**HOME LOAN APPLICATION**

**INTRODUCTION-**

Home Loan Application : taking Home Loan in online.

In this application we can Apply loan by sitting in the home itself and we can Update loan, Delete loan, Retrieve all loan Application and Retrieve all loan Application by Id.

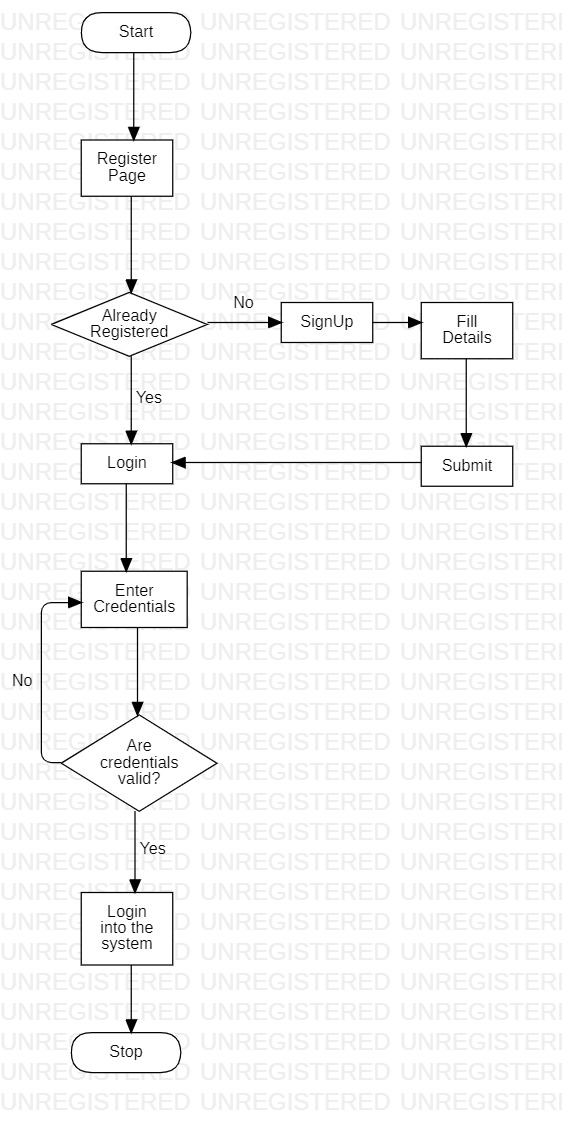
After applying Loan ticket is raised, those pending tickets are verified by the Land Verification Officer. After the approval of Land Verification Officer, a ticket is raised for Financial Officer, he approve loan and send it to Admin.

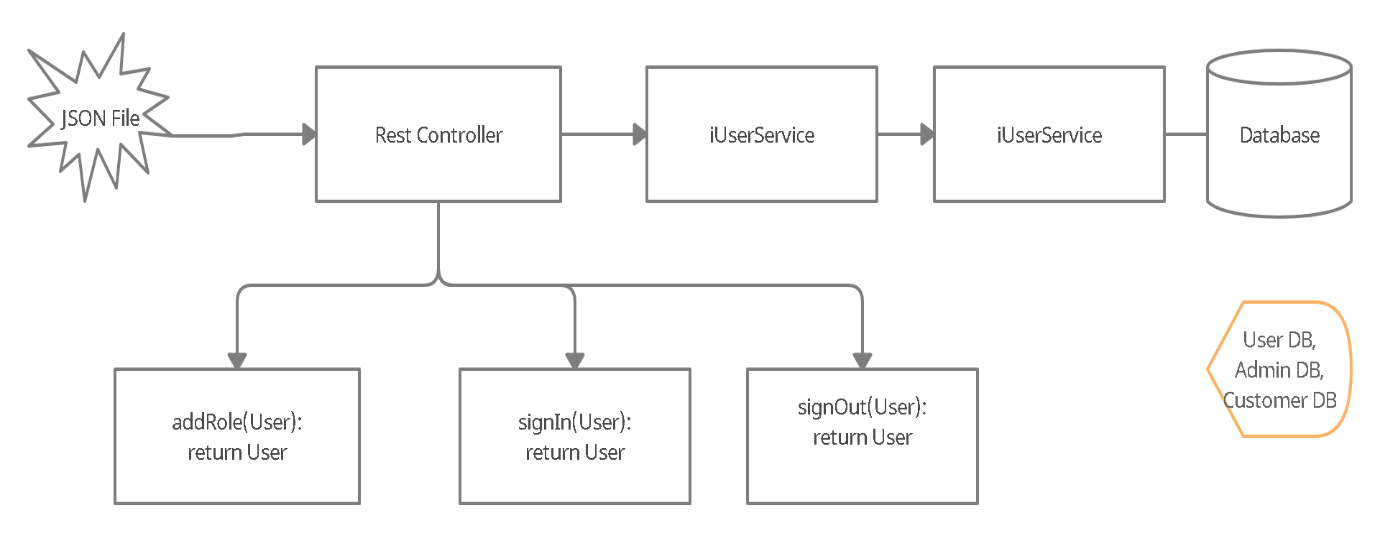
Admin view pending approvals and decide whether to accept Loan or Reject Loan and if possible he can create new schemes.

