

**(LEND.IN LOAN AUTOMATION SYSTEM USING
FLOWABLE)**

Project report submitted in partial fulfillment of the requirement for
the degree of Bachelor of Technology
in
Computer Science and Engineering
By

Vineet Aggarwal (171251)

Under the supervision of (Mr. Manjunath C.H)



Department of Computer Science & Engineering and Information
Technology

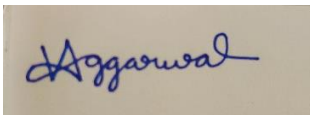
**Jaypee University of Information Technology Waknaghat, Solan
173234, Himachal Pradesh**

Certificate

Candidate's Declaration

I decree hereby that the work presented in this report entitled “**LEND.IN LOAN AUTOMATION SYSTEM USING FLOWABLE**” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is a real record of our own work carried out over a period from February 2021 to May 2021 under the supervision of Mr. Manjunath C.H (Lead Engineer, Kuliza Technologies).

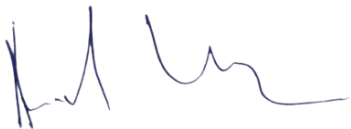
The matter embodied in the report has not been submitted for the award of any other degree or diploma.



(Student Signature)

Vineet Agarwal, 171251

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



(Project Guide Signature)

Dr. Amol Vasudeva

Assistant Professor (Senior Grade)

Department of Computer Science

and Engineering & Information Technology

Dated: 17/06/21

(Supervisor)

Mr. Manjunath C.H

Lead Engineer, Kuliza

ACKNOWLEDGEMENT

With great pleasure I avail this unique opportunity to express my profound sense of gratitude and indebtedness to **Mr. Manjunath C.H, Principal Engineer at Kuliza Technologies**, for this efficacious advise, perpetual and prolific encouragement and constructive criticism during the course of the project. I would also like to express my sincere gratitude to **Mr. Gaurav Jadhav** and **Mr. Mandip Gothadiya, software developers at Kuliza Technologies**, and other developers of this project group for their help rendered in completion of this project work. Additionally, I would like to thank **Dr. Amol Vasudeva (Assistant Professor, JUIT)** for consistently guiding me during college on various projects.

I am also obliged to JUIT College and the staff members for the useful knowledge they provide in their respective fields during the time of my assignment. I am thankful for their cooperation.

Finally, I thank the Almighty for his relentless support, our parents and our classmates, without whom this task would not have been possible.

TABLE OF CONTENTS

Chapter-1 INTRODUCTION	1 to 12
1.1 Introduction	1
1.2 Problem Statement	3
1.3 Objectives	5
1.4 Empirical Study	6
Chapter-2 LITERATURE SURVEY	7
2.1 Summary of papers studied	7
2.2 Table-1 Summary	8
2.3 Technologies Used	9
Chapter-3 PROPOSED METHODOLOGY	10
3.1 Description of the project	10
3.2 Requirement Analysis	13
3.3 Functional Requirements	15
3.4 Non-Functional Requirements	16
3.5 Solution Approach	17
Chapter-4 IMPLEMENTATION DETAILS	37
4.1 Use Cases	37
4.2 Dependent Fields	38
4.3 Types of Actions	39
4.4 Unit Testing	
Chapter-5 CONCLUSION	44

5.1 Conclusion	44
5.2 Future Scope and Contribution	44
5.3 Future Work	45
REFERENCES	45

LIST OF ABBREVIATIONS

- 1.) **IB**- Integration Broker
- 2.) **MSME** - Micro, Small & Medium Enterprises
- 3.) **MUDRA** - Micro Units Development and Refinance Agency
- 4.) **ASYNC** - Asynchronous
- 5.) **JPA** - Java Persistence API
- 6.) **HTTP** - Hypertext Transfer Protocol
- 7.) **SHA** - Secure Hash Algorithm
- 8.) **NSDL** - National Securities Depository Limited
- 9.) **MYINFO** - My Information
- 10.) **WIP** - Work in progress
- 11.) **API** - Application program interface
- 12.) **UI** - User Interface
- 13.) **URL** - Uniform Resource Locator
- 14.) **Fintech** - Financial Technology
- 15.) **JSON** - JavaScript Object Notation
- 16.) **OOPS** - Object Oriented Programming Language and Systems

17.) IDE - Integrated Development Environment

18.) NPA - Non Performing Assets

19.) LOS – Loan Origination System

20.) LMS- Loan Management System

LIST OF FIGURES

Fig 1) Lend.in

Fig 2) Spring Framework

Fig 3) Flowable Engine

Fig 4) Flow of keycloak

Fig 5) Postman interface

Fig 6) Most popular spring projects

Fig 7) Unit Testing using Mockito

Fig 8) Process of loan application

Fig 9) Inside cycle of loan application

Fig 10) Login page - portals

Fig 11) Role selection page

Fig 12) Selection page

Fig 13) Detailed description of user

Fig 14) Upload document window

Fig 15) Comment section

Fig 16) Importing API

Fig 17) Provide request body

Fig 18) Provide authorisation

Fig 19) Provide headers

Fig 20) Request methods

Fig 21) Role map of user

Fig 22) Server client model

Fig 23) Loan Approval Flow Diagram

ABSTRACT

Lending or debt is one of the most widely used financial instruments to raise capital, whether it is a multi-billion-dollar conglomerate or a recently established MSME. For almost two decades there hasn't been much innovation in B2C lending. We've been following the same, obsolete, labor intensive framework for providing loans to retail as well as institutional clients. In recent times, the retail clients for loans have exponentially increased which signifies an increasing level of trust in the banking system, but this has also proportionately increased the load on the banking system as the number of applications for loans are at an all-time high, what we need is to automate this process as whole, while keeping a strict client evaluation system as well, because banks cannot handle major defaulters any longer. Kuliza technologies has developed a low code solution to automate the complete lending cycle. From customer registering themselves, providing their KYC details and the type of loan they wish to procure to, evaluations of the value of customers' collateral such as their property, to determine their eligibility for loans, Lend.In also provides a user side portal, using which a bank's employees can review all the arriving applications and take the final decision on whether to grant them a line of credit or not. The system is based on Flowable, which is an open-source business automation engine.

Chapter-1 INTRODUCTION

1.1 Introduction

Kuliza Technologies :

Kuliza is a service-based firm that has worked on over 120 projects for startups as well as major large organisations with its technical innovation, providing digital knowledge solutions for financial sector organisations since 2006.

Kuliza's self-made endeavour, Lend. In, is also the company's flagship lending enterprise. It has concentrated in the development of enterprise software for the past 12 years, and Kuliza's hands-on experience in experimental design methodologies, open source computing, and artificial intelligence enables financial companies to fuel their digital businesses.

This solution aids enterprises in the design and development of end-to-end smart digital lending systems, as well as their transformation suite.

Kuliza has also been named by Deloitte and Nasscom, and has won numerous important accolades for NBFCs, including EY's Technoviti Award and Banking Frontiers, Digital Transformation Partner, and Best Lending Program of the Year.



Figure 1 Lend.In

Lend.In:

Lend.in is a product produced by Kuliza Technologies for financial institutions. With this product, Kuliza is attempting to provide a comprehensive digital platform for loan administration for banks and other Fintech firms. It can be extremely useful in many Loan systems like :-

1. Personal Loan
2. Auto/ two – four-wheeler loan
3. Home Loan
4. Gold Loan
5. Loan against securities
6. MSME Loans, including MUDRA loans
7. Education Loans
8. Top up Loans
9. Pension Loans

Portals

Imagine a web-based platform that allows business users to perform business actions depending on particular risks, criteria, and circumstances. This is the most straightforward approach to grasp portals.

Various modules in the product lending suite make use of portals on occasion dependent on their usage. Once the user has successfully submitted his loan application for approval, a back-office, which is essentially Portals, enters the scene. Portals is a high-level DEM layer that controls the back-end of the loan process, the client-side of the LMS, and the full Collections and Collaterals process.

In terms of the loan application process, numerous qualitative and quantitative factors of the client are taken into account before final confirmation and approval.

Quantitative factors may be factored in and managed by automation of an objective system during a Straight-Through-Process, but when qualitative factors are included, human input is required at various phases of the evaluation. Only competent human interaction can assess and implement risk appetite, as well as a subjective assessment of an application.

These human roles which are the authority of these analyses are broadly classified under the following segments:

- 1.Risk and Compliance Regulation Executive
- 2.Underwriters and Appraisers
- 3.Operations Executives

Lend.In designs the system in such a manner that all these roles can be understood and controlled

under a single interface, with control over the:

- 1.Read/Write/View Accesses
- 2.Overview of relevant information
- 3.Flow of application
- 4.Ability to customize inbox view as per role/authority
- 5.Integration with 3rd party roles

Across the industry, the salient features that dominate the design of a gateway portal for all demands of credit appraisers, revolve around the following frameworks:

- 1.Regulation of assessment processes across geographies
- 2.Maintenance of various loan products, roles and users
- 3.Agility to authorize changes in practices pivoted around risk and compliances regulations
- 4.Capacity to integrate with 3rd party apps, portals and/or APIs for faster processing of information
- 5.Inherent smart features to achieve real-time MIS and analysis reports
- 6.Capability to support the accelerated implementation of Go-To-Market strategies

A complete product that encompasses all use cases of the above scenarios would provide lending organisations an advantage in terms of efficiency, accuracy, and cost reduction.

Lending Portal is a single portal through which all loan applications are routed for end-to-end assessment and underwriting after the customer/applicant has successfully completed the loan application. The application is subsequently forwarded to this portal after completing initial checks and objective sanctions included in the customer's experience.

From here, the second tier of stakeholders, viz. risk and credit analysts, underwriters, can subjectively appraise the loan application according to the risk appetite and demands of macro business aspects of financial institutions.

The salient features of Lending Portal can be broadly classified as:

- 1.Design Characteristics
- 2.Functional Features
- 3.Interfacing Capabilities

Spring Boot Framework:

Spring Boot is an open source and application system that serves as a container for the inversion of control on the Java platform. The main aspects of any Java programme will be implemented. There are several extensions available for building web applications on top of the Java Enterprise Edition (EE) platform. The main benefit is that it does not follow any particular programming paradigm. Spring is utilised by many large corporations and has gained popularity in the Java community as a replacement for the JavaBeans Enterprise (EJB) paradigm.

