|  |
| --- |
| IFMR RURAL FINANCE |
| Collections |
|  |
| **Sarthak Shah | IFMR Rural Finance** |
| **8/10/2016** |

Contents

[1. Process Diagram 4](#_Toc459903473)

[2. Stage definition 5](#_Toc459903474)

[3. Stage-role access 6](#_Toc459903475)

[4. Bounce Queue 7](#_Toc459903476)

[4.1 UI specification 7](#_Toc459903477)

[4.2 Screenshot 7](#_Toc459903478)

[4.3 Functional Requirements 8](#_Toc459903479)

[4.4 Uploads 8](#_Toc459903480)

[4.5 Downloads 8](#_Toc459903481)

[4.6 Reports 8](#_Toc459903482)

[5. Promise to pay 10](#_Toc459903483)

[5.1 UI specification 10](#_Toc459903484)

[5.2 Screenshot 10](#_Toc459903485)

[5.3 Functional requirements 11](#_Toc459903486)

[5.4 Uploads 12](#_Toc459903487)

[5.5 Downloads 12](#_Toc459903488)

[5.6 Reports 12](#_Toc459903489)

[6. Collection Queue-Hub Manager 13](#_Toc459903490)

[6.1 UI specification 13](#_Toc459903491)

[6.2 Screenshot 13](#_Toc459903492)

[6.3 functional requirements 14](#_Toc459903493)

[6.4 Uploads 15](#_Toc459903494)

[6.5 Downloads 15](#_Toc459903495)

[6.6 Reports 15](#_Toc459903496)

[7. Collect Payment 16](#_Toc459903497)

[7.1 UI specification 16](#_Toc459903498)

[7.2 Screenshots 18](#_Toc459903499)

[7.3 Functional requirements 21](#_Toc459903500)

[7.4 Uploads 21](#_Toc459903501)

[7.5 Downloads 21](#_Toc459903502)

[7.6 Reports 21](#_Toc459903503)

[8 Deposit Stage Queue 23](#_Toc459903504)

[8.1 UI specification 23](#_Toc459903505)

[8.2 Screenshot 23](#_Toc459903506)

[8.3 Functional requirements 25](#_Toc459903507)

[8.4 Uploads 25](#_Toc459903508)

[8.5 Downloads 25](#_Toc459903509)

[8.6 Reports 25](#_Toc459903510)

[9 Credit Validation 26](#_Toc459903511)

[9.1 UI specification 26](#_Toc459903512)

[9.2 Screenshot 26](#_Toc459903513)

[9.3 Functional requirements 27](#_Toc459903514)

[9.4 Uploads 28](#_Toc459903515)

[9.5 Downloads 28](#_Toc459903516)

[9.6 Reports 28](#_Toc459903517)

[10 Transaction Authorization 29](#_Toc459903518)

[10.1 UI specification 29](#_Toc459903519)

[10.2 Screenshot 29](#_Toc459903520)

[10.3 Functional Requirements 30](#_Toc459903521)

[10.4 Uploads 31](#_Toc459903522)

[10.5 Downloads 31](#_Toc459903523)

[10.6 Reports 31](#_Toc459903524)

# Process Diagram



# Stage definition

|  |  |
| --- | --- |
| **Stage** | **Description** |
| Bounce Queue | All the bounced demands as on date will be listed in the bounce queue. |
| Promise to Pay | The Loan Officer will capture the P2P reasons with P2P date.  In this screen, Loan Officer can mark the customer into different categories as A,B,C,D |
| Collect Payment | Loan Officer collects repayment in the form of Cash, NEFT , Cheque. Details are captured with respect to the type of repayment. Deposit details for cash collections can also be done. |
| Credit Validation | At this stage, the MIS team will verify the Overdue amount and the amount collected by Loan Officer and also validates the amount with the bank statement. MIS team can approve and send for transaction authorisation if the Overdue amount > collection amount and MIS team can reject and send back to bounce stage with remarks. |
| Transaction Authorisation | The authoriser will login and waive off the bounce charges and penal interest and close. If they do not waive then the difference amount will be shown again in the bounce queue. |
| Deposit Stage | Loan officer will deposit the amount collected via cash mode |
| Bounce Queue for Hub Manager | All the bounced demands for which P2P details are captured by Loan officer gets listed. Hub manager may choose to send it for Agent Recovery |
| Bounce Queue for Agent Recovery | All demands classified for recovery by Agent by hub manager gets listed |