**REQUIREMENT ANALYSIS:**

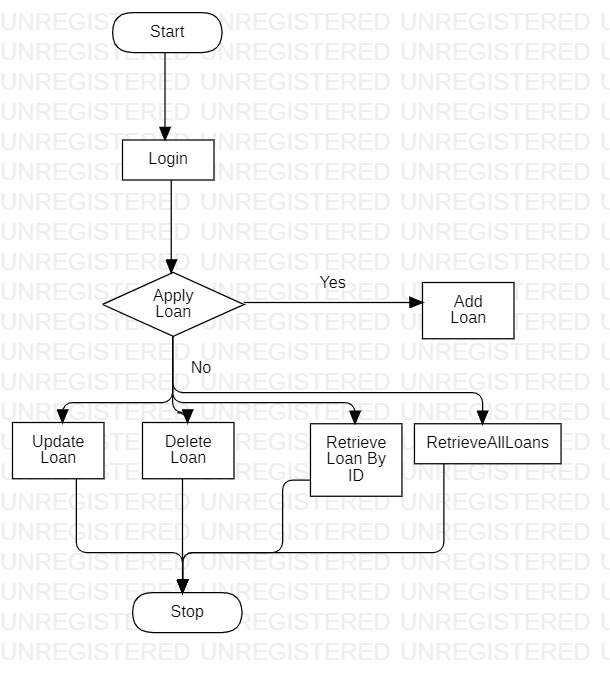
* Basically , the customer should register in application, after that the user can login to his account.
* Every module has their entity class which will be used for creating their respective tables in database using spring and jpa annotations. All modules have interdependency so each module’s entity class will have a field or a collection element to store the records of other modules.
* In **Customer Module**, we will provide all details of the customer like user id, name, mobile number, email id, date of birth, gender, nationality, Aadhar number, Pan number.
* In **Login Module**, we will provide user details like user id, password.
* In **the Loan Application Module**, we will provide details for how we apply loan , application id, loan applied amount, loan approved amount, status.
* In **Loan Tracker Module** , we will provide status like approved, pending, land verification approval, finance verification approval, admin approval, rejected.
* In **EMI Calculation Module**, we will provide EMI id, loan amount, due date, rate of Interest, Tenure, Interest Amount, Loan Agreement Id.
* Wherever there is chance of an error/exception throw it in the service implementation class using the exception module provided in the zip.