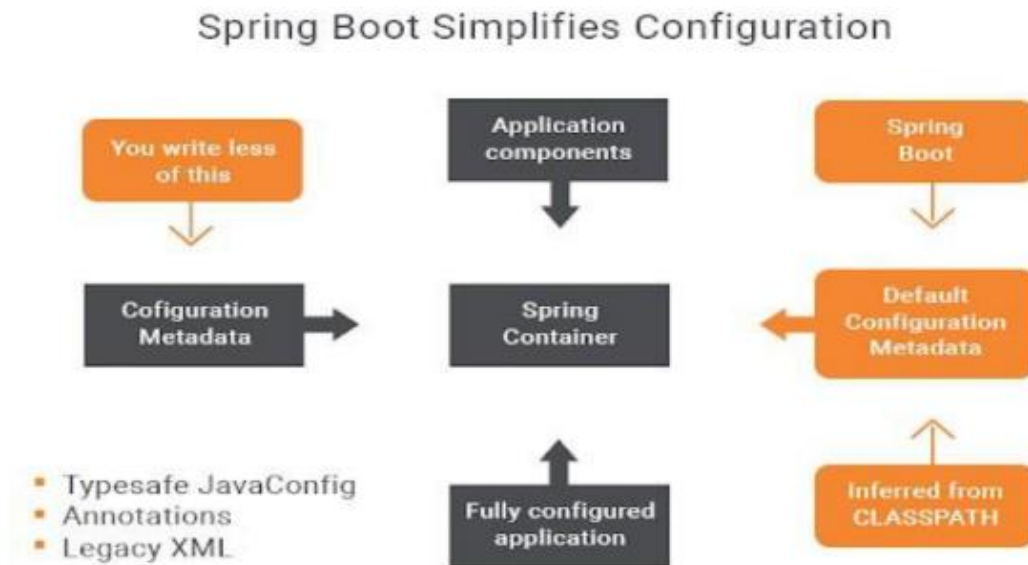


Figure 2 Spring Framework

It is an open-source workflow engine created in Java 8 that enables various businesses to execute BPMN-defined processes. There are multiple methods for creating a flowable, and some of them resemble the ones in RxJava (first version) in Observable. RxJava was designed from the ground up by Oracle employees, and it includes a slew of new features that not only provide flowables a whole new design, but also fix errors in the previous version and provide a reaction to framework difficulties. Bugs in the previous version of RxJava—There was only one base class for managing back pressure, even for non-back pressure sources, and this caused a lot of confusion.

The second RxJava edition, known as Flowable, made a clear separation between these two services. Back pressure is not recommended by observable sources, therefore this may be a huge benefit for employing it for services that are routinely consumed without control.

Different ways to create Flowable are as mentioned below -

1. Flowable using just() method -

```
Flowable<Integer> integerFlowable = Flowable.just(1, 2, 3, 4);
```

2. Flowable using toFlowable() method -

```
Observable<Integer> integerObservable = Observable.just(1, 2, 3);
```

```
Flowable<Integer> integerFlowable = integerObservable
```

```
.toFlowable(BackpressureStrategy.BUFFER)
```

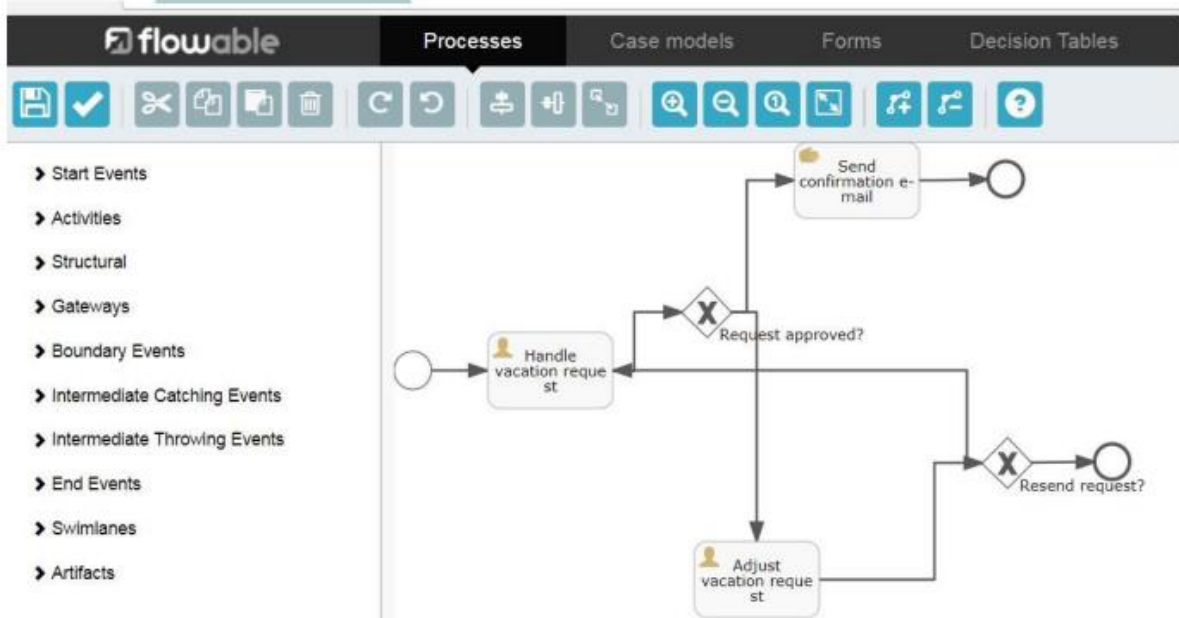


Figure 3 Flowable Engine

KeyCloak:

KeyCloak is an open source software application that supports single sign-on, identity management, and access management for future technologies and services. It was built by the JBoss group in March 2018, and it will now be managed by Red Hat, who will utilise it as the flagship product for their RH-SSO. The main goal of this application is to safeguard our software while also providing the benefits of coding little or no. It may be customised and utilised by enterprises. There's no need to sign in again if you've already signed in to Keycloak servers, and the same goes for signing out. This provides its customers with a comprehensive interface that makes it simple for them to handle all administration difficulties. It works on all platforms, including mobile, web, and iOS.

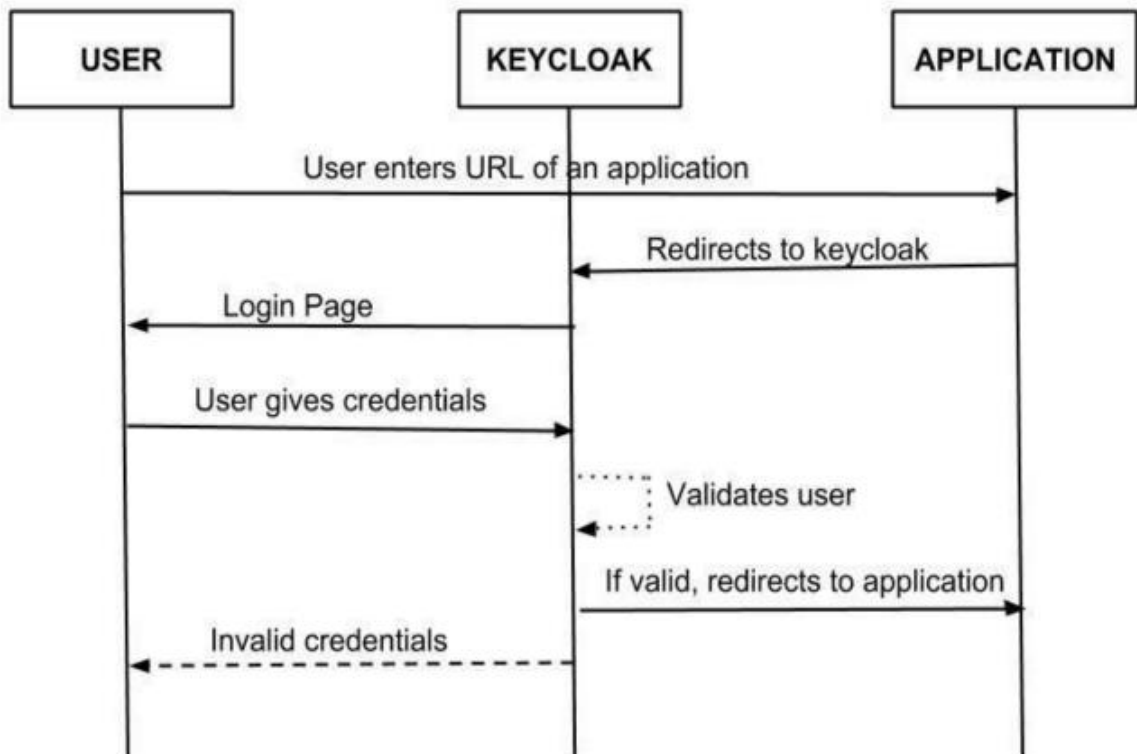


Fig 4. Flow of Keycloak

Credit Engine:

It is a Fintech company that has been operating in Japan since 2006. It operates and concentrates on online lending services, such as LENDY (<https://www.lendy.jp/>), a Japanese internet lending service for small and medium-sized businesses. We employ machine learning

algorithms to assess credit risk, which is something that most banks, particularly major hotshots, aren't interested in. The major focus in the future years will be on developing a B2B product and providing a lending platform for other financial institutions.

Postman:

It is an Api client tool which is used to test the APIs. It provides many features like to test a single API against many different environment specific variables.

Link - <https://www.getpostman.com/>

API testing is similar to software testing in that it provides the user with a direct interface to check if the anticipated data matches the actual data received and to assess the dependability, performance, and security. Since APIs lack a graphical user interface, it is difficult for developers to test their APIs on a regular basis because deployment must be done on a server. Postman aids in this process by providing a platform to validate the API's integrity by developing and testing the project on a local server. The challenge of automating API testing is seen as vital. The principal interface of the application logic and GUI tests are difficult to maintain with the short release cycles and rapid changes often employed with Agile software development, hence automating API testing is considered a key effort.

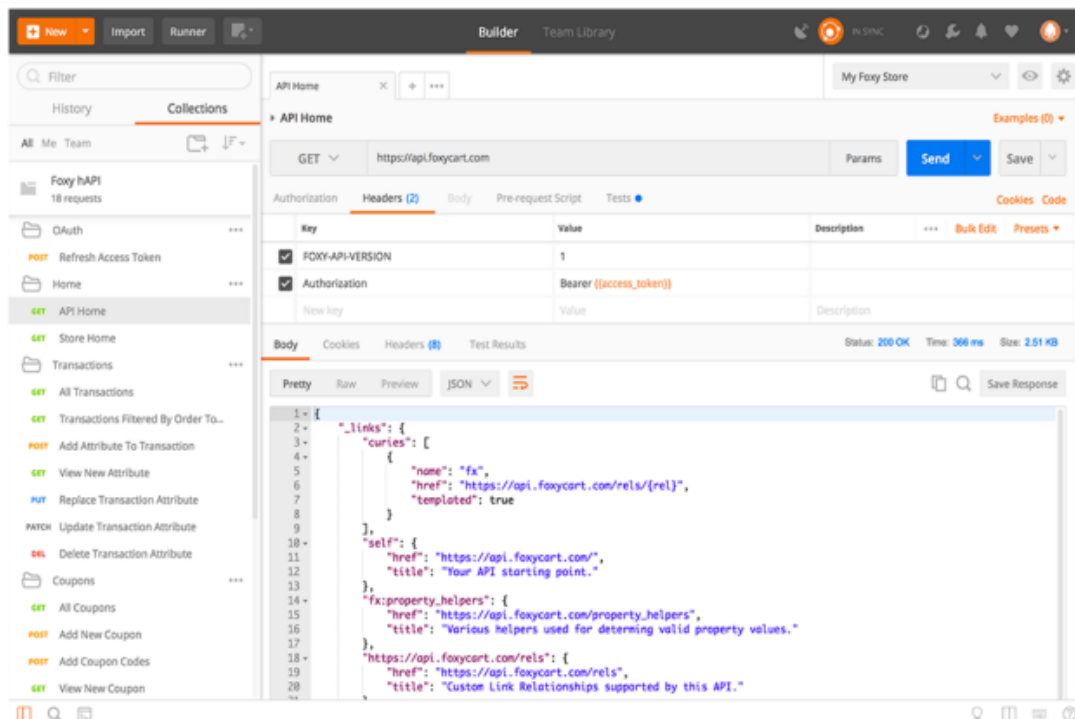


Fig 5: Postman Interface

Elastic Search:

ElasticSearch is a type of search engine that is based on the Lucene library. It allows users to search full-text search engines via an HTTP web interface, as well as JSON documents. It is written in the object-oriented Java programming language and includes OOPS features. It is licenced under a variety of open source licences, the majority of which are Apache Licenses, and it also supports the open-core business model as the Elastic License continues to decline. Many of its official clients are available, including Java,.NET, PHP, Python, Apache, and a variety of other computer languages.