# Stage-role access

|  |  |
| --- | --- |
| **Stage** | **Applicable Role** |
| Bounce queue | Loan Officer |
| Promise to Pay | Loan Officer |
| Collection Payment | Loan Officer |
| Promise to Pay | Hub Manager |
| Collection queue | Recovery Officer |
| Promise to Pay | Recovery Officer |
| Credit Validation | MIS team |
| Transaction Authorisation | Finance Team |
| Deposit Stage | Loan Officer |
| Deposit Stage | Recovery Officer |

# Bounce Queue

## UI specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Section | Data Type | Attribute | Validations/ Restrictions | Population logic |
| Bounce Queue | Bounce Queue | Queue | Read only |  | Auto |
| Enterprise Name | Bounce Queue | String | Read only |  | Auto |
| Applicant | Bounce Queue | String | Read only |  | Auto |
| Co-applicant | Bounce Queue | String | Read only |  | Auto |
| Loan Account | Bounce Queue | String | Read-Write | - | Auto |
| Total amount due | Bounce Queue | Number | Read only |  | Auto |
| Principal | Bounce Queue | Number | Read only |  | Auto |
| Interest | Bounce Queue | Number | Read only |  | Auto |
| Penal interest | Bounce queue | Number | Read only |  | Auto |
| Charges | Bounce queue | Number | Read only |  | Auto |
| Fees | Bounce queue | Number | Read only |  | Auto |
| Overdue Days | Bounce Queue | String | Read-Write | - | Auto |
| Promise to pay date | Bounce Queue | Date | Read-Write | - | Auto /Can edit/can add |
| Promise to pay reasons | Bounce Queue | String | Read-Write | - | Auto /Can edit/can add |
| Repayment Amount | Bounce Queue | String | Read-Write | - | Auto |
| Type of repayment | Bounce Queue | Drop down | Read-Write | - | Auto |
| Collect Payment | Bounce Queue | String | Click | - | - |
| PTP date | Bounce Queue | Date | Click | - | - |

## Screenshot

Sample due amounts:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Loan account** | **Due** | **Principal** | **Interest due** | **Penal interest** | **Charges** | **fees** | **No.of due** |
| 508640101335 | 9754 | 7422.85 | 2131.15 | 80 | 100 | 20 |  |
| 508640101335 | 9794 | 7449.51 | 2104.49 | 120 | 100 | 20 |  |
| Total | **19548** | **14872.36** | **4235.64** | **200** | **200** | **40** | **2** |
| 508640108276 | 19397 | 14844.7 | 4262.3 | 150 | 100 | 40 | 1 |
| 508651508978 | 49816 | 37110.26 | 10655.74 | 1200 | 750 | 100 | 1 |

## Functional Requirements

**Actor: Loan officer**

1. Log-in screen for loan officer  
   Data fields: Login ID and Password
2. Option to click on Bounce Queue in order to view bounce Queue
3. Once Bounce Queue will be clicked, show loans which are there in the bounce queue for particular loan officer
4. In the bounce queue, each loan will be displayed with total repayment amount.
5. The loans in the bounce queue should have facility to **filter AREA WISE/SPOKE WISE/OVERDUE DAYS WISE/PTP DATE WISE**
6. In the same screen, provide an option to collect payment or navigate option to enter PTP date.
7. Bounce Queue data fields:

Loan Account:

Entity Name:

Overdue Days:

Promise to pay dates:

Promise to pay reasons:

Repayment Amount:

Type of repayment:

## Uploads

NA

## Downloads

NA

## Reports

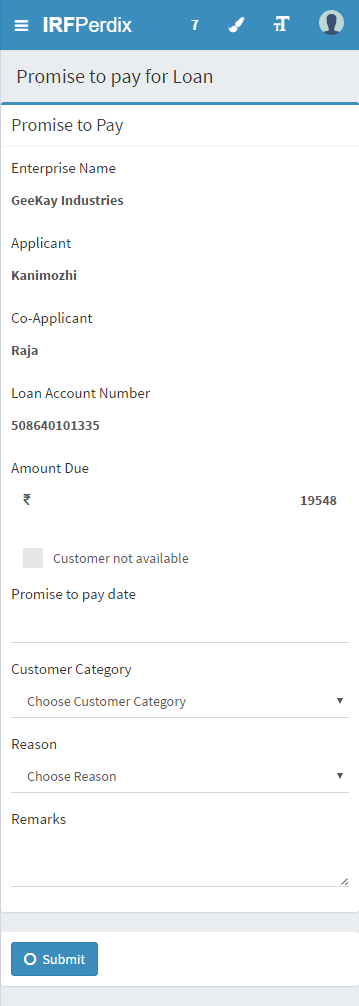
NA

# Promise to pay

## UI specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Section | Data Type | Attribute | Validations/ Restrictions | Population logic |
| Customer Available | Collection Queue | Check Box | Read-write | - | - |
| Type of customer | Collection Queue | Drop Down | Read-write | - | - |
| PTP Date | Collection Queue | Date | Read-Write | Should be future date | - |
| Remarks | Collection Queue | String | Read-Write | - | - |

