

Presented by: George Antony, Sai Charan, Boppudi Vivek and Varun Raj

MFPE

Collateral Loan - Risk Assessment

*Mentor : Ramaraj, Ashok
Trainer : Wazeem, Akram*

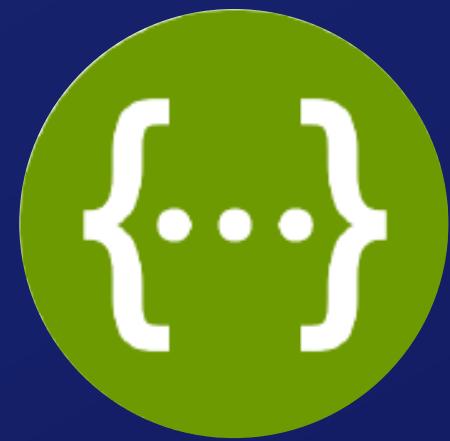
What does the system do?

The system helps the banks back-office staff to submit collateral details into the database through a form, calculates the risk each of these collaterals possesses to the bank and displays the calculated risk back to the office staff.

Technologies Used



Dotnet Core API



Swagger



Angular

Moq



Azure



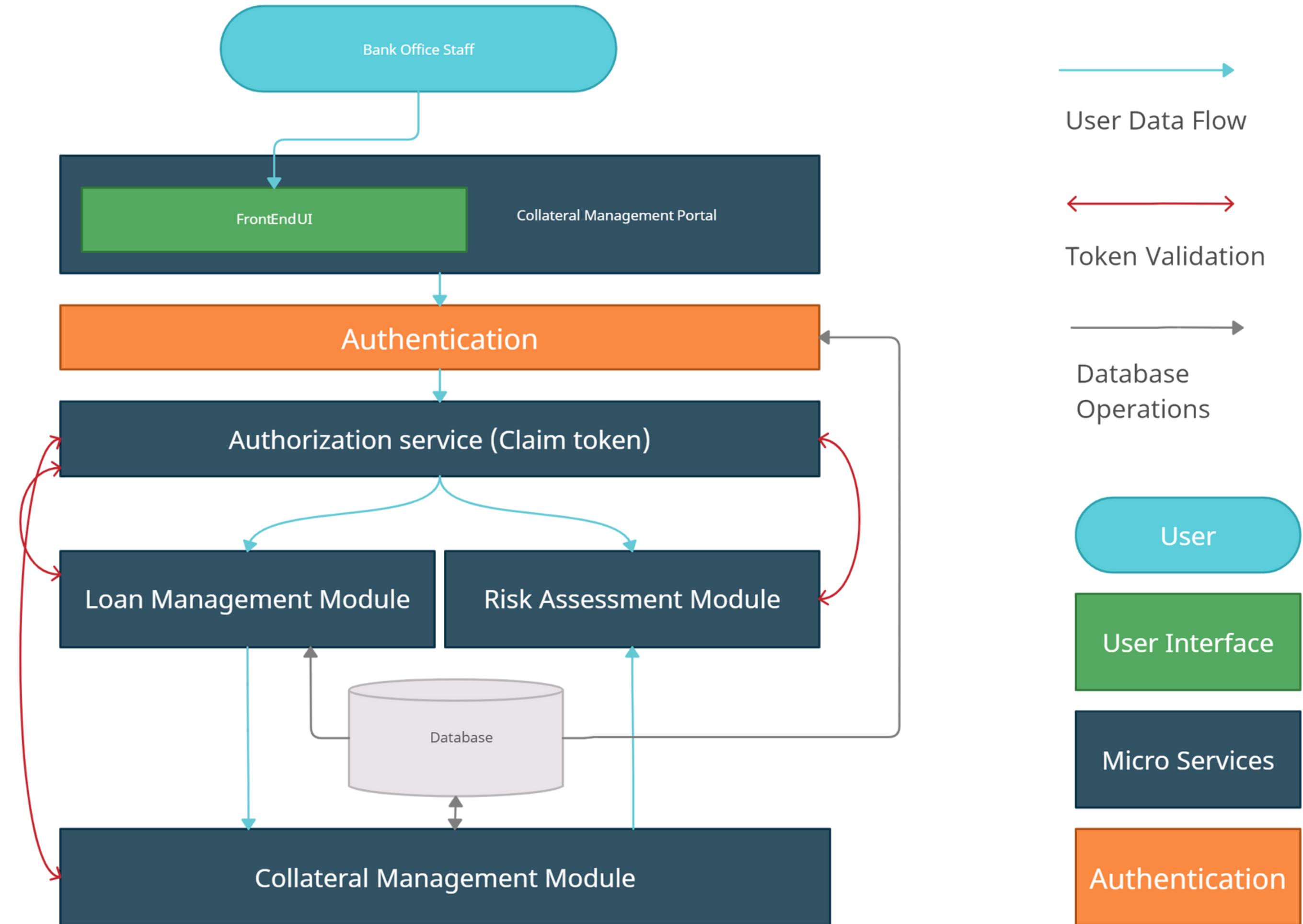
Nunit



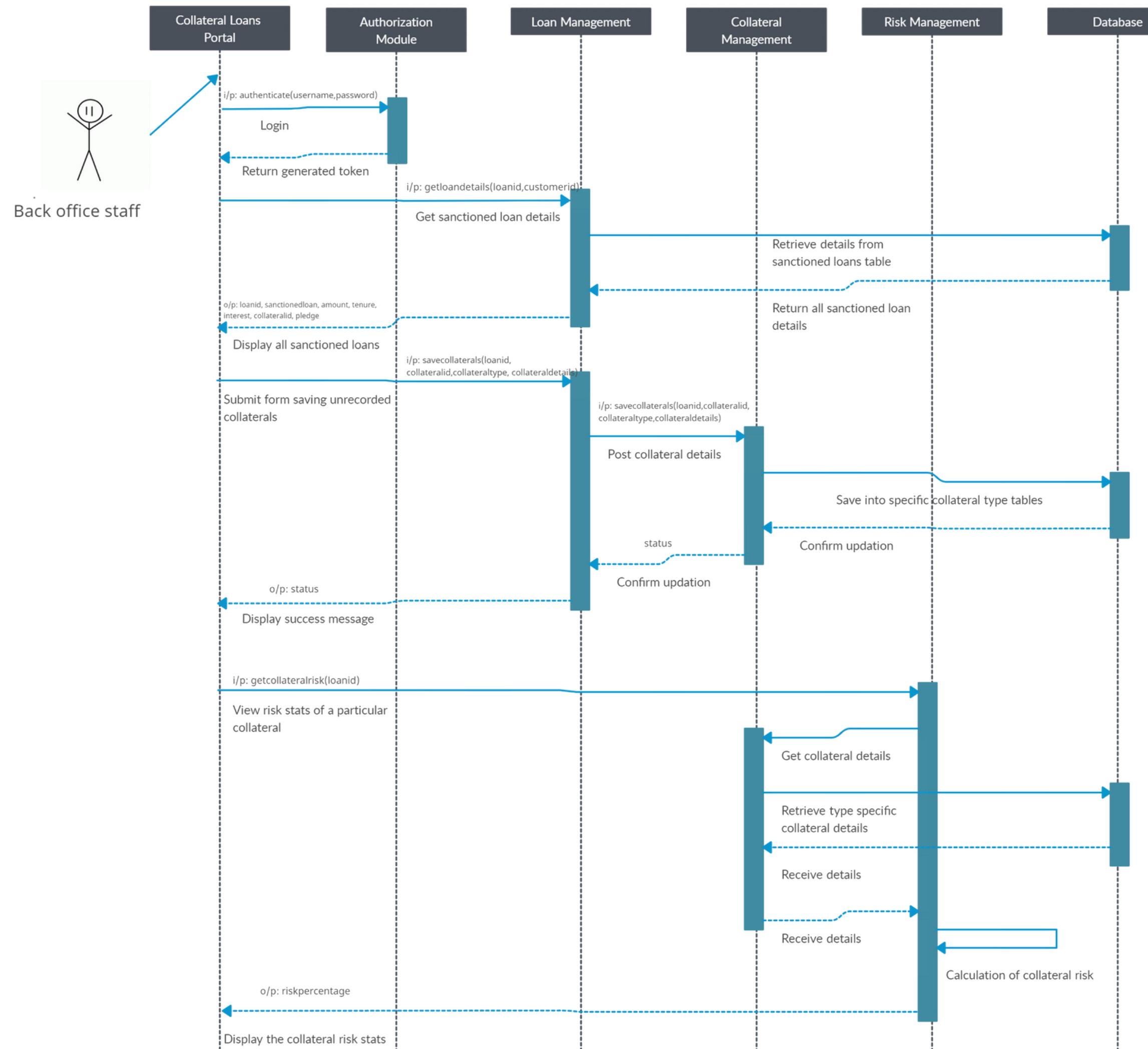
Work- Flow Diagrams

The diagrams created from the requirement documentation, that was used as the blueprint

