

Loan Eligibility and Approval System: Design and Analysis Report

1. Project Overview

- **Objective:** Build a system to manage loan applications and determine loan eligibility using predefined rules with Drools.
 - **Tech Stack:** Angular (Frontend), Spring Boot (Backend), MySQL (Database), Drools (Rule Engine).
 - **Addons:** Bootstrap (CSS Framework) , Swagger(For testing REST API for Backend), BASE64 (For Encoding & Decoding / For Masking the Query Parameters).
-

2. What is Drools?

- Drools is a business rule management system (BRMS) and rule engine.
- It separates business rules from application code, making the logic more maintainable and reusable.

Advantages of Using Drools in the Project:

- Rules can be updated in .drl files without modifying application code.
 - Business rules are independent of specific application logic.
 - Supports complex rule evaluation efficiently.
-

3. High-Level Design

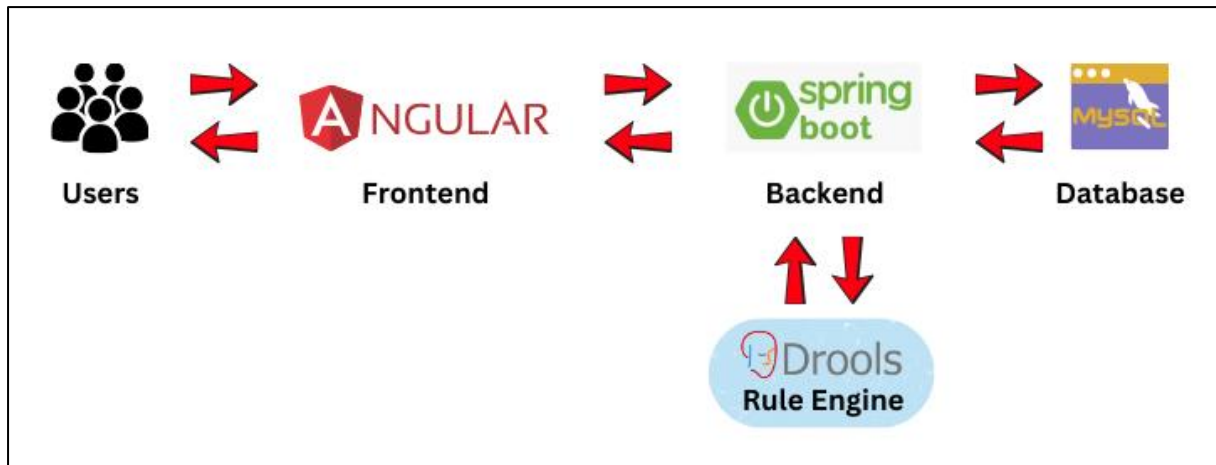
Architecture Diagram:

- **Frontend:** Angular application for managing users and loan applications (UI).
- **Backend:** REST APIs built with Spring Boot for CRUD operations and rule evaluation.
- **Database:** MySQL to store user and loan data.
- **Drools:** Rule engine integrated with Spring Boot to evaluate business logic for loan eligibility.

High-Level Flow:

- User interacts with Angular frontend → Sends requests to Spring Boot API.
- API interacts with the database for CRUD operations.
- Drools evaluates business rules for loan eligibility when a loan application is submitted.

High-Level Diagram



4. Low-Level Design

1. Frontend Components:

- **User Management:** Form and table for adding, updating, and viewing users (CRUD operations on User).
- **Loan Management:** Form to apply for loans and a table to list loan applications with their status (Pending / Eligible for Loan / Rejected).

2. Backend Components:

- **Entities:** User, Loan.
- **Repositories:** Interfaces for database operations (JPA).
- **Services:**
 - **LoanService:** Processes loan applications using Drools.
 - **UserService:** CRUD operations on users.
- **Controllers:**
 - **UserController:** Handles API request.
 - **LoanController:** Handles loan application submissions and status updates.

3. Drools Integration:

- Rules stored in .drl files.
- KieSession processes rules at runtime. (KieSession is **the most common way to interact with the engine**. A KieSession allows the application to establish an iterative conversation with the engine, where the state of the session is kept across invocations. The reasoning process may be triggered multiple times for the same set of data.)
- Rules determine the status of loan applications (APPROVED/REJECTED).
- LoanService ---> KieSession ---> Rules (.drl files) ---> LoanApplication Status

4. Database Schema:

- **User Table:**
 - id, name, age, monthlyIncome, creditScore, existingLoans.
- **LoanApplication Table:**
 - id, userId, loanAmount, loanType, Status, StatusReason.

Low-Level Diagram