## Screenshot



## Functional requirements

**Actor: Loan officer**

1. Select loan account from the bounce Queue and click on promise to pay
2. Option to check whether customer is available or not
3. Drop down box to categorise customer in A,B,C,D category

A-Under control

B-Tough

C-Difficult

D-Declared

1. Option to enter PTP date
2. Provide Text box to enter remarks for PTP

## Uploads

NA

## Downloads

NA

## Reports

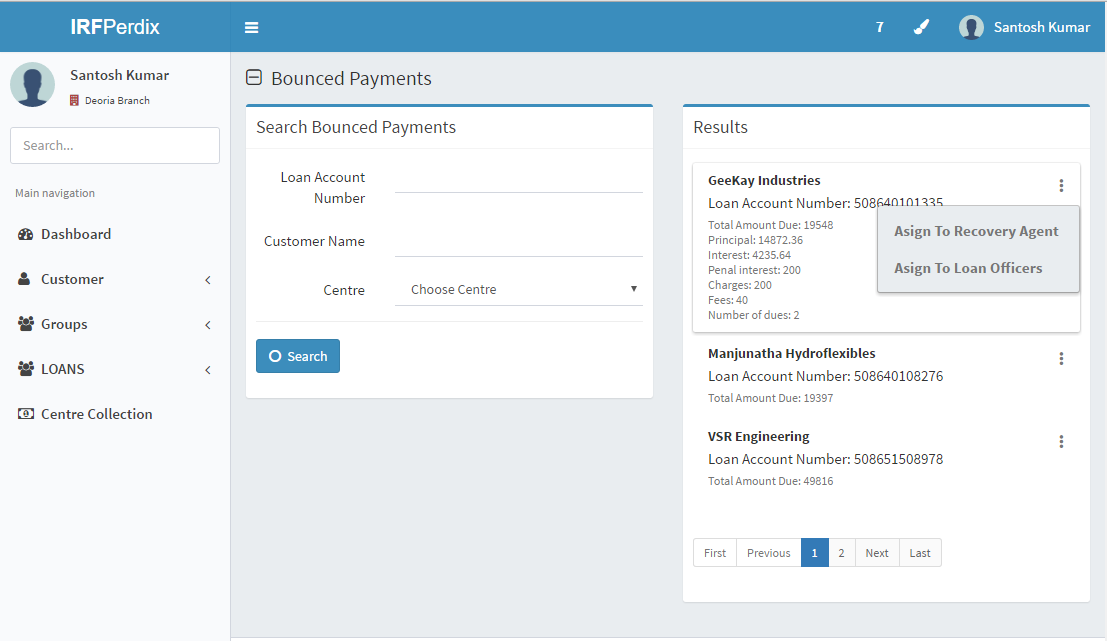
NA

# Collection Queue-Hub Manager

## UI specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Section | Data Type | Attribute | Validations/ Restrictions | Population logic |
| Loan Account | Collection Queue | String | Read-write | - | Auto |
| Customer Name | Collection Queue | String | Read-Write | - | Auto |
| Amount Overdue | Collection Queue | String | Read-Write | - | Auto |
| PTP Date | Collection Queue | Date | Read-Write | Should be future date/should be visible only when agent recovery option selected as NO | Auto populate/Can edit |
| Reasons | Collection Queue | String | Read-Write | - | Auto/can edit |
| Type of customer | Collection Queue | Drop down Box | Read-Write | - | Auto populate/can edit |
| Agent recovery | Collection Queue | Radio | Read-Write | - | Yes/No |

## Screenshot



## functional requirements

**Actor: Hub Manager**

1. Hub manager will login into the system
2. He will select collection queue
3. Multiple loan accounts will get displayed
4. To view/edit loan details, he can click on loan and edit
5. Fields are:  
   loan Account:

Customer Name:

Amount overdue:

PTP Date:

Reasons:

Type of customer:

1. Provide four options A,B,C and D to categorise customer-
   1. A-Under control
   2. B-Tough
   3. C-Difficult
   4. D-Declared
2. The reasons and remarks for hub manager will be different from loan officer and this will be reflected ,after updating, in loan officer’s screen
3. If Hub manager sends any loans for recovery then the recovery officer will login and see the demands marked to him.