Use-case Diagram



Sequence Diagram



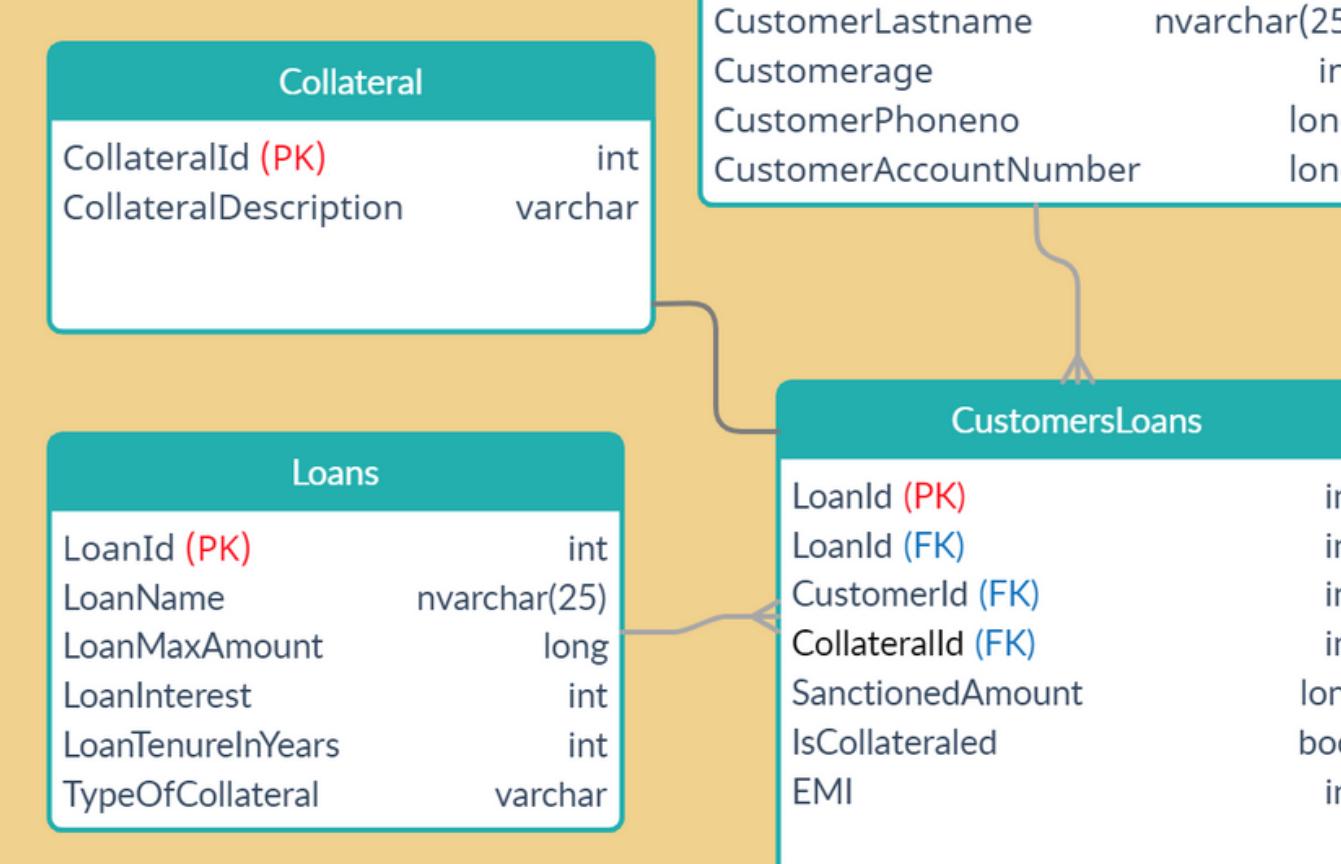
Authorization Micro-Service

Back end staff

Username
Password

nvarchar(25)
nvarchar(25)

Loan Management Micro-Service



Land Collateral

LandId	int
CollateralId	int
LandOwner	varchar
LandAddress	varchar
LandRatePerSquareFeet	int
LandArea	int
LandDepreciationRate	int
LandPledgedDate	datetime

House Collateral

HouseId	int
HouseOwner	nvarchar(25)
HouseArea	int
HouseCRatePerSquareFeet	int
HouseAddress	varchar
HouseYearBuilt	int
HouseDepreciationRate	int
HousePledgedDate	DateTime

Gold Collateral

GoldId	int
CollateralId	int
GoldWeight	int
GoldOwnerName	nvarchar(25)
GoldCurrentGramRate	int
GoldPledgedDate	datetime

Collaterals

CollateralType	int
CollateralId	int
LoanId	int
CustomerId	int

Collateral Management Micro-Service

Database Mapping

Overview

Visual Studio used for compiling.

