsoft Azure

# Technical Skills

|  |  |
| --- | --- |
| **Layers** | **Skills** |
| Frontend | HTML5, CSS3, Angular, TypeSript |
| Backend | Java 8  Spring Boot, Spring Cloud, Spring Data JPA  MySQL  SpringTest/JUnit  SonarQube/SonarCloud |
| DevOps | Azure DevOps |
| Cloud | PCF |

# Architecture Diagram

.

# User Stories

|  |  |
| --- | --- |
| User Story # | User Story |
| US\_01 | As a user, I should be to login the Loan management application using User id and Password. I should be able to logout of the system safely.  Acceptance criteria:  I should be able to login with a User Id and Password that exists in database. On clicking logout the session should be invalidated and login page must be displayed |
| US\_02 | As a user, I should be able to search for loan information existing in the system. I should not have the ability modify or cancel the loan.  Acceptance criteria:  The search screen must have User’s First name, Last name, Loan Number. On entering any of the fields, I should get matching search results. If there are multiple search results indicate to the user to narrow down the search. Display only Loan detail screen only if matching results are found. No wild card search is required. |
| US\_03 | As an admin user I should be able to originate/add new loan  Acceptance criteria:  New Loan add screen must have User’s First Name, Last Name, Loan Number, and Property Address. All fields are required. If validation is successful, then add the record if not show validation message. |
| US\_04 | As an admin user I should be able to modify a loan (loan details – loan amount, loan type, loan term)  Acceptance criteria:  After a successful search from the Loan Search screen, display loan detail screen. Loan detail screen must have Update button to update any modification to the loan details. Fields will be displayed in text box. |
| US\_05 | As a developer I should be able to build and deploy the application using Azure DevOps pipeline to PCF.  Acceptance criteria:  Application should be successfully built, deployed and running. |

# Expected Deliverables

Capture the deliverables to be submitted by associate after completing the development.

* Readme document on the complete application
  + Setup of the application
  + How to run the application
  + Screen snap shots
* Reports
  + Code Quality Reports
* Automation Scripts
  + Script file for Continuous Build, Continuous Integration & Continuous Deployment

# Milestone and duration

|  |  |
| --- | --- |
| Milestone | Deliverables |
| Milestone -1 | UI Components using Angular |
| Milestone -2 | Spring Boot services. |
| Milestone -3 | Application deployment to PCF using Azure DevOps pipeline |

# Implementation Notes

|  |  |
| --- | --- |
| Frontend - Angular | Milestone – 1   * Implement UI functionalities using Angular, Typescript, HTML5 and CSS3 * Unit Test the UI using any test framework * Ensure the code quality using any tool like Codelyzer\TSLint |
| Backend -Java | Milestone-2   * Use Domain Driven Design * Use Microservice Architecture(with patterns like service registry, discovery, circuit breaker) * Ensure 12 factor app methodology is followed * Use Spring Boot- Rest APIs to develop the services * Use Core Java 8 features wherever applicable. * Use Spring Data JPA to work with database (MySQL) * Use Maven to build the application * Use OpenAPI to invoke REST APIs * User authorization to allow/disallow CRUD operations * Any error message or exception should be logged (and help in refactor) * Unit test the application * All implementation should publish Code Quality Metrics using SonarCloud/SonarQube * Technical Debt – lower-the-better * Code Smell – lower-the-better * Cyclomatic Complexity - lower-the-better * Code Coverage – higher-the-better * Secure coding practices * Follow coding standards * Do performance test of the application using JMeter |
| Cloud – PCF and Azure | Milestone-3   * Develop pipeline using Azure DevOps * Azure Database for MySQL to be used to store and retrieve data * Deploy the application to PCF |