## Uploads

NA

## Downloads

NA

## Reports

NA

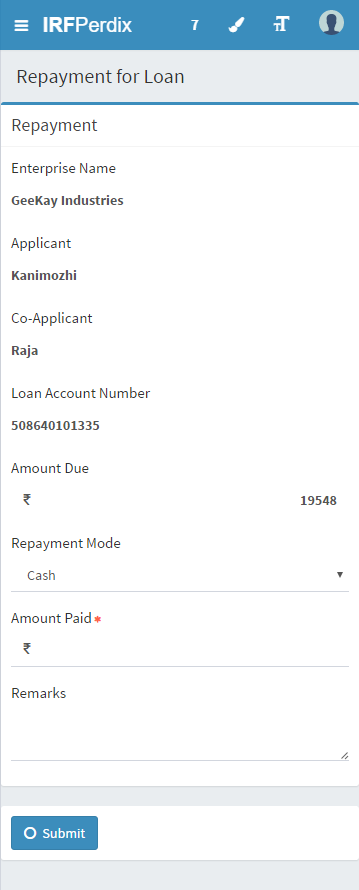
# Collect Payment

## UI specification

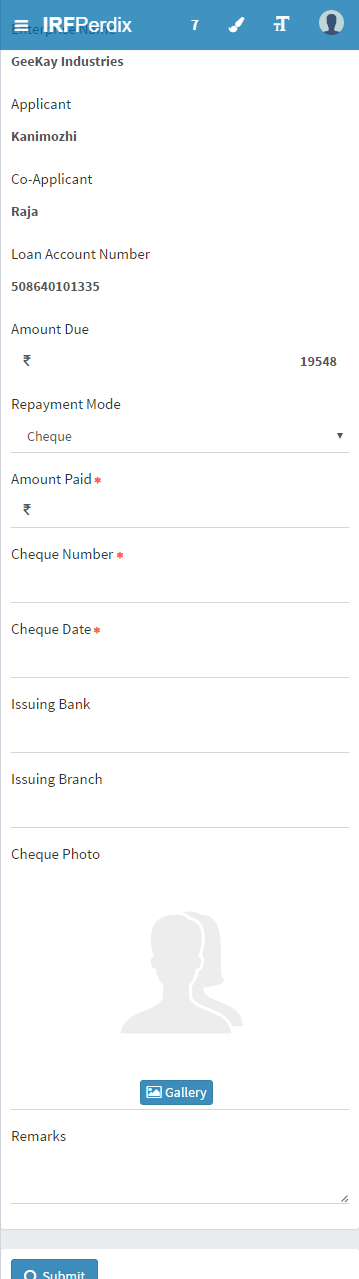
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Section | Data Type | Attribute | Validations/ Restrictions | Population logic |
| Amount Collected | Payment against demand | Numeric | Read-write | Numeric input only | - |
| Transaction Type | Payment against demand | Drop Down Box | Read-write | - | Cash/NEFT/Cheque |
| Cheque Number | Payment against demand | Alpha-numeric | Read-write | If cheque is selected as transaction type, then this field is mandatory | - |
| Bank Name | Payment against demand | Drop down | Read-Write | For cheque and NEFT | Auto-drop down |
| Branch Name | Payment against demand | Drop down | Read- Write |  | Auto-drop down |
| NEFT Reference number | Payment against demand | Alpha-numeric | Read-write | If NEFT is selected as transaction type, then this field is mandatory | - |
| Collected on | Payment against demand | Date | Read- write | Current date | Auto |
| Payment done on | Payment against demand | Date | Read write | Today or less than today | Default-today date |
| Remarks | Payment against demand | String | Read-write | - | - |
| Cheque Date | Payment against demand | Date | Read- Write | If cheque is selected, then cheque date is mandatory. |  |

## Screenshots

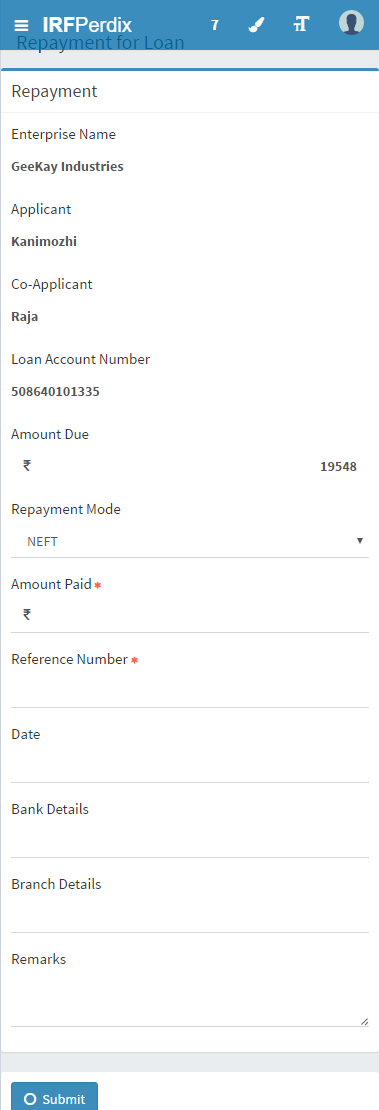
Cash Payment



Cheque Mode



NEFT Mode



## Functional requirements

**Loan Officer logs in**

1. Loan Officer will open bounce queue and select the customer for whom collection has to be done
2. Data fields: (for particular customer)