Entity Framework code first approach used throughout.

Sql Server Express used as database.

Swagger and *Postman* used to test the endpoints.

Log4net used for logging purposes.

All services deployed to *Azure* cloud after development and testing.

Service Modules

Loan
Management
Microservice

Authorization
Microservice

Risk
Assessment
Microservice

Collateral
Management
Microservice

Authorization Service

To authenticate and authorize the back-end staff, they need to login with valid credentials. Once validated, a token is issued by the service to authorize the use of all the other three services.

Authorization Service

- 1 REST End points
POST: Authenticate([FromBody] LoginModel model)
- 2 Input/Output: LoginModel/JWT
Url: <https://authorizationservicemfpe.azurewebsites.net/api/login>
- 3 Invalid/Successfull Login
- 4 Generation of JWT (token expiration: 15mins)

Loan Management Service

This service retrieves all of the sanctioned loans details that are to be displayed, from the database. This service also gets the submitted collateral details and calls the collateral management service for saving them.

Loan Management

Service

1

REST End points

GET: GetSanctionedLoansById(int LoanId,int CustomerId)

POST: SaveCollateral([FromBody] JsonElement value)

2

Input/Output: (i) CollateralID/AllDetails

Url: <https://loanmanagementservice.azurewebsites.net/api/loanmanagement/getsanctionedloans/1/2>

3

Input/Output: (ii) CollateralDetails/Status

Url: <https://loanmanagementservice.azurewebsites.net/api/loanmanagement/savecollateral>

4

Both the endpoints are accessed from the client application

Collateral Management Service

This service is not directly exposed to the front-end, it deals with the collaterals and its specific types. It saves collateral details to the database on request from loan management service by classifying it into a type of collateral. It also provides risk assessment service with all the necessary details to calculate risk.

Collateral Management

Service

1

REST End points

POST: SaveCollateral([FromBody] JsonElement element)

GET: GetCollaterals(int collateralid)

2

Input/Output: (i) AllDetails/status

Url: <https://collateralmanagementservice.azurewebsites.net/api/collateralmanagement/savecollateral>

3

Input/Output: (ii) CollateralId/CollateralDetails

Url: <https://collateralmanagementservice.azurewebsites.net/api/collateralmanagement/getcollateral/5>

4

Endpoints (i),(ii) accessed from loan management and risk assessment services respectively.

Risk Assessment Service

This service requests collateral management service for details of a specific collateral, calculates its risk and saves it to a static list. The calculated risk is displayed to the user. The risk is calculated again and the list is updated when the update button on the UI is triggered.

Risk Assessment

Service

1 REST End points

`GET: GetCollateralRisk(int CollateralId, bool Update)`

2 Input/Output: (i) CollateralID,Update/RiskValue

`Url:https://riskmanagementapimspe.azurewebsites.net/api/riskassessment/riskassessment/3/true`

3 Called from the Angular application

Testing

Unit testing with Nunit and Moq

Overview

Nunit and *MOQ* are used for testing

Testing of all Controllers are carried out

Testing of all Service methods are carried out

Testing of all Repository methods are carried out

Authorization Service Testing

The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- Solution Explorer:** Shows the project structure for 'AuthorizationService' containing files like 'Connected Services', 'Dependencies', 'Properties', 'Controllers' (with 'LoginController.cs' selected), 'MemberRepo', 'IMemberRepository.cs', 'MemberRepository.cs', 'Migrations', 'Models', 'Repo', 'TokenRepo', 'appsettings.json', and 'Program.cs'.
- Code Editor:** Displays the 'LoginController.cs' file with C# code for handling login requests. The code includes methods for getting a member from the repository and creating a JWT token if credentials are valid.
- Test Explorer:** Shows 3 passed unit tests under the 'Test' category, grouped by 'TestProject3'. The tests are: 'Tests (3)', ' GetUserCred_Returns_Null("use...")', ' GetUserCred_Returns_Object("u...")', and ' GetUserCred_Returns_Object2("...")'.
- Properties Window:** Opened for 'LoginController.cs', showing settings for 'Build Action' (C# compiler), 'Copy to Output Directory' (Do not copy), 'Custom Tool', 'Custom Tool Namespace', and 'Misc' (File Name: LoginController.cs).