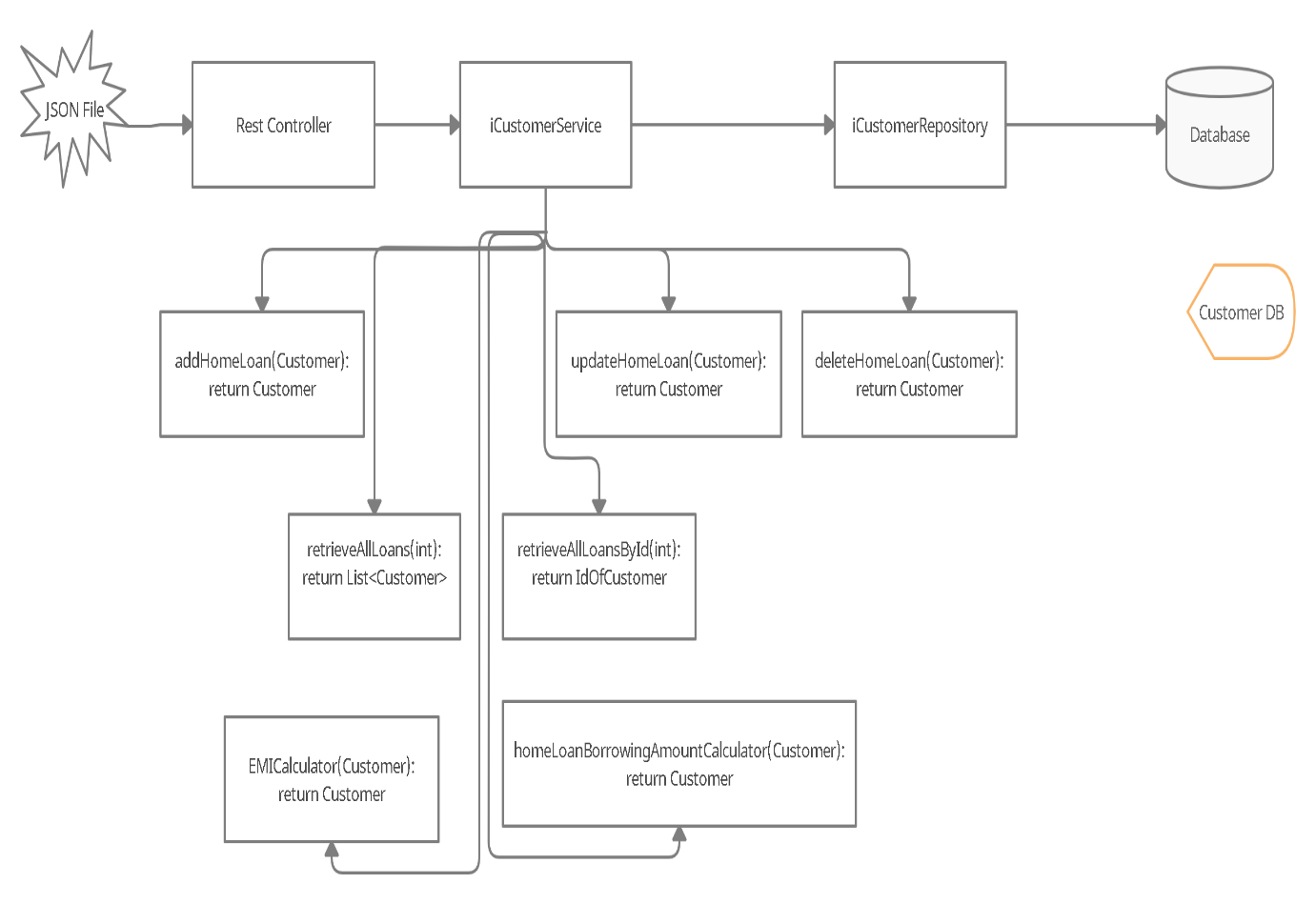
**Login Page:**



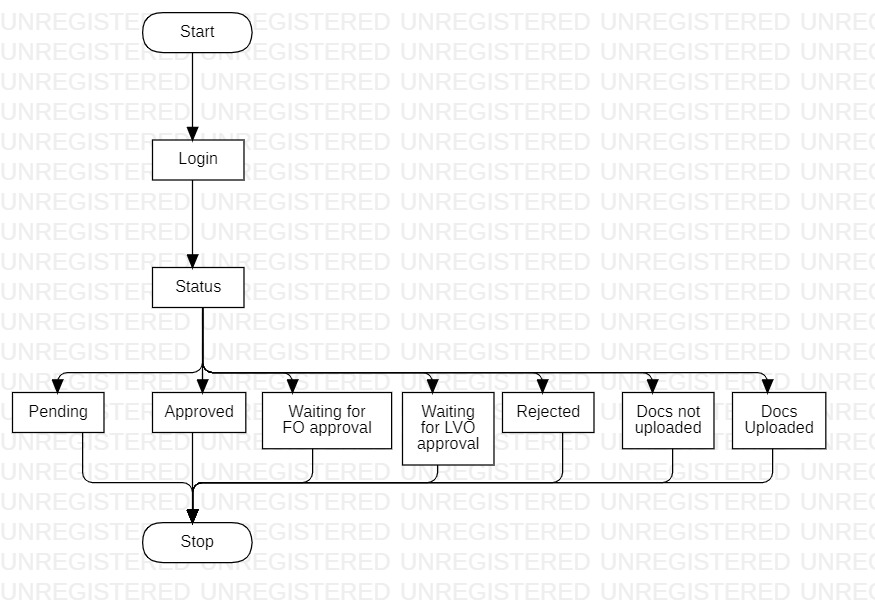


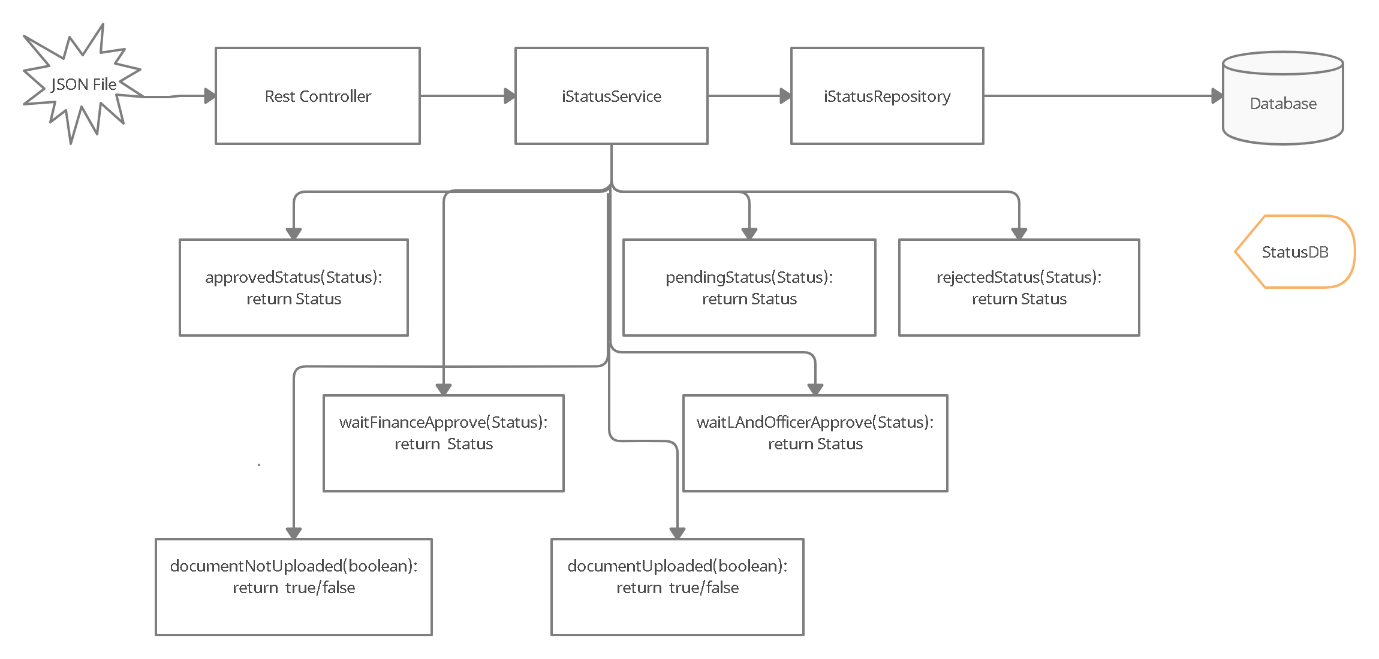
**Apply Loan:**

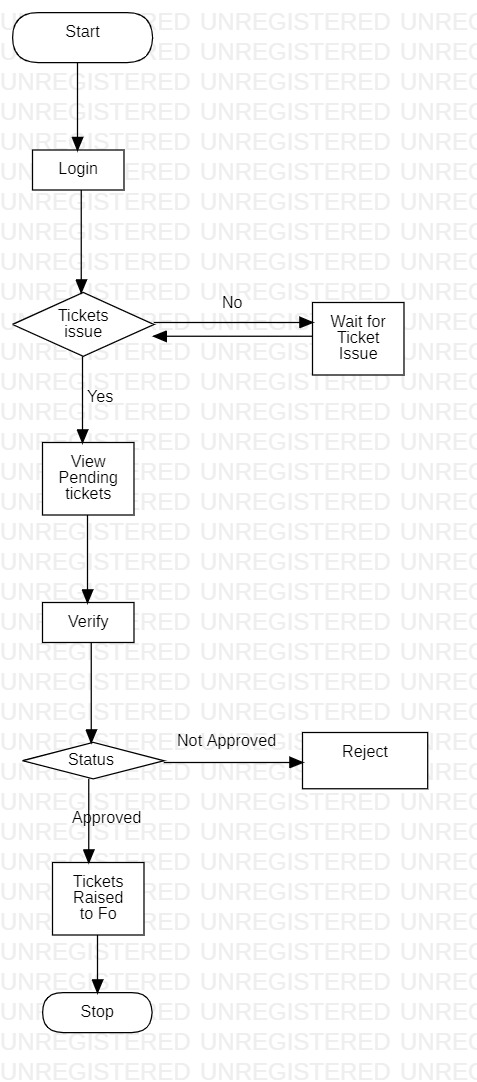