Amount collected:

Transaction type:

Cheque details/NEFT details:

Bank:

Branch:

Collected On(date):

Payment On(date):

Remarks:

1. Loan officer will keep on repeating the collection for all customers.
2. If type of repayment is cash then one more deposit stage will be created and addition details will be entered there.

Data fields:

Amount deposited:

Bank Name:

Account number:

1. Loan officer can update deposits at any point of time. For all customers, the amount will be added and loan officer can update the details in deposit cash stage

## Uploads

NA

## Downloads

NA

## Reports

1.Report showing the time taken for processes to be tracked

a.Time difference between the collected time by Loan Officer and the deposit time by Loan Officer.

b. Time difference between the collection date and cheque realisation date.

c. Time taken between the collected date by loan officer and the amount validation by the MIS team.

d.Transaction authorisation date and the payment realisation date.

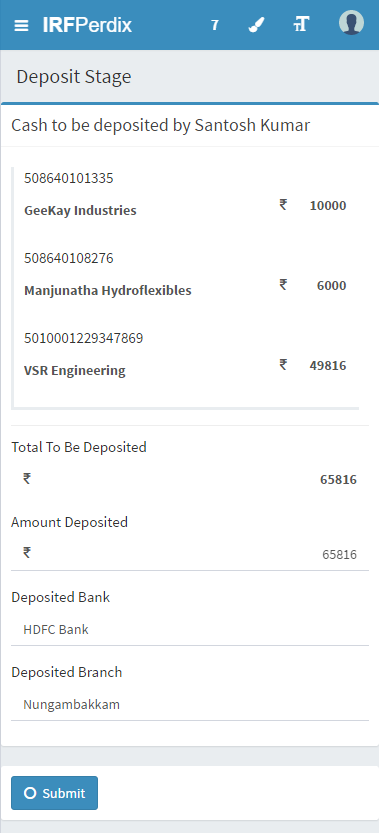
e.Trend showing patterns in the time delays.

# Deposit Stage Queue

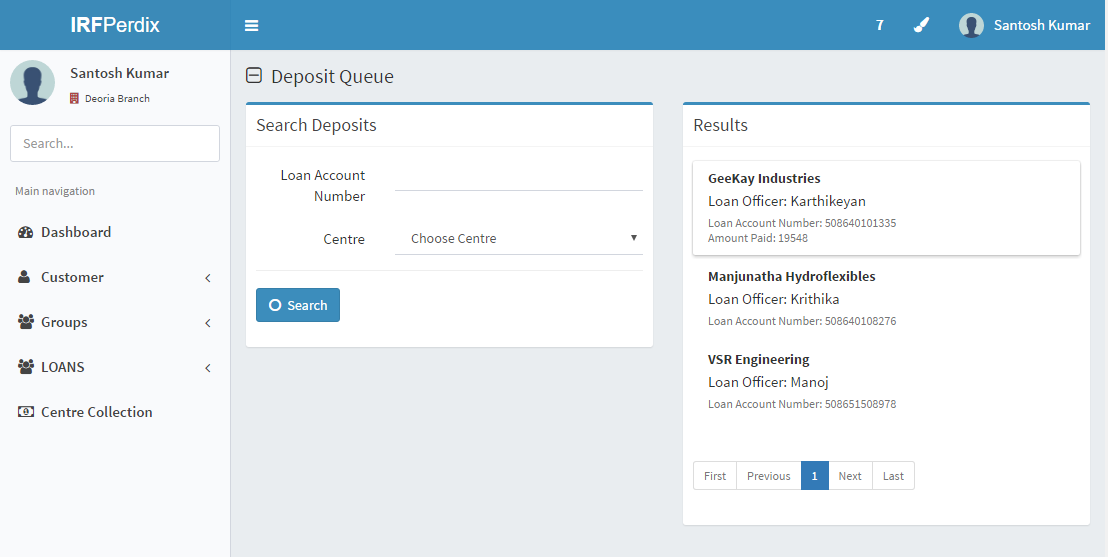
## UI specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Section | Data Type | Attribute | Validations/ Restrictions | Population logic |
| Amount for deposit | Payment against demand | Numeric | Read-Write | - | Auto |
| Deposit date | Payment against demand | Date | Read-Write | Current date | Auto-Current Date |
| Bank | Payment against demand | Drop Down Box | Read-Write | - | Auto-Drop Down |
| Branch | Payment against demand | Drop Down Box | Read-Write | - | Auto-Drop Down |