IntelliJ:

IntelliJ is a Java integrated development environment (IDE) for developing applications in the field of computer science. It was created by JetBrains developers and is available under the Apache 2 Licensed community edition licence. Code completion, navigation through the code (which allows switching from one function to another immediately), refracting of the code, and choices to remedy any problems like as spelling problems, notations, and so on are the main aspects of using it. It uses its capacity to analyse the context to aid in code completion.

1.2 Problem Statement

A credit will be overhauled for each consumer, who will have various payback dates, terms and restrictions, and so on. You'll make certain that accounting is in order, and that an advance handling system seeks to integrate particular types of accounting in order to ensure that the business is prepared when reimbursements arrive. From the standpoint of specialised strategy, the new business advance-administration systems are confusing and vital. This means that the time to market is long, and the architecture's execution isn't flexible enough to meet the business sector's changing demands.

There are also many requests from Fintech firms for a completely new and innovative adaptable platform to meet their needs, because all organisations operate differently, and supplying them with a comfort zone that meets their needs is quite difficult, thus Portals come into play to assist address this problem.

Many banks, including PSBs and private banks, are fighting the NPA battle, and as a result of

NPA issues in corporate loans, they are now focusing only on retail loans. As a result, under PCA, banks can only make Retail Loans. As a result, there is fierce rivalry among lenders for Retail Loans. Pricing has lost its appeal, whereas delivery has increased in importance in terms of speed and convenience. Additionally, each lender is obligated to cut expenses by automating processes. Most PSBs and older Private Banks still lack Digital Lending systems, with the exception of SBI and a few others. Those banks who have previously created digital platforms are still looking for acceptable solutions to upgrade their existing platforms in order to stay ahead of the competition. Furthermore, the government's focus on MSME loans, as well as the requirement that all banks approve MSME loans up to Rs.1 crore in 59 minutes, would force all banks to update their lending systems. NBFCs and HFCs, which were formerly quite active in increasing their loan book, are now facing serious liquidity issues.

The pace of current expenditures has slowed. It will very certainly persist for some time. With the exception of a few NBFCs and HFCs, most of these companies are likely to reduce their IT investment. Others, on the other hand, may use this time to modernise their lending systems or aggressively expand their market share. These businesses want to modernise their lending systems. RRBs (over 50) are also aiming to expand their lending channels, which are mostly untapped at the moment.

1.3 Objectives

The primary goal of this item's structure is to create a lightweight, simple-to-understand, adaptable, and capable system for advanced administration. The goal is to establish a system that allows advancements to be prepared and managed as early as feasible, and to assist the organisation in improving their dashboard in accordance with regulations. The entire process hinges on the creation of a framework that can be altered and updated as requirements change. This results in a work process that should be overseen by a representative or a firm.

To establish a good relationship with the clients, they should be provided a basic grasp of loaning. At that point, they are pushed to use the organisation for future credit needs. The point of view is to alter the UI using Ant-structure to meet the configurators because it is an arrangement-based object.

1.4 Empirical Study

Many developers in kuliza have come up with this strategy of employing a self-made library in ReactJS called as the Rendering Engine that can fulfil the demands of the configuration and backend teams by building REST APIs to link it with the database after doing a lot of study on the loan framework and its capabilities and how it works. For the conceptualization and manufacturing of this bespoke product, new and cutting-edge technology is applied. It has been ranked as the most popular and resourceful search engine in the last few years.

The major goal and aim of the Portal module is to give its users with a fully functional product that allows them to create customised dashboards that follow the workflow defined on the flowable. It gathers data from the backend via the remaining APIs and only considers accurate data, and any third-party APIs and frameworks may gather and utilise this data as well. A single API is responsible for all third-party and internal API requests, and it also calls them (Other APIs). This module also includes a generic authentication layer function that may be reused. Other systems on the market do not provide the ability to customise the dashboard, but they are stable, whereas Lend.in is still in the development phase. Validations and upgrades of information submitted when filling out an application form on portals were also completed.

Chapter-2 LITERATURE SURVEY

After analysing the whole structure and requirement needed for the development of the product, we consumed a few papers of IEEE, Springer etc. and other resources available on the internet and following is the literature survey for it-

2.1 Summary of Papers Studies;

The structure diagram done helped the clarification of the problem to be enumerated and refined. This was also helpful in identifying the key opportunities that should be pursued to organize a down to earth strategy that should be found, and the necessity for which it is appropriate to do the errands. With the purposes of recognizing the dynamic nature of the problem being monitored and dreaming about a feasible solution, the articles were discussed and referenced. The conclusions drawn after detailed study of all the papers are written below:-

1. “After 2000 there is a lot of development in the field of computer science and the first revolution that came is in the field of Web Development, it became an hour of the need to develop such websites that can be used in any conditions and can withhold any amount of traffic, precision is one of the main keys to maintaining such platform. Making J2EEconsistent programming without using Enterprise Java Beans (EJB) was essential, and the main focus. There are several options available, one of which is Spring Structure, one of the most relevant and reliable, which is kind of a much less code framework and less complicated than EJB.”
2. “The main motive is to develop an open source framework that can help bridge the gap between existing benchmark frameworks and other modern microservice frameworks, as well as design and build a whole new platform for microservice design benchmarking. The use of microservice architecture has many drawbacks, as often they lack the speed i.e. performance problem and also the reliance factor and its architectural design matters a lot in the design of our web application.”
3. “MSA which is the Service-situate Architecture takes help from the benefits of programming structure square design, it is just the research and development phase is still in progress.”
4. “Main features of Software Design Principles are studied in detail for the integration and developing the enterprise application.”

5. “Both the architectures microservices and monolithic are totally and both have their advantages and disadvantages, also the development processes are different for both, they both are used for increasing the capability and scalability of the system and improve the performance, efficiency, flexibility, capability and adaptability.”

2.2 Table – Summary papers

<u>S.no</u>	<u>Title</u>	<u>Publication</u>	<u>Problem Definition</u>
<u>1.</u>	Development of open source J2EE Web application in Spring framework	Seventh International Conference on Artificial Intelligence and Distributed Database, IEEE	From this paper we got to know about the basics of spring framework and about its annotations and how to use it in web development.
<u>2.</u>	Implementing Microservices Structure System for Software Engineering and benchmarking	40th International Conference 2018 IEEE on Software Engineering	The main motive is to develop an open source framework which can help to bridge the gap between the existing benchmark frameworks and other modern microservice frameworks and also design and create a whole new platform for benchmarking of microservice design.

<u>3.</u>	Increase in performance of Data Transfer among cross platform Web Applications through XML and JSON files	Mobile Computations and Communications, International Conference - 3rd, IEEE	They were broken down and decided how to calculate the accuracy of the information correctly. The first approach is the recursive calculation to interpret what are the differences between the above written information and also serialising structures were given based on a tree structure of XML and JSON objects.
-----------	---	--	---

<u>4.</u>	Service-Oriented and Microservice Architecture - Difference between Architectural structure	Software Architecture Workshops 2017 IEEE International Conference	MSA which is the Service-situate Architecture takes help from the benefits of programming structure square design, it is just the research and development phase is still in progress.

<u>5.</u>	Microservices : Advantages and disadvantages	Springer publication	Both the architectures microservices and monolithic are totally and both have their advantages and disadvantages, also the development processes are different for both, they both are used for increasing the capability and scalability of the system and improve the performance, efficiency, flexibility, and adaptability.
-----------	--	----------------------	--

2.3 Technologies Used

Elastic Search

Elastic search has emerged as the most effective technique for searching a wide range of documents, thanks to its scalability, speed, and multi-latency capabilities. This allows for sharding and may be distributed across several sources without causing copy artefacts. This does not cause any issues because it is done in real time and all setup, such as rebalancing and routing, is done automatically. It works in the same way as Trie, and the same data is saved on the same index without any copies or replicas. These are set as soon as one shard is allocated to the index, and they may be changed by adding or deleting nodes. . It's a log processing engine with data collecting called Lagstash, and a visualisation platform called Kibana. Beats refers to all of the light understanding. Elastic Stack is a fully integrated solution. Javascript Object Notation (JSON) and APIs are used to access all of its functionalities. Other features include faceting and percolating, which are used to tell the user how well the papers fit

together. The gateway function is another frequent function that is used to keep indexing for a long time. When we get the credentials from the server crash, we may use this capability to restore the index. GET request can be accessed in no time i.e in real time and it works best with NoSQL database.

Credit Engine:

When a firm or a customer is in the process of extending a loan, they confront a number of issues. To disregard the significant credit risk, it also requires time to produce papers and then confirm the legitimacy of such papers. Money-related foundations' assessments may not always represent current company situations because they are typically based on financial summaries from a long time ago. These kind of issues are remedied with the use of technology via Credit Engine. Data is gathered in the form of many forms of information, such as POS registers and payment services. Only thorough information is provided on it, and no paperwork is required of any commercial organisation.

Key Cloak:

Main features offered by keycloak are as follows -

1. Registration of the user
2. Login platform for social platform
3. Feature of single sign in and single sign out
4. Authentication providing 2 layers
5. Integration for LDAP server
6. Broker of kerberos
7. Change of skin, by providing customisation with multi latency Main Components

which can be integrated with KeyCloak are -

1. Server of KeyCloak
2. Application adapter

Java:

Java is a popular programming language that emphasises class-based architecture. the position of things Oracle intends to have as few potential situations as possible. It is often used for a variety of purposes, including the development of mobile apps such as Android apps,

Web apps, Desktop apps, database management frameworks, game development, and so on.

and so on. It is an open source programming language that is also freely available. Java is also speedy and secure when compared to other programming languages. This is meant to give "application builders the flexibility to compose once and execute anywhere," implying that structured Java code will continue to be used.

Java programmes are often compiled into "bytecode," which allows them to execute on any Java virtual machine (JVM), regardless of the PC's hidden architecture. Java's language structure is similar to that of C and C++, in that it contains as few low-level functions as possible. One of the most important features of Java as a programming language is its automated board memory, often known as garbage specialist. When articles are created, the author decides, and the Java runtime is in charge of recovering memory until the items are no longer needed. If no references to an article exist, the trash racer is eventually able to release the difficult memory to enter. At the moment, anything like a memory opening will happen if a designer's code includes a reference to something that will never be used again, which commonly happens when things that will never be used again are kept in still-used compartments. A "invalid pointer exclusion" is raised in when systems are named after non-existent objects.