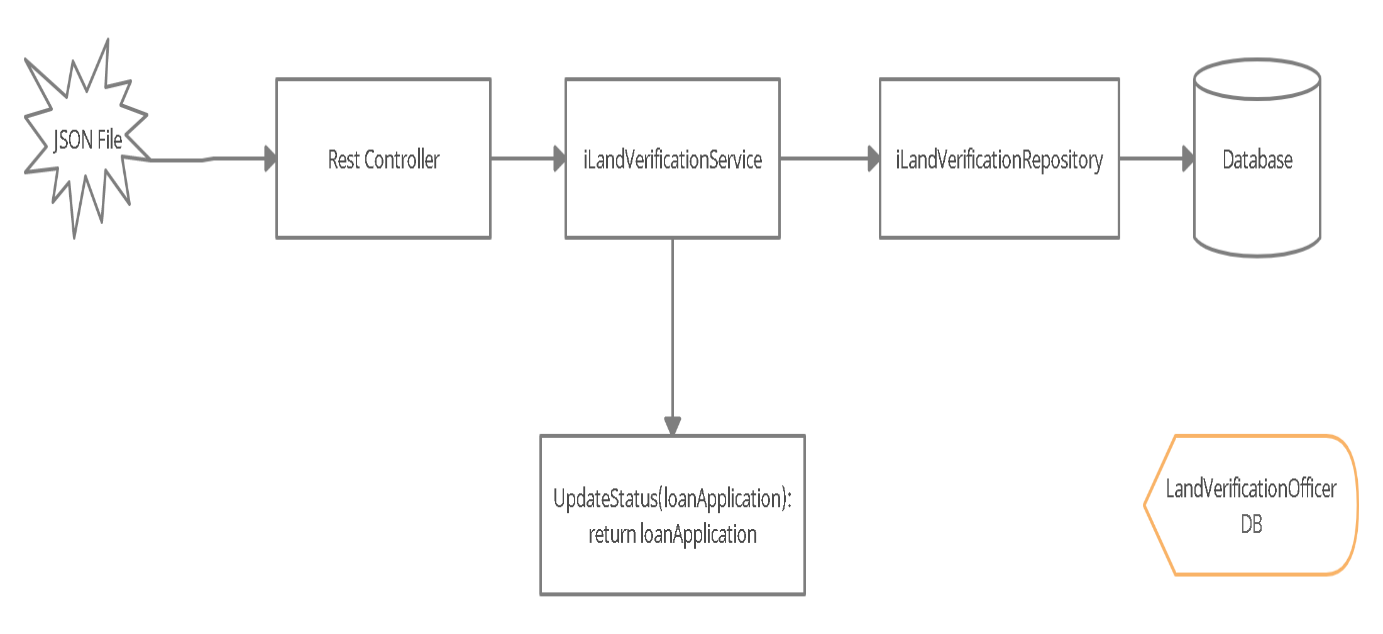


**Track Status:**

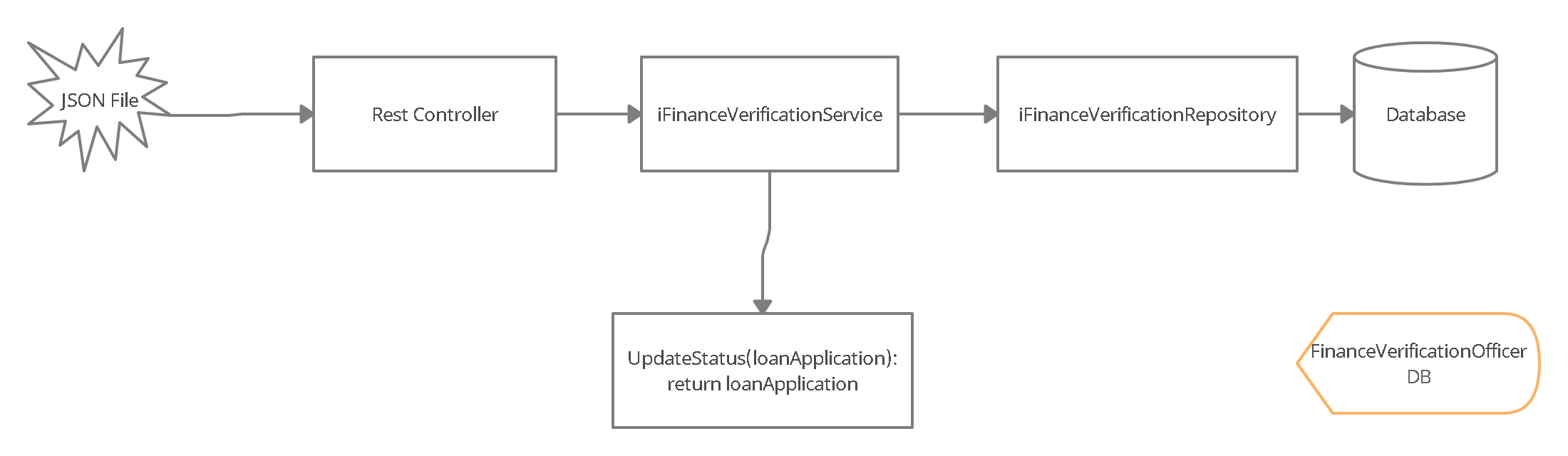


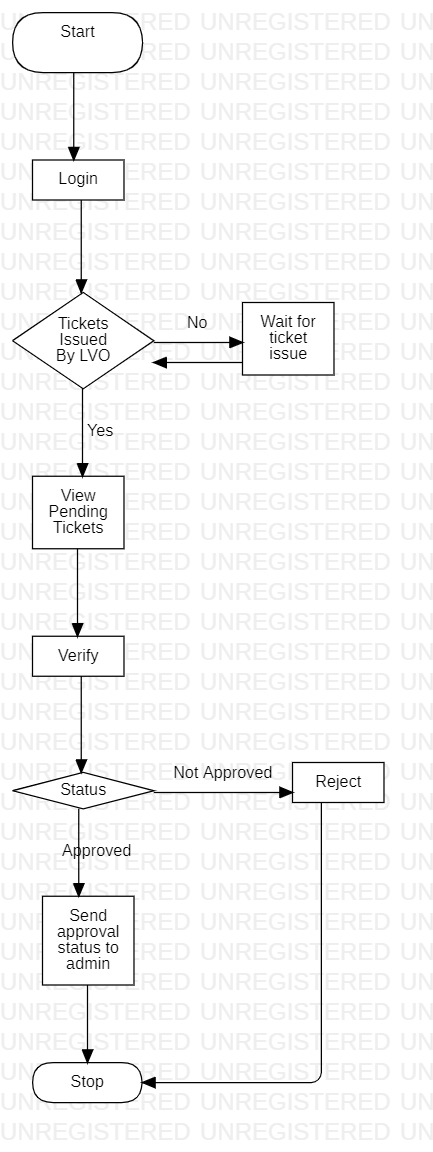