## Screenshot



Deposit Queue for Hub Manager:



## Functional requirements

**Loan Officer**

1. At the end of the day, Loan officer is supposed to validate amount collected in whole day
2. Enter details about bank and branch
3. Submit the details

## Uploads

NA

## Downloads

NA

## Reports

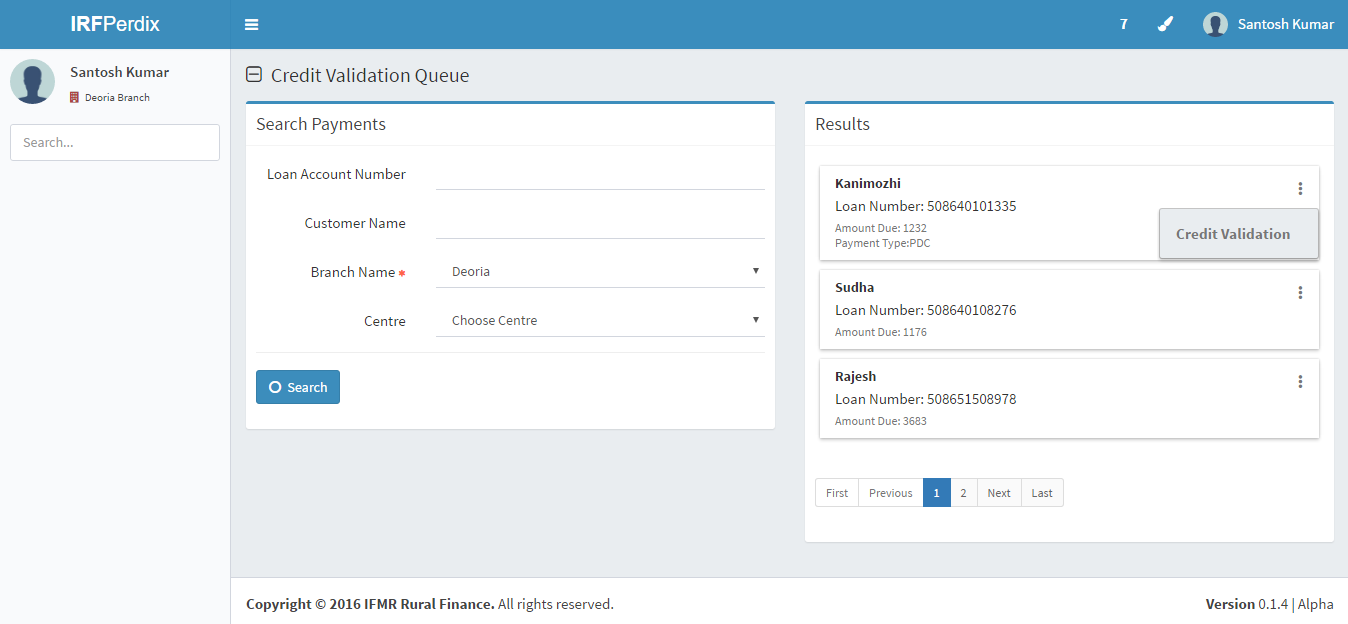
NA

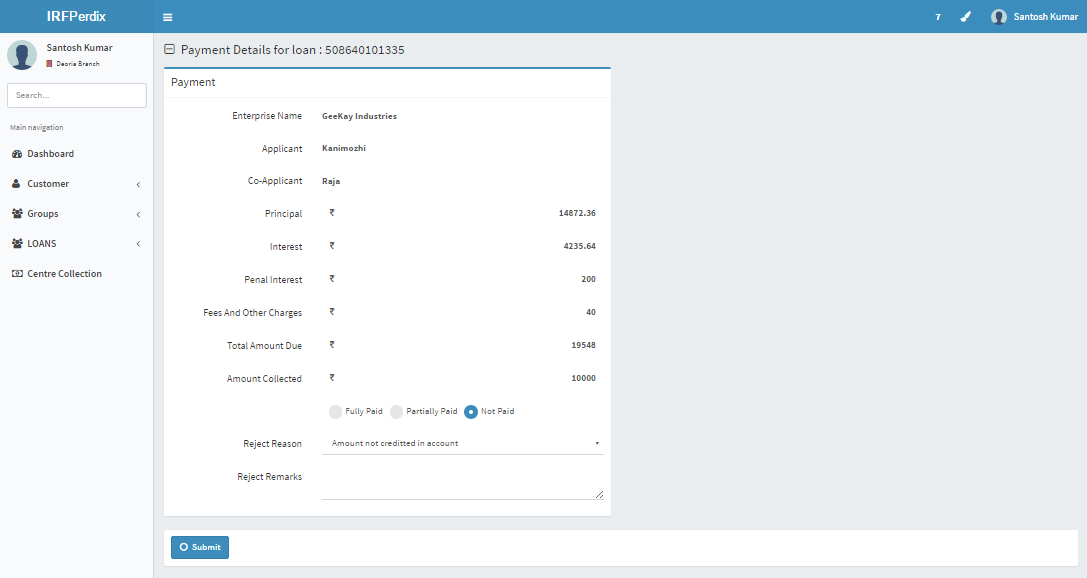
# Credit Validation

## UI specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Section | Data Type | Attribute | Validations/ Restrictions | Population logic |
| Validation Queue | Validation Queue | Queue | Read-Write | - | Auto |
| Loan Account | Validation Queue | String | Read-Write | - | Auto |
| Customer name | Validation Queue | String | Read-write | - | Auto |
| Amount to be collected | Validation Queue-(Amount to be collected) | String | Read-Write | - | Auto |
| Penal Interest | Amount to be collected | String | Read | - | Auto |
| Interest | Amount to be collected | String | Read | - | Auto |
| Fees | Amount to be collected | String | Read-Write |  | Auto |
| Total | Amount to be collected | String | Read-Write | - | Auto |
| Amount Collected | Validation Queue | String | Read-Write | - | Auto |
| Reasons | Reject | String/Drop Down Box | Read-Write | Only when Reject button pressed | - |
| Remarks | Reject | String | Read-Write | Only when Reject button pressed | - |

## Screenshot





## Functional requirements

**Head Office/MIS team logs in:**

1. Enter login id and password to view screen
2. View credit validation queue
3. All the loans for credit validation will be displayed
4. Click on the loan to view details  
   Data fields:  
   Loan Account:

Customer Name:

Amount to be collected:

Amount collected:

Verify/Reject: Submit type button

On clicking reject:

Reject reasons: Drop down box

Reject remarks:

1. Operations to verify the credit amount with the demand amount inclusive of all charges, penal interest
2. If amount is matching then trigger a SMS notification of payment approval to the customer and the demand is moved out of bounced queue
3. On rejection, loan will be sent to transaction authoriser
4. If credit is not affected in the system, Operations will route the task back to the field officer to verify and re-initiate this process. This can happen in case of Amount mentioned by Loan Officer and shown in bank statement mismatch or wrong details entered
5. Three validation cases
   1. If amount to be collected=Amount collected then Verifies the transaction and loan will be removed from credit validation queue
   2. If amount collected=0 then send back to loan officers with reasons
   3. If amount collected is in range of 0 to “amount to be collected” then send the loan to transaction authoriser