Loan Management Testing

The screenshot shows a Visual Studio interface with the following components:

- MenuBar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- WindowTitle:** AuthorizationService.
- Toolbars:** Standard toolbar with icons for file operations.
- Solution Explorer:** Shows the solution structure:
 - Solution 'AuthorizationService' (4 of 4 projects)
 - AuthorizationService
 - CollateralManagement
 - LoanManagementService
 - LoanManagementServiceTests
 - Dependencies
 - LoanManagementService.cs
- Code Editor:** Displays the `LoanManagementServiceTests.cs` file with the following code:

```
129
130
131
132     options = new DbContextOptionsBuilder<LoanManagementDbContext>()
133         .UseInMemoryDatabase(databaseName: "LoanManagementService")
134         .Options;
135     using (var context = new LoanManagementDbContext(options))
136     {
137         context.Customers.AddRangeAsync(Customers);
138         context.Loans.AddRangeAsync(Loans);
139         context.Collaterals.AddRangeAsync(Collaterals);
140         context.CustomersLoans.AddRangeAsync(CustomerLoans);
```
- Test Explorer:** Shows the test results:

Test	Duration
LoanManagementServiceTests	1.3 sec
Tests (5)	1.3 sec
TestGetLoanDetails	1.2 sec
TestGetLoanDetails	42 ms
TestGetLoanDetails	1.1 sec
TestGetLoanDetails	5 ms
TestGetSanction	18 ms
TestLoanManagement	90 ms

Group Summary:
Tests: Tests in group: 5
Outcomes: 5 Passed
- Status Bar:** Test run finished: 5 Tests (5 Passed, 0 Failed, 0 Skipped) run in 1.7 sec.

Collateral Management Testing

The screenshot shows a Microsoft Visual Studio interface with the following components:

- Solution Explorer:** Shows the solution structure for 'MFPE_FINAL_PROJECT' with seven projects: AuthorizationService, CollateralManagement, LoanManagementService, RiskmanagementService, UnitTestForCollateralManagementService, Dependencies, and two test files (ControllerTests.cs and DbServiceTests.cs).
- Test Explorer:** Displays the execution results of unit tests. The summary indicates 8 Passed tests in 1.5 seconds. The list includes:
 - UnitTestForCollateralManagementService (1.5 sec)
 - UnitTestForCollateralManagementService (1.5 sec)
 - DbServiceTests (3) (1.2 sec)
 - TestSavingCollateral (1.2 sec)
 - TestSavingCollateral... (27 ms)
 - TestSavingCollateral... (1.1 sec)
 - TestSavingCollateral... (23 ms)
 - Tests (5) (275 ms)
 - TestSavingController (11 ms)
 - TestRequestClassifi... (264 ms)
 - TestRequestClassifi... (< 1 ms)
 - TestRequestClassifi... (2 ms)
 - TestRequestClassifi... (1 ms)
- Output:** Shows the build log:

```
Build started...
1>----- Build started: Project: UnitTestForCollateralManagementService, Configuration: Debug Any CPU -----
1>UnitTestForCollateralManagementService -> E:\CTS_Workzz\FinalProjectFolder\CollateralRiskAssessment\MFPE_FINAL_PROJECT\UnitTestForCollateralManagementService\bin\Debug\net5.0\UnitTestForCollateralManagementService.dll
----- Build: 1 succeeded, 0 failed, 4 up-to-date, 0 skipped -----
```
- Code Editor:** Shows the source code for 'ControllerTests.cs'. The code defines a test fixture 'Tests' with a setup method 'Setup' that initializes a database context. It then contains a test method 'TestSavingController' that uses mocks for classification and database saver services to verify a controller's behavior.

Risk Assessment Testing

The screenshot displays the Microsoft Visual Studio IDE interface with the following components:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), and a search bar containing "Riskmanagementapi".
- Toolbar:** Includes icons for file operations like Open, Save, Print, and a Live Share button.
- Test Explorer:** Shows 3 passed, 0 failed, and 0 skipped tests. The results are categorized by project:
 - RiskAssesTest (3) - 20 ms
 - RiskAssesTest (3) - 20 ms
 - Tests (3) - 20 ms
 - TestGold - 18 ms
 - TestHouse - 1 ms
 - TestLand - 1 ms
- Solution Explorer:** Shows the solution structure with projects: RiskAssesTest, Riskmanagementapi, and their respective files like UnitTest1.cs, CollateralGold.cs, etc.
- Code Editor:** The current file is RiskAssesTest.cs, showing the following code:

```
1 using NUnit.Framework;
2 using Riskmanagementapi.Models;
3 using Riskmanagementapi.Services;
4 
5 namespace RiskAssesTest
6 {
7     [TestFixture]
8 }
```

A tooltip for the `readonly struct System.Int32` is displayed, stating "Represents a 32-bit signed integer."
- Output Window:** Shows the build log:

```
Build started...
1>----- Build started: Project: RiskAssesTest, Configuration: Debug Any CPU -----
1>RiskAssesTest -> C:\Users\varun\Downloads\Riskmanagementapi\Riskmanagementapi\RiskAssesTest\bin\Debug\net5.0\RiskAssesTest.dll
===== Build: 1 succeeded, 0 failed, 1 up-to-date, 0 skipped ======
```
- Status Bar:** Shows "Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 0:10 ms".