Spring Boot

“Spring boot came into existence from late 90s and became popular in the last decade as it supports

the development of microservices. The main reasons why it is used are as follows -

1. Applications build from are production ready and ready to be deployed.
2. Many features are also provided to the users like metrics, health checks, etc.
3. Code automation is there.
4. There is no need for XML configuration which was first needed in older version.
5. Lastly, Configuration which is to be done by the developers in the other frameworks, it is not needed now, all of it is automated. All functions like third party API handling and import of libraries in the module are automated inside this framework.”

There are many projects which comes under this which can be used and they all are starter

projects, including dependencies in application.properties. They are turned out to be very useful for the developers as they do not have to write the code for setup of projects and also in developing microservice applications. Examples of the starter projects are -

1. Spring-boot-starter-web
2. Spring-boot-starter-test
3. Spring-boot-starter-jpa



Fig 6: Most Popular Spring Projects

Mockito Testing Frame Work

Developer side testing is highly vital for any application to obtain better results, and today every reputable business company is doing it. It is not the task of a Quality Assurance engineer; it is solely the responsibility of a Software engineer, and Mockito is one of the most popular frameworks for it. It mocks the object as well as the method that has to be tested. It is built on Java and makes use of annotations. It involves creating false objects that act as though they are calling the real function, and then comparing the actual result to the expected result supplied

by the user. It's a method of testing code coverage to determine how much of our code is covered before releasing it, as well as to determine whether any code is included that isn't helpful. In it, a false object is created with the same information as the real item, and the user compares the outcome to the result supplied.

Mockito unit testing is completed by mocking the conditions for the classes under test, followed by code execution. Approve the result after the code has been performed to see if the taunting class restores a comparable reaction true to form or not.

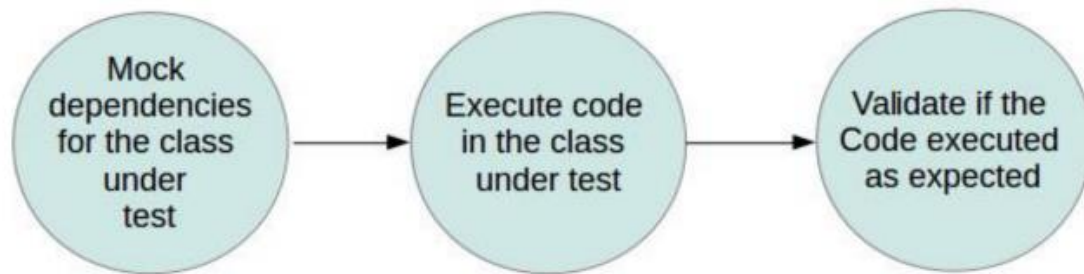


Fig 7: Unit Testing using Mockito

Liquibase

All database modifications are saved as text files (XML, YAML, JSON, or SQL) and are identified by a combination of the "I d" and "author" tags, as well as the file name. A record of all recent modifications is kept in each database that is examined for all database enhancements in order to determine what new modifications should be made. As a result, there is no database version number, but this method allows it to function with a variety of developers and code branches in different contexts.

Liquibase creates the DatabaseChangeLog Table and the DatabaseChangeLogLock Table as soon as you run a changeLog file.

Chapter-3 Proposed Methodology

3.1 Description of the Project

The goal of the project is to bring together all of the distinct credit board frameworks for various forms of advances into a single application. The project is an online interface, which is a redesigned website that combines data from a variety of sources in a logical and formal manner. It adheres to the new Single Page Application development methodology, which makes it easier to access the offered resources. It allows bank employees to access and verify information given by users, as well as assisting consumers in getting their loans authorised promptly and without any processing delays. It also gives users the option of changing the theme and the appearance of the application, which may be customised according to the banks.



Fig 8. Process of Loan Application

3.2 Requirement Analysis

Before a web-based interface can be made available for use, either internally or remotely, a software developer must examine a number of factors. There are various aspects to change from an engineer's perspective, such as structure against stage, strength against early form, and rules vs company aims.

3.3 Functional Requirements

3.3.1 Authentication

- Depending on the role of the user different screens should be shown.
- Login onto the server is mandatory for the user before processing any information further.
 - Tokenization is used for authentication whenever any API call is made from frontend to backend.
 - This process generates a token which is dependent on the type of user logged hitting any API.

3.3.2 Security

- Prevents hackers from injecting their own SQL injections and information.
- Encryption of all information of its users.
- Encryption for password.
- It follows the process of two factor authentication.
- Secure login for users by the HTTPS protocol.
- SSL level protection is also provided.
- Denies any third party web application to load, all things are implemented onto same platform

3.3.3 User experience and interface

- Compatible with different browsers.
- Follows section 508.
- Regular updates of the platform.
- Full review for the user and error handling so user can understand of the issue.
- User friendly interface.
- Beta version of UI checked with many firms.
- Customisable UI for business firms.

3.3.4 Tracking

Tracking is a vital field for the user in the process of having a loan accepted; this feature is also available in it, and it is for both the user and the company authorising the loan; it is one of the elements that sets it apart from its rivals.

3.3.5 Performance

- Caching is available for the user.
- To make the size of application not too heavy, optimisation over the image is also implemented
- Management of Cookie
- To make the application up and running for all time and to protect it from lagging, number of HTTP requests are reduced.
- Taking care of user information by encrypting it.

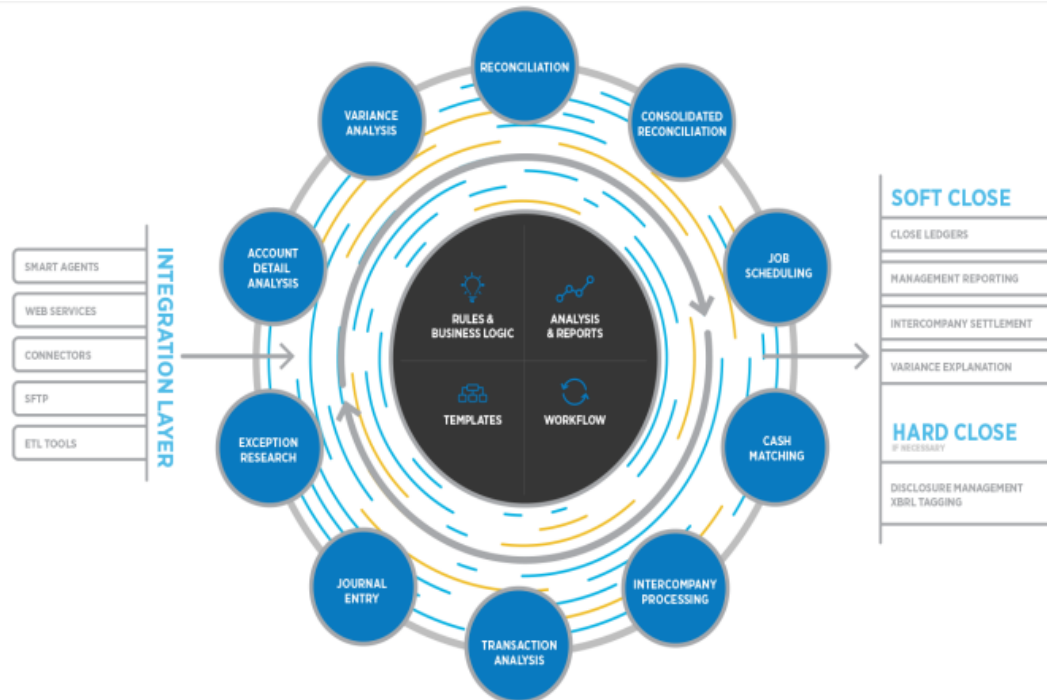


Fig 9. Inside cycle of loan application

3.4 Non-Functional Requirements

3.4.1 Self Service

Many individuals used to have to travel to any official premises and wait a long time for their application to be moved from one department to another, but this application is a self-servicing platform that does not require any interaction with agents or workers; just the validation is done by the company.

3.4.2 Responsiveness

Lend.in provides the best user experience and this is because of fast response and it provides equal priority to every user

3.4.3 Accessibility

It provides maximum accessibility to its users and with this comes with the use of minimum technical knowledge, this is accessible to every age member with ease.

3.5 Solution Approach

Specifications - This project utilises a variety of technologies, the implementation of which takes less time and is simple; further, it ensures that the technology's full potential is utilised. The following technologies are utilised for the frontend and backend:

Backend -

1. Java
2. Spring boot
3. SQL
4. KeyCloak

Frontend -

1. React
2. Javascript
3. Ant Design
4. HTML
5. CSS

Tools used -

1. IntelliJ
2. VScode
3. Postman
4. Eclipse
5. Linux

Portals is a user interface that will be utilised by various LendIn team modules. This is the major page. LOGIN PAGE: This page logs in the user using the credentials already stored in the database. When a user signs in, his or her credentials are transmitted to the server for verification; if they are correct, a token is issued, produced, which is encrypted and contains crucial information about the user as well as additional information gives the user access to additional features that he is capable of. The newly formed token has a unique identifier. This is done to safeguard the service from abuse. It has a set lifetime after which it expires and is no longer useful any type of cyber danger or data breach. Because the entire product is in the sector of financial services, one of the most significant aspects it should have is the ability to preserve user data.

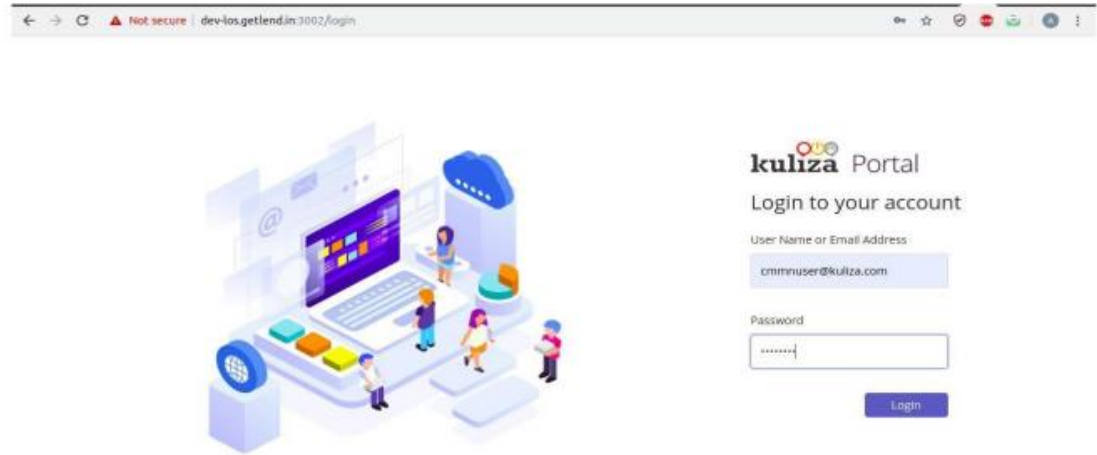


Figure 10 Login Page: Portals

After signing in, the user is brought to a page with several roles; the product type refers to the numerous roles that a user may have based on data contained in the database, and this information is encrypted in the server-generated token. Different responsibilities are assigned to each job, and only the tasks that the consumer is capable of are displayed. It also describes the ability to remark on something based on the user's position. This article contains facts about global header variables that share product logo details, current user name, and customer Id information. It also has a logout option, which allows the user to logoff from the server. This starts the API request and instantly expires the token, protecting the user's information from any danger or infection. Often we are also required to pass certain along with the API request, these parameters are either passed in body of the request, then referred to as Body Params or separately, called as Request Params. Request Params become part of the url of the request as well. There are various data formats of passing the request parameters such as Raw, json, etc depending upon the user requirement we can select any one of these format and then pass the requisite parameters in this to obtain the desired Response.