**View and Verification of Pending Tickets(Land Verfication Officer):**

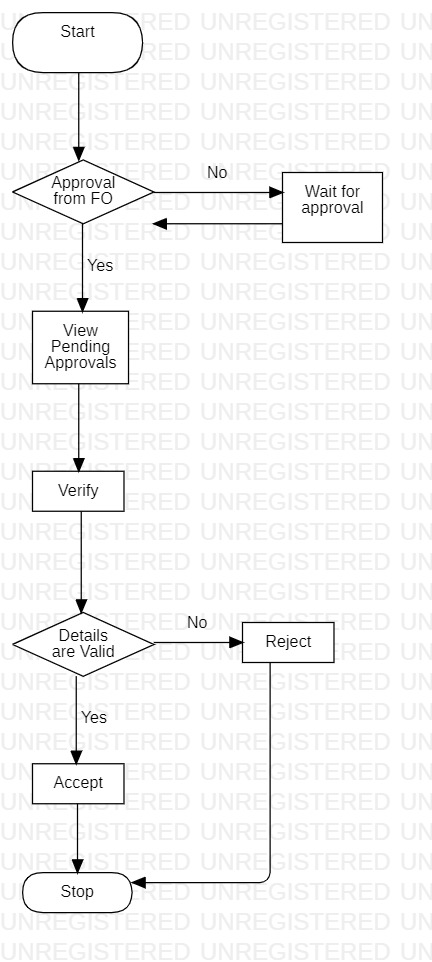


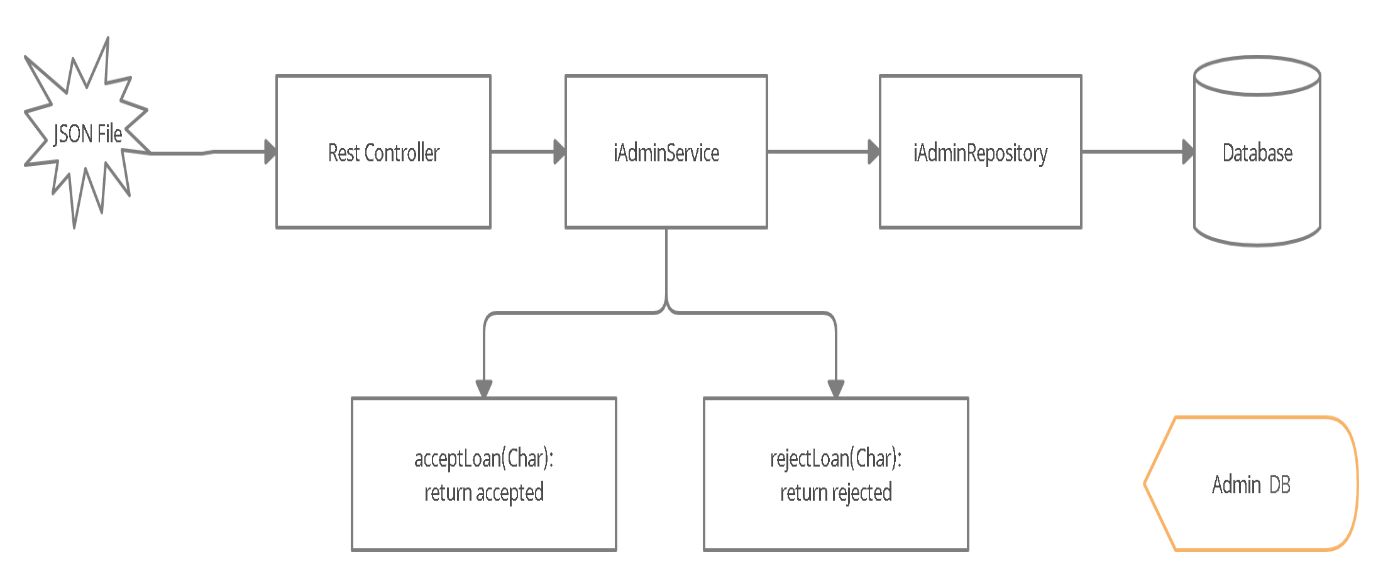
**Verification and Approval of Loan Application(Finanace Officer):**





**Approval for the Loan(Admin):**





**Class Design: POJO/Entity Classes**

Graphical user interface, diagram

Description automatically generated

**SOFTWARE REQUIREMENTS-**

* MySQL, Oracle
* Eclipse(J2EE)
* Postman
* DevOps
* GitHub

**USE CASE DIAGRAM**

Diagram

Description automatically generated

**Flow Chart:**