Hosting

Deploying in Azure with App Services

Authorization Microservice

The screenshot shows the Swagger UI interface for the AuthorizationService. At the top, the URL is authorizationservicemfpe.azurewebsites.net/swagger/index.html. The top right corner includes a star icon, a refresh button, and a user profile icon. Below the header, there's a "Select a definition" dropdown set to "Authorization Service v1". The main content area displays the "AuthorizationService" title with "v1 OAS3" badges. A link to "/swagger/v1/swagger.json" is provided. Under the "Login" section, a green "POST /api/Login" button is visible.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes "Microsoft Azure", a search bar, and user information for "vivekboppudi123@outlook.com". The main content area shows the "AuthorizationServicemfpe" App Service resource under "MFPE > Resource groups > Home". The "Overview" tab is selected on the left sidebar. The "Essentials" section provides detailed information about the app service, including its URL, status, location, subscription, and various connection details. The "JSON View" link is located at the bottom right of the essentials table.

Essentials	
Resource group (change)	: MFPE
Status	: Running
Location	: South Central US
Subscription (change)	: Azure Pass - Sponsorship
Subscription ID	: df993f57-38d2-411e-bfdf-90b7fae7faa2
Tags (change)	: Click here to add tags
URL	: https://authorizationservicemfpe.azurewebsites.net
Health Check	: Not Configured
App Service Plan	: AuthorizationService20210805091731Plan (S1: 1)
FTP/deployment username	: No FTP/deployment user set
FTP hostname	: ftp://waws-prod-sn1-091.ftp.azurewebsites.windo...
FTPS hostname	: https://waws-prod-sn1-091.ftp.azurewebsites.windo...

Collateral Management Microservice

The screenshot shows the Swagger UI interface for the CollateralManagement microservice. At the top, the URL is collateralmanagementservice.azurewebsites.net/swagger/index.html. The title is "CollateralManagement v1 OAS3". On the left, there's a sidebar with "CollateralManagement" and a "CollateralManagement" section containing two API endpoints:

- POST /SaveCollateral**
- GET /GetCollateral/{collateralid}**

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes "Microsoft Azure", a search bar, and user information. Below it, the navigation path is "Home > Resource groups > MFPE > CollateralManagementService". The main content area displays the "CollateralManagementService" App Service details:

Overview		Essentials	
Activity log	Resource group (change) : MFPE	URL	: https://collateralmanagementservice.azurewebsites...
Access control (IAM)	Status : Running	Health Check	: Not Configured
Tags	Location : South Central US	App Service Plan	: AuthorizationService20210805091731Plan (S1: 1)
Diagnose and solve problems	Subscription (change) : Azure Pass - Sponsorship	FTP/deployment username	: No FTP/deployment user set
Security	Subscription ID : df993f57-38d2-411e-bfdf-90b7fae7faa2	FTP hostname	: ftp://waws-prod-sn1-091.ftp.azurewebsites.windo...
Events (preview)	Tags (change) : Click here to add tags	FTPS hostname	: https://waws-prod-sn1-091.ftp.azurewebsites.windo...

Loan Management Microservice

The screenshot shows the Swagger UI interface for the **LoanManagementModule v1**. At the top, there's a navigation bar with icons for back, forward, and search, followed by the URL `loanmanagementservice.azurewebsites.net/swagger/index.html`. On the right side of the header, there are links for "Select a definition" and "LoanManagementModule v1". The main content area features the title **LoanManagementModule v1 OAS3** and a link to `/swagger/v1/swagger.json`. Below this, under the heading **LoanManagement**, there are three API operations listed:

- GET /api/LoanManagement/getsanctionedloans**
- GET /api/LoanManagement/getsanctionedloans/{LoanId}/{CustomerId}**
- POST /api/LoanManagement/savecollateral**

The screenshot shows the Microsoft Azure portal interface for the **LoanManagementService** App Service. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information for `saicharankalyanreddy@...` in the `DEFAULT DIRECTORY`. The main content area displays the service details:

LoanManagementService (App Service)

Overview (selected) | Activity log | Access control (IAM) | Tags | Diagnose and solve problems | Security | Events (preview) | Deployment

Browse | Stop | Swap | Restart | Delete | Refresh | Get publish profile | Reset publish profile | Share to mobile | ...