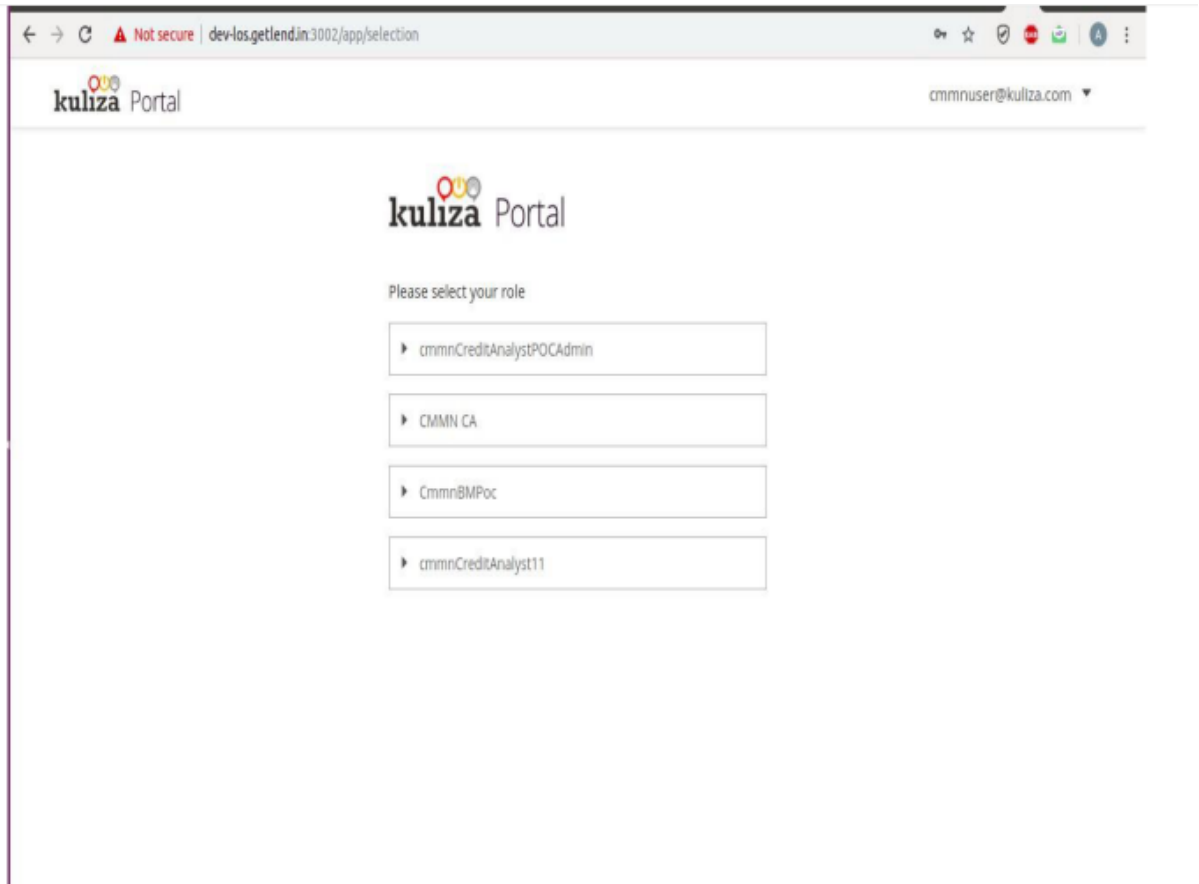


Fig 11: Role Selection page

The user is brought to the listing page after selecting the role from that page. This page is critical since it displays the many sorts of loans offered by the bank from whom the customer is attempting to obtain a loan. It displays the complete version of every sort of loan as well as all applications based on the bucket selected from claim, pending, history, and authorised from the drop-down menu. These are known as buckets, and they store information about each loan. For example, claim displays the applications that have been claimed by one or more agents to be checked, history displays the applications that have been completed and all payments have been made in the past only, pending displays the applications that have yet to be checked but have been taken up by the agent, and approved displays the applications that have been approved.

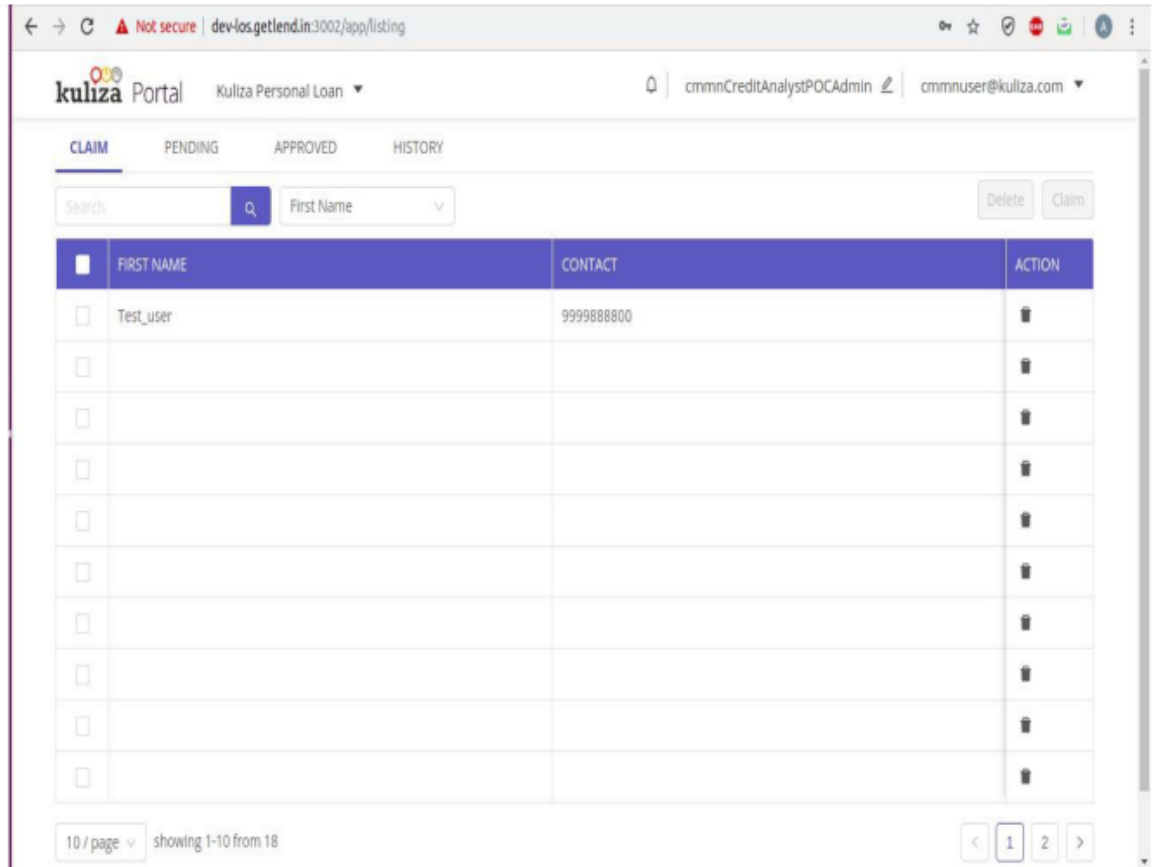


Fig 12: Selection page

The list appears in any bucket; the user may click on any application to access its detailed view; the agent working on that application has the authority to approve or refuse the application based on the detailed view. All of the fields are double-checked to ensure that no unauthorized users or hackers get access to the system. Some fields are created when the user enters in one column and it appears in another. This can also be referred to as sub fields of the major fields and there are access button to access these sub fields while a portal user is using the portal and browsing through all the options in the portal. This is one the most instrumental requirements of the banking clients these days, as in the day and age of information, there are tons of kinds of information that need to be stored and organized in an efficient and easy to access manner.

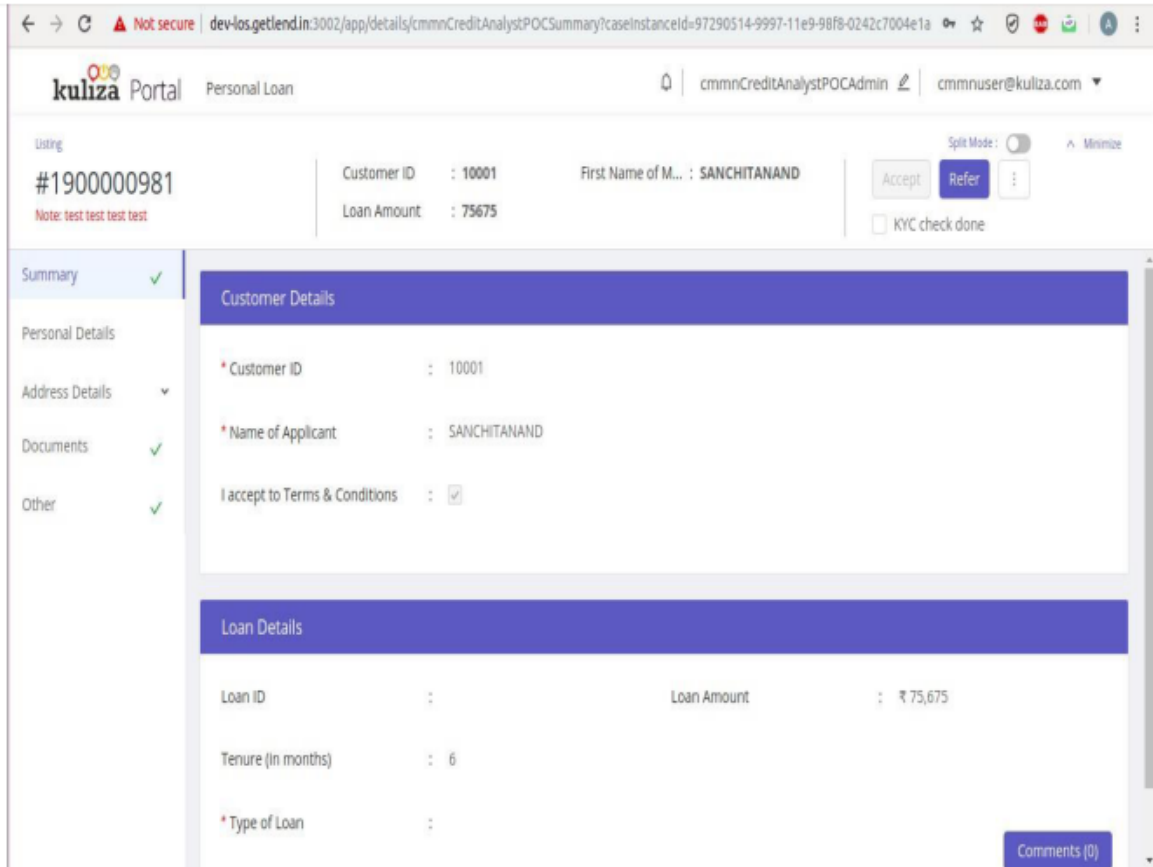


Figure 13 Detailed Description of user

When the user tries to alter the information, a pop-up appears, asking whether the user is sure he wants to proceed since the changes will be reflected in the database.