## Uploads

NA

## Downloads

NA

## Reports

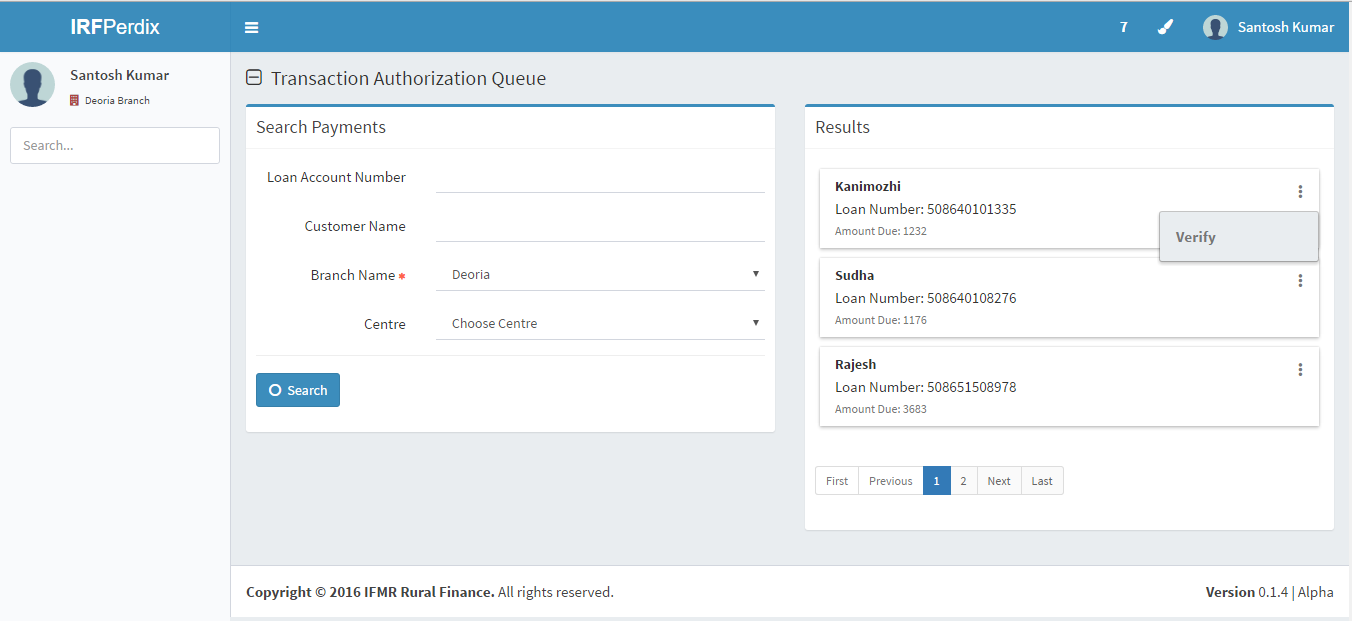
NA

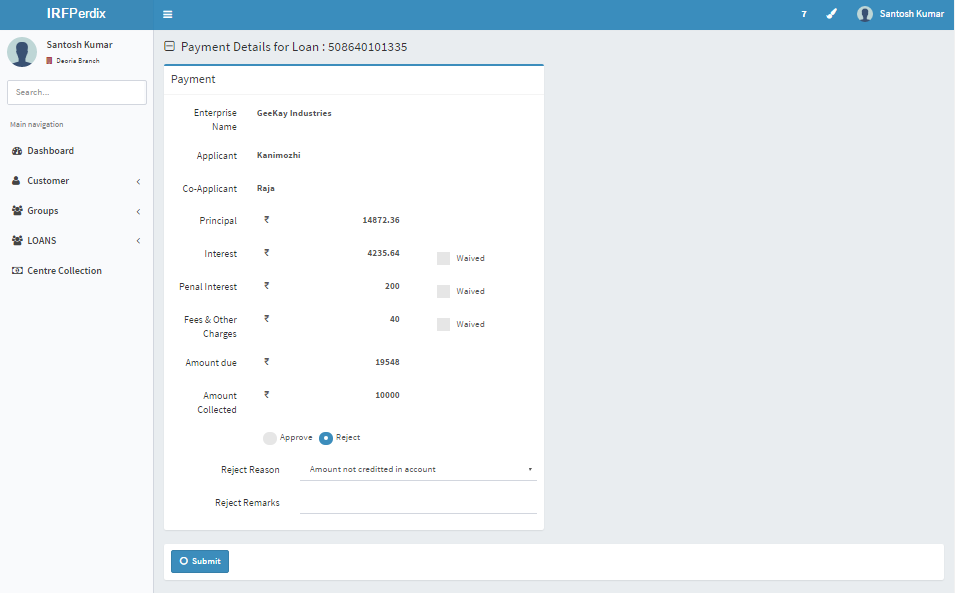
# Transaction Authorization

## UI specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Section** | **Data Type** | **Attribute** | **Validations/ Restrictions** | **Population logic** |
| Loan Account | Transaction Authorization Queue |  | Read-write | - | Auto |
| Customer Name | Transaction Authorization Queue |  | Read-write | - | Auto |
| Type of customer | Transaction Authorization Queue |  | Read-Write | - | Auto |
| Amount to be collected | Transaction Authorization Queue |  | Read-Write | - | Auto |
| Penal Interest | Transaction Authorization Queue |  | Read-Write | - | Auto |
| Interest | Transaction Authorization Queue |  | Read-Write | - | Auto |
| Fees | Transaction Authorization Queue |  | Read-Write | - | Auto |
| Amount collected | Transaction Authorization Queue |  | Read-Write | - | Auto |

## Screenshot





## Functional Requirements

Head Office/MIS team logs in:

1. All the details will be displayed in transaction authorization queue
2. Transaction authorizer verifies the credit amount and demand amount and decides to accept the credit by waiving the charges/penal interest
3. Provide an option to waive off penal interest, interest , fees completely/
4. Upon acceptance by Transaction authorizer, the payment is reversed and a fresh payment is initiated by waiving the charges/penal interest
5. Payment made against the demand and updated in the system.
6. SMS Notification sent to customer on the payment status

## Uploads

NA

## Downloads

NA

## Reports

NA