Click here to access Application Insights for monitoring and profiling for your ASP.NET Core app. →

Essentials

Resource group (change)	:	MFPE	URL	:	https://loanmanagementservice.azurewebsites.net
Status	:	Running	Health Check	:	Not Configured
Location	:	South Central US	App Service Plan	:	LoanManagementService20210805105437Plan (S1...)
Subscription (change)	:	Azure Pass - Sponsorship	FTP/deployment username	:	No FTP/deployment user set
Subscription ID	:	83f1f429-00f1-4386-86cb-05816b1d21db	FTP hostname	:	ftp://waws-prod-sn1-097.ftp.azurewebsites.windo...
			FTPS hostname	:	https://waws-prod-sn1-097.ftp.azurewebsites.windo...

Risk Assessment Microservice

The screenshot shows the Swagger UI for the Riskmanagementapi v1 endpoint. The URL in the browser is `riskmanagementapi.azurewebsites.net/swagger/index.html`. The top navigation bar includes 'IE Bookmarks' and 'Using Refresh Toke...'. The main header says 'Swagger' supported by SMARTBEAR. On the right, there are buttons for 'Select a definition' and 'Riskmanagementapi v1'. Below the header, the title 'Riskmanagementapi v1 OAS3' is displayed, along with the URL `/swagger/v1/swagger.json`. The main content area is titled 'RiskAssessment' and contains a single API endpoint: `GET /RiskAssessment/{CollateralId}/{Update}`.

The screenshot shows the Microsoft Azure portal's Overview page for the 'Riskmanagementapimspe' App Service. The left sidebar shows navigation links: Home, Resource groups, Riskasses, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events (preview), Deployment, Quickstart, Deployment slots, and Deployment Center. The main content area displays the following details:

Setting	Value
Resource group (change)	Riskasses
Status	Running
Location	South Central US
Subscription (change)	Azure Pass - Sponsorship
Subscription ID	2ed48a5a-fe3b-4952-99b8-1548a1c615fe
URL	https://riskmanagementapimspe.azurewebsites.net
Health Check	Not Configured
App Service Plan	Riskmanagementapi20210805104107Plan (S1: 1)
FTP/deployment username	No FTP/deployment user set
FTP hostname	ftp://waws-prod-sn1-087.ftp.azurewebsites.windows.net/site/www
FTPS hostname	https://waws-prod-sn1-087.ftp.azurewebsites.windows.net/site/www

Below the main content, there are three cards: 'Diagnose and solve problems', 'Application Insights', and 'App Service Advisor'.

Angular Client Application

Collateral Risk
Assessment
Portal

Overview

Visual Studio Code used for compilation

angular2-notifications and *ng-spinner* npm packages added

Form module, routing module imported

Bootstrap, JQuery, font-awesome and other front-end frameworks added

Collateral Risk Assessment Portal



Components

Login Component
Landing Page Component
Collateral Submission Component
Risk Analysis Component



Services/Modules

Data Service
Auth-Guard Service
ngForms Module
Routing Module
HttpClient Module



External Modules

Angular2-notifications
Ngxspinner

Major Challenges

Overcome

Obstacles we overcame
during development phase

Limiting routing with session storage/local storage in Angular client app.

Working with JObject and JsonElement to send and retrieve JSON data.

Installing npm modules giving peer dependency issues. (version mismatches)

Future Scope

Areas of improvement that can be taken up in the future

Passwords in the database can be stored after encryption..NET Identity can be implemented

The Collateral Risk Assessment portal/ Client Angular app can be deployed to the cloud as well.

Displaying live values of collaterals based on current market values of land, gold and property

Demo

Live Demo of our hosted application



Thank You!

Screenshots for any reference

Welcome to vcvg Loans

Collateral Risk Assessment Portal

Login

If you are an authorized back office staff please proceed to the login page for managing collaterals...

Login Page

vcvg Loans

Username

george@123

Password

.....

Login

Login Page 2

vcvg Loans

Username

Password

Server Offline
Try Again Later



Invalid prompt

vcvg Loans

Username

george@123

Password

.....

Login

Loading screen

[Logout](#)**Home Page**

List of all Sanctioned Loans

loanId

CustomerId

Search

Loan ID	Name	Account Number	Sanctioned Loan Amount	EMI	Collateral
1	Saicharan	1301567	2000000	30000	<input checked="" type="checkbox"/>
2	George	1501322	300000	4300	<input checked="" type="checkbox"/>
3	Varun	1439087	700000	8400	<input checked="" type="checkbox"/>
		1209873	5000000	46400	<input checked="" type="checkbox"/>

Success
Logged In



Form for collateral submission

Collateral Description

3000 sq feet Land Located in Anantapur Bought in 2015

Fill according to reference

Collateral Type

select type

Loan ID

3

Submit Collateral

Collateral Form

Collateral Description

3000 sq feet Land Located in Anantapur Bought in 2015

Fill according to reference

Collateral Type

Land

Loan ID

3

Property

Property Owner Name

Owner Name

Land Owner Name is Required.