The user may also upload documents by pressing a button and then selecting "Upload" from the drop-down menu. A new window appears, displaying the user's desktop download window; from there, the user may download files, may upload the document after selecting it. After you've chosen a document, it'll open in a split-screen mode. The user may then check to see if it is correct.

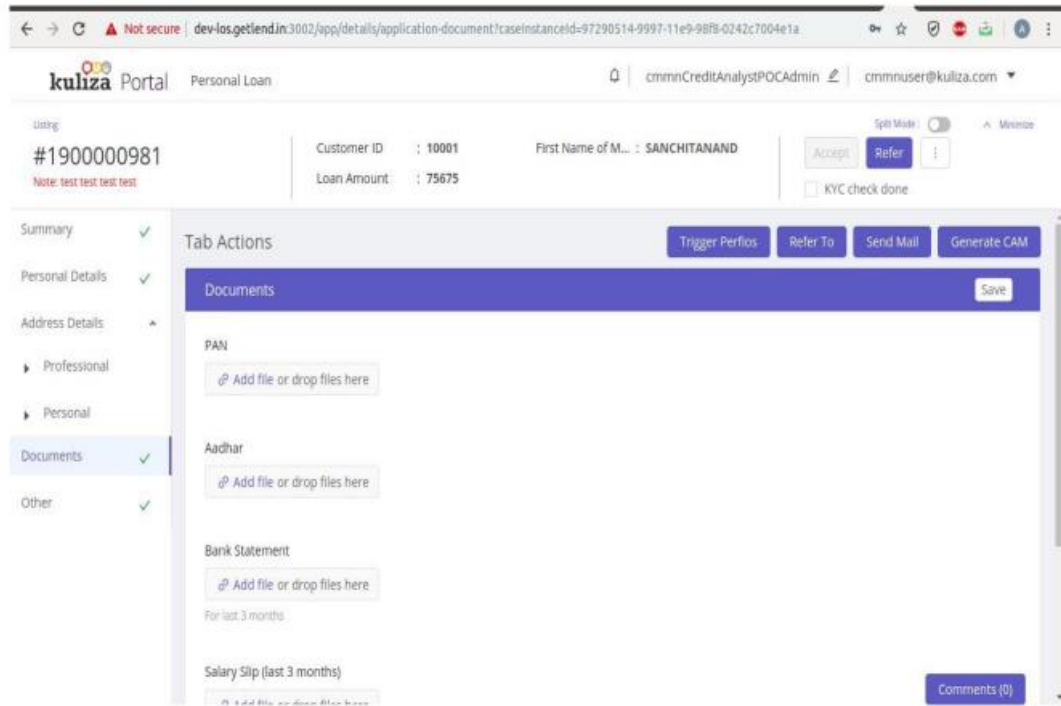


Fig 14: Upload document window

The project also has a comment option, which allows users to submit complaints, requests, or concerns about document verification. Whether or not a user is capable of commenting is entirely dependent on his job; nonetheless, these remarks can be beneficial in resolving any issue as quickly as possible.

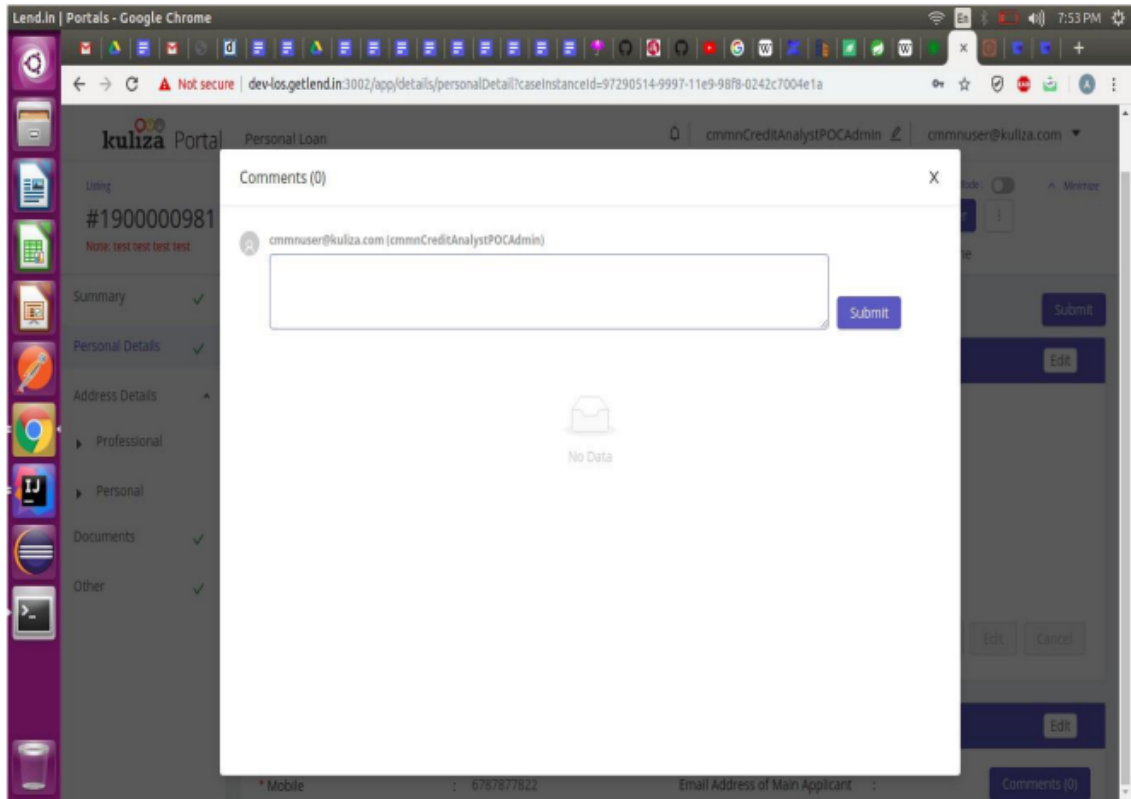


Fig 15: Comment section

It also includes an editable component that is kept in the database table as a string, and when it is retrieved from the table, it is transformed to a table format and information is extracted from it. It may be used to gather information and conduct a SQL query on the data retrieved from the database. A temporary table is created and data is saved in it while the sql query is executing, and the custom query runs over it.

Testing with Postman –

1. Either write the API manually or Import it in any format like text file, folder, link or raw text of curl.
2. If the API is of POST method then write request body in the provided column, and run the API.
3. Authorisation can also be provided either by Bearer token of some credentials.

4. “Headers can be given, by this postman acts as the browser running the API, the difference is that it does not shows the html page as the result instead shows the response data.”
5. Postman supports multiple request methods like GET, POST, PUT, DELETE etc.
6. The response can be checked in the response column.

