# 3. Specific Requirements

## 3.1 Functional Requirements.

### 3.1.1 User Registration

3.1.1.1 Introduction

Any user either borrower or investor have to register on the site to get loan or to invest.

There will be separate registration forms for borrowers and investors.

Different fields will be collected like personal, education, professional and financial details.

3.1.1.2 Inputs

The fields that are collected from users(both investors and borrowers) are as follows:

|  |  |
| --- | --- |
| id | int(11) |
| user\_code | varchar(255) |
| email | varchar(255) |
| password | varchar(255) |
| screen\_name | varchar(128) |
| mobile | varchar(10) |
| type | enum('Investor','Borrower','Guarantor') |
| employment\_type | enum('Salaried','Self-Employed','Salaried-Professionals', 'Self-Employed-Professionals','Others','Retired') |
| gender | enum('Male','Female') |
| date\_of\_birth | date |
| marital\_status | enum('Unmarried','Married','Divorced','Widow') |
| first\_name | varchar(255) |
| last\_name | varchar(255) |
| middle\_name | varchar(25) |
| father\_name | varchar(255) |
| spouse\_name | varchar(255) |
| education\_qualification\_id | int(11) |
| institute\_