Address

Address

Address is Required

Pledged Date

mm/dd/yyyy

Pledged Date is Required

Validation

Area of the Property

Area of the Land

Area is Required

Collateral Successfully Saved!

[Return to list](#)

Confirmation

Risk analysis

Risk Percentage 13%

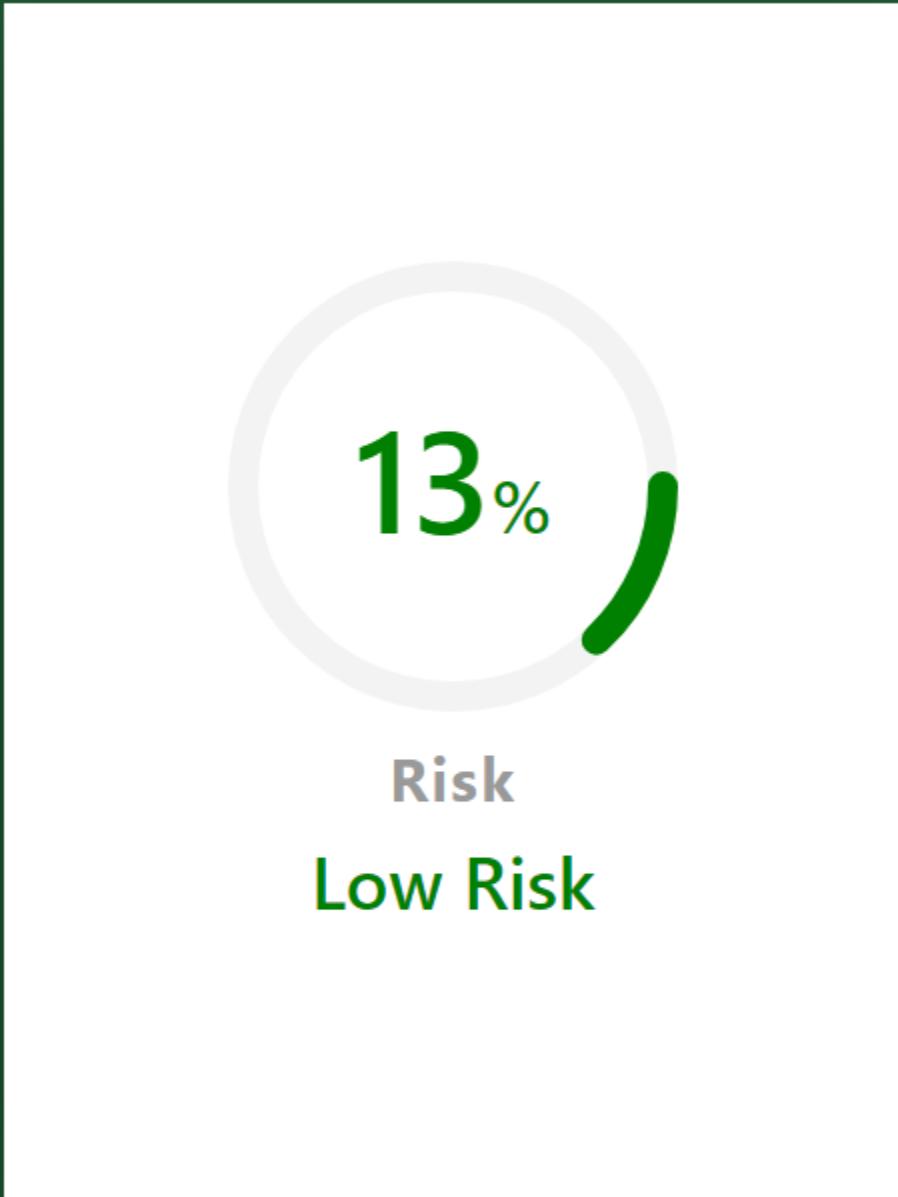
Date Assesed Wed Aug 04 2021 17:14:05 GMT+0530 (India Standard Time)

Collateral Id 1

[Return to list](#)

[Calculate current risk](#)

Risk analysis



Risk analysis

Risk Percentage 45%

Date Assesed Wed Aug 04 2021 17:14:24 GMT+0530 (India Standard Time)

Collateral Id 2

[Return to list](#)

[Calculate current risk](#)

Risk analysis



Moderate Risk

Thank
you!