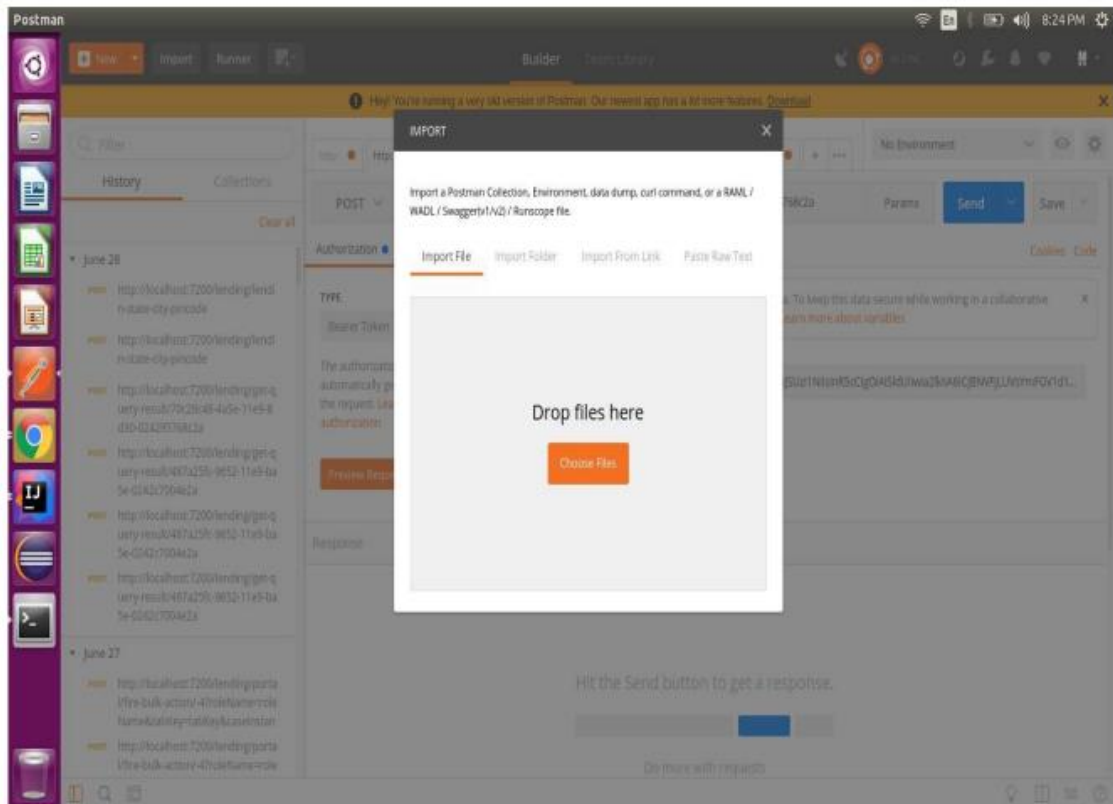


Figure 16 Importing Api

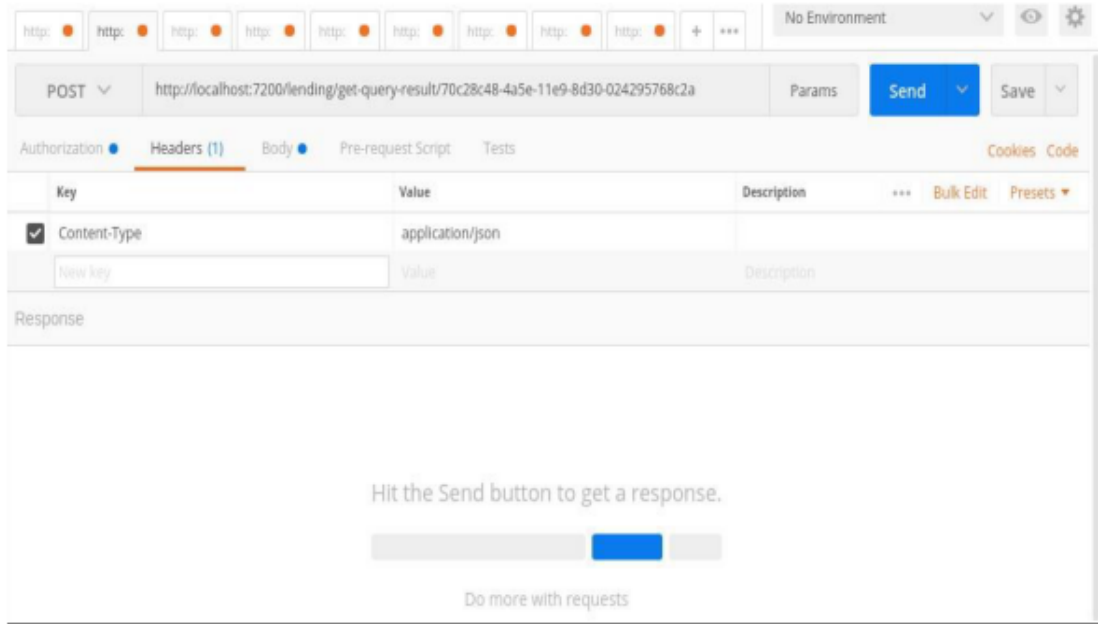


Fig 19: Provide headers

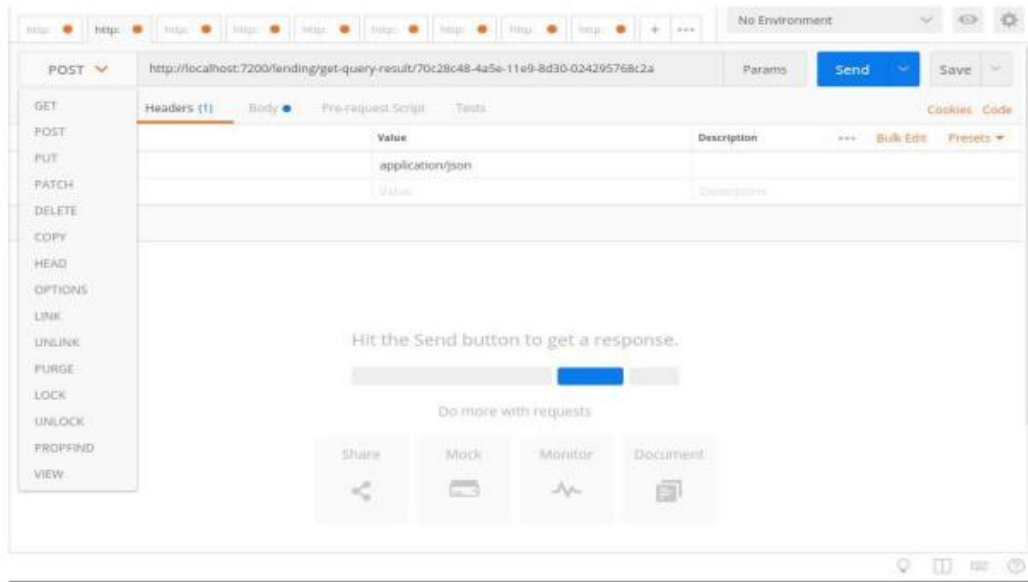


Fig 20: Request methods

Chapter 4 – Implementation Details

4.1 Use Cases

Claim/Unclaim:

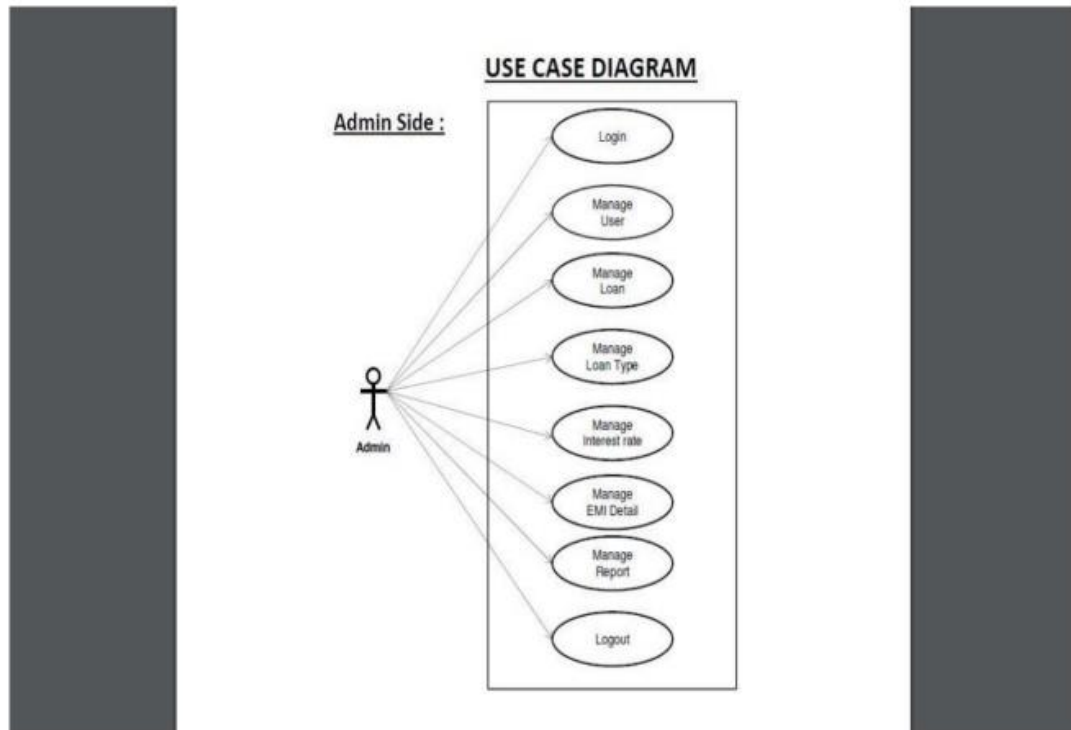


Fig 21: Role Map of user

1. If Claim allocation is selected for group, all the cases assigned to the group are visible to all the users of that group in their 'Claim' buckets. - done.
2. User cannot go to detailed view from a Claim bucket - done
3. Once a Case has been claimed by a user, it should move to his assigned/pending bucket as per configuration. The case should also move out of the Claim buckets of other users so that the same case is not claimed by multiple users. - done
4. "The User who can Claim a case, should also have the action for Release/Unclaiming the case.(This should be allowed only after entering a comment) On Release, the case moves back to the Claim buckets of all users. Release / Unclaim CTA should be in Details View"- **configurable**

5. If one of the Users is a supervisor for this group, he has to choose between the supervisor view & the user view.
6. “As a supervisor he can see which cases are allotted to whom and can Unclaim/Reassign a case assigned to any user in that group to another user after entering a comment. If a user is working on the case, the supervisor is reassigning, the supervisor should get a notification of an active session and should not be allowed to reassign the case or a confirmation popup asking if he/she wants to proceed with the action. Supervisor should have Unclaim / Reassign CTA on List view.”

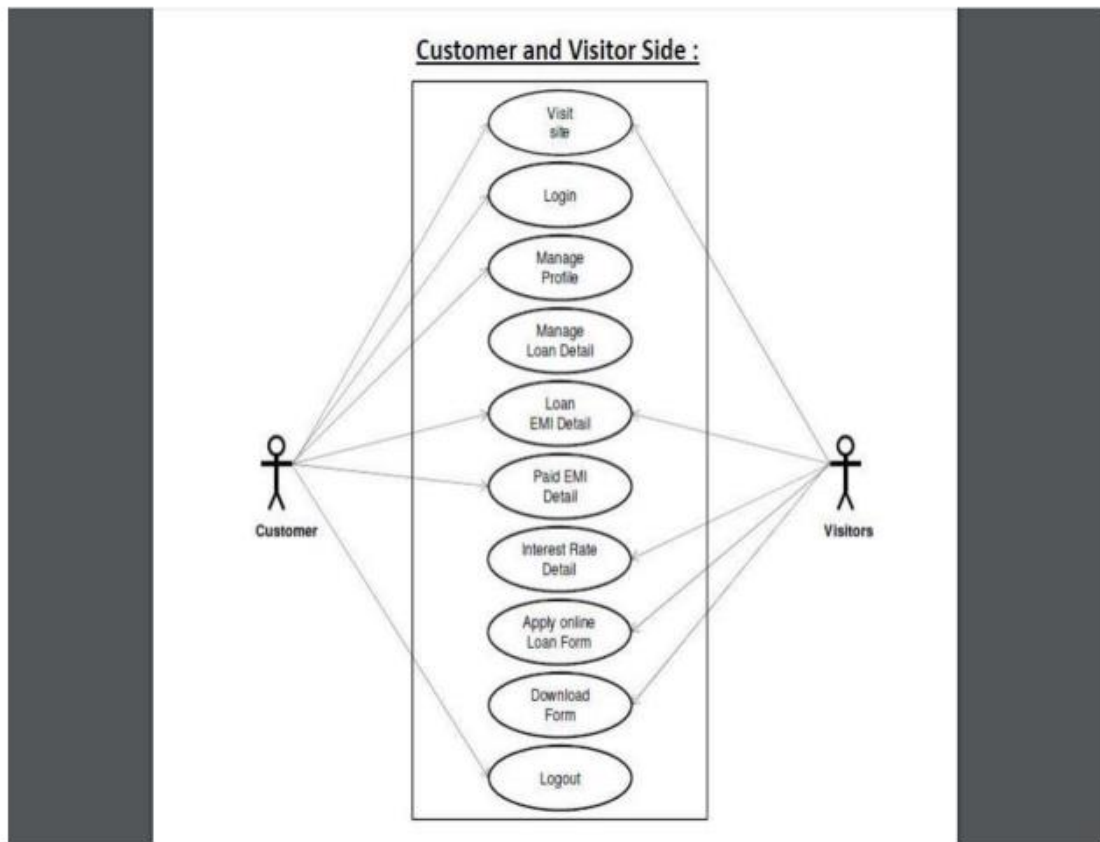


Figure 22 Server Client Model

Disable logic for Action buttons :

1. An action button can be dependent on fields in the same card or another card.
2. It can be dependent on fields in the same tab or another tab
3. It can be dependent on other actions in the same card, same tab or another tab.
4. Global actions can be dependent on fields anywhere in the entire application and other actions in the application.
5. Only when the fields it depends on is filled and validated, & the action it depends on is successfully triggered, the action button should be enabled.
6. If any changes are made to the fields it depends on, the value of the button should be reset to false and user must be asked to trigger it again.

Below is an example of the above use case:

1. “An application contains an action ‘Trigger CIBIL’ inside a card C1, in tab T1.
2. C1 depends on the field PAN (v1) and auto-populates the field Cibil Score (v2)
3. There is another action button in Tab T2 , Trigger Credit Model which depends on various fields(v2,v3, v4, v5,v6) in tabs T1 & T2 and action Trigger Cibil.
4. Global actions Accept/Reject depend on Trigger Credit Model , Generate CAM and other fields (v7,v8,v9,v10,v11) in the application.
5. Now Trigger CIBIL should not be enabled until PAN is filled & validated.
6. Trigger Credit Model should not be enabled until the fields v2, v3,v4,v5,v6 are filled and validated & Trigger Cibil successfully triggered.
7. Global actions should not be enabled until Credit Model is successfully triggered , CAM successfully generated and fields v7,v8,v9,v10,v11 are filled and validated.
8. Now in case v1 ie. PAN has been changed, the entire cycle/chain should be reset. Trigger Cibil should be set to False and user must be asked to Trigger it again. And Trigger Credit model should be set to False and disabled and Global actions disabled.
9. Similarly, in any fields v2 to v6 are changed, Triggered Credit model should be set to false and global actions disabled.”

4.2 Dependent Fields

“**Dependent Fields** by definition means fields which would be auto populated with values or options on current field's trigger api or otherwise”. Dependent fields can be used with the following components: -

1. Rest button: On successful trigger, certain fields are auto populated
2. Rest action: On the Api call in this field, certain other fields are auto populated
3. Rest dropdown: The dropdown value for this field can come from a master or any other API. Based on a selected value in this field, certain other fields are auto populated
4. Autocomplete: The dropdown value for this field can come from a master Based on a selected value in this field, certain other fields are auto populated
5. Elastic Autocomplete: The dropdown value for this field can come from a master Based on a selected value in this field, certain other fields are auto populated
6. For a case where the dropdowns of a field Y depend on a value selected in field X , autocomplete and elastic- auto complete can be used.
7. In this case, the in meta for field Y, the API call to fetch the values uses field X. Thus, the values returned from the API for Y are filtered based on value of X.

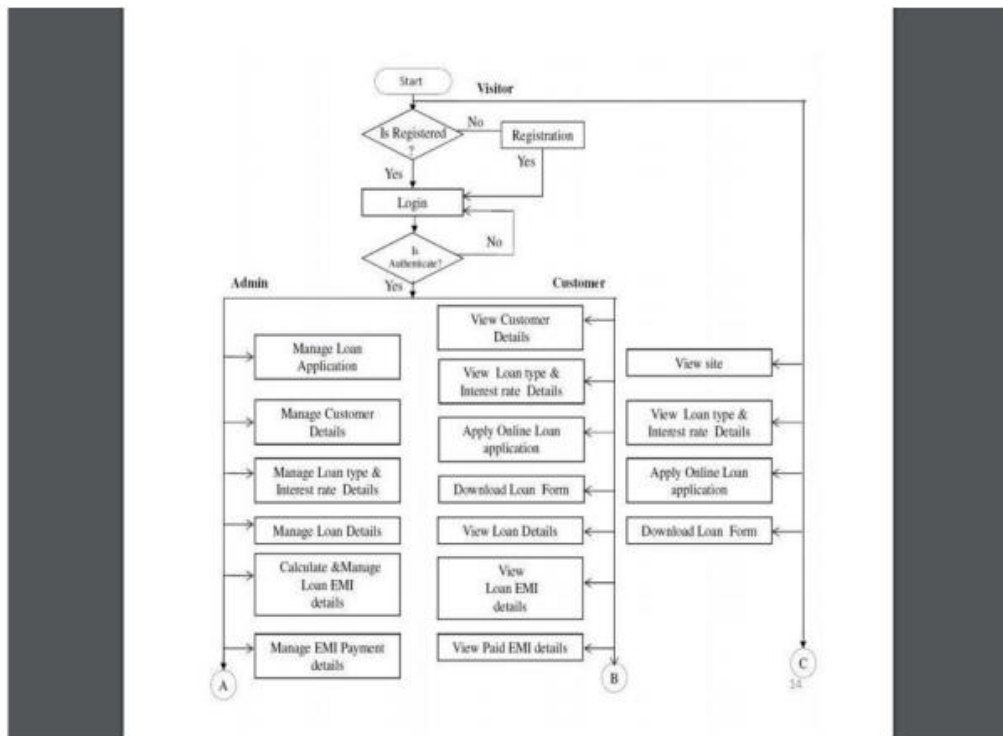


Fig 23: Loan Approval Flow diagram

4.3 Types of Actions

1. Global : Example Accept, Reject.

Once the user clicks on these actions, he has to enter a comment and submit.

On Submit, a message is shown to the user.

When the user clicks on Okay, User goes back to list view

2. Email/Notification Triggers: Once user clicks on these actions, an API is triggered and a Success/Failure message is shown to the user. User stays on the same card/page

3. “Actions through which certain fields are populated, Ex: Trigger Credit Model, Trigger CIBIL. When a user clicks on this action, fields that are populated are saved using Partial Save API.”

Success/Failure message is shown to the user

User stays on the same card/page

4. Document generation : Ex: CAM generation - The document generated opens in a new tab and is saved in the variable(form_type:upload) that is mapped in the meta. User stays on the same card/page.

5. Modal/Popup on an action: On click a form opens that has fields and actions (Ex: Save/Submit, etc.)

6. Claim action: The case moves to another bucket (defined in CMMN). Button is disabled. On refresh, case moves out of the bucket. User stays on the same list/bucket.

7. Assign/Refer: Dropdown is shown on the click of these actions. The values of dropdowns will come from an API (masters/keycloak/assignment logic).

a) Detailed view: On the selection of a particular dropdown value, user has to enter a comment and submit. On Submit, a message is shown to the user. When the user clicks on Okay, User goes back to list view

b) List view: On selection of a particular dropdown value, assignee of the case will change. User will stay on the same page.

Functionalities needed:

1. Meta to have an attribute to set the max number of times any action can be triggered. There could be no upper limit for triggering certain actions

2. The success and error messages displayed to the user, should be configured through meta
3. Dependencies to be defined in the meta. Any prerequisites: fields or actions that are needed for an action to be enabled should be defined in the meta.



Notes:

- Each tab will be considered as a task.
- History bucket will fetch all cases for which list view screen is closed.
- Each portal will have separate caseModel (e.g. BackOffice, LMS, Collection, Collateral, Customer Dashboard)
- Each portal will have multiple roles (e.g. CA, OPS, FI) ● Each role will have separate configuration.
- One configuration will have separate stage.

4.4 Unit Testing

Unit testing is one of the most important activities that a developer should adopt and practise; he or she should make it a rule that if the developer writes a piece of code, he or she should also have his or her unit test code ready; this not only benefits the user by allowing them to test it, but it also helps the developer if he or she decides to include any code, making it easier for him or her to enforce his or her code.

Unit testing examines the code, and one of its most important properties is coverage. The primary goal of unit testing is to ensure that each piece of code functions as intended.

When a large function is written in a modular manner, each piece of code/method in which it is broken is referred to as a unit.

It is very important as developer sees the proper execution of code and helps him to debug the code. Tools used to unit test are -

1. Jtest
2. Junit
3. Mockito

Code Coverage

In Portal, testing of every controller and service including all common helper function is done and the code coverage of 95.7% came out of 100%. Whenever any new piece of code is added then it can be tested immediately and precise areas of concern can be recognized by the developer.

COMMON ERRORS:

Below is a list of errors that are frequently encountered while configuring a journey.

It contains a list of possible reasons that cause that error.

It may happen that the error occurs due to a reason not in the list. We will keep updating as we come across more.

You can check these under Inspect : Network & Console

1. 500 : Something went wrong

Possible Reasons :

1. Error in Script task format
2. Field used in script task (Get Variable) or Service task is not defined before the tasks are triggered
3. Error in service task format
4. Error in function/API that was called in service task
5. The ids used in script task or service task can be different compared to the field ids used in forms
6. Server related error
7. API error in case the field has an API error
8. Mapping error in case an API has been triggered
9. Assignee or Task Id not present in the User task
10. Assignee is incorrect.
11. Mapping of variables in a Decision Table
12. Boolean values such as TRUE/FALSE are defined as string
13. If the field is rest-input, make sure to add type : rest in the meta

2. Missing field error

Possible Reasons :

1. A field called as child to another field has not been defined. Ex : In case of an accordion,table or display-of-selection

3. 401

Possible Reasons :

1. Invalid token. The token could have expired
2. Invalid user credentials. Either credentials entered are incorrect or have not been defined in keycloak

4. TypeError: Cannot read property 'map' of null : This is mostly due to error in javascript

Possible Reasons :

1. Script task error
2. Error in meta . For example the component dragged is text but form_type is dropdown and no enums and options have been defined for the dropdown
3. The field expects an array but value is being saved in a different format

5. Unable to Publish an app with BPMN Flow

Possible Reasons :

1. “One of the tasks does not have an id or assignee
2. Script task has no script , not even a dummy one
3. Service task has no expression
4. Exclusive gateway has no conditions defined or not marked as default
5. The condition defined on an exclusive gateway is in incorrect format
6. The ids for two tasks are same
7. Exclusive gateway has been used in the flow but only one flow is defined post that. There is no forking, no other flow coming out of an exclusive gateway”.

CHAPTER 5: Conclusions

5.1 Findings

During the creation of this project, I learned a variety of skills, including how to create clean code, how to build a database, many phases of web development, backend development, system design concepts, Java, Spring framework, SQL, and Rest APIs. Today's world is making enormous strides in web development, and now is the time to act. Many large corporations are concentrating their efforts only on this sector. Microservice architecture is superior than monolithic architecture in terms of functionality and speed. This is because when a monolithic design architecture's server fails, the entire product suffers, however this is not the case with microservice architecture. JPA - how to use it and develop a whole application from the ground up is a huge undertaking, but the JPA library aided in optimization and design.

5.2 Conclusion and Future Scope

My role at Lend.in is that of a backend developer, where I write validations and enhance functionality to make the site perform as it should. It is a user-defined and adaptable online application that may be used by businesses such as banks to approve loans and other financial services. It comes in handy and may decrease employee workload, paperwork, and the need to visit any official premises, as well as saving a lot of time for both the user and the employee working in that application. This procedure significantly reduces the number of mistakes because everything is computerized and requires very little human participation.

We intend to improve the application's presentation by implementing a resurrection store on the finer points page and connecting each leaf portion to it. As soon as the user types something, it will assist in the automated process. This will also eliminate the needless re-rendering of distinct portions that haven't been updated.

REFERENCES

- [1] "Postman" : <https://wiki.onap.org/display/DW/Testing+with+Postman>
- [2] "Spring Framework" : https://en.wikipedia.org/wiki/Spring_Framework
- [3] "Lend.in" : <http://dev-los.getlend.in>
- [4]"KeyCloak":<https://www.comakeit.com/quick-guide-using-keycloak-identityaccessmanagemen>
- [5] "Spring boot projects" : <https://spring.io/projects/spring-boot>
- [6] HaiTao Wang, Baozian Jia, "Research Based on Web Development of Spring Integration Framework", Published in 2010 International Forum on Information Technology and Applications
- [7] Dandan Zhang, Zhiqiang Wei, Yongquan Yang, "Research on Lightweight MVC Framework Based on Spring MVC and Mybatis", 2013 Sixth International Symposium on Computational Intelligence and Design
- [8] Deepak Kasgar, Harish Chandra Maurya, "Integration of Struts & Spring & Hibernate for Enterprise Applications", Published in International Journal Of Modern Engineering Research (IJME)

Certificate

Project Report Undertaking

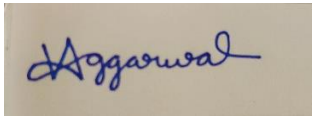
I Mr. Vineet Aggarwal -Roll No. 171251
Branch Computer Science is doing my internship with Kuliza
Technologies from 1st Feb 2021 to 30th July
2021

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Signature



Name Vineet Aggarwal

Roll No. 171251

Date 22nd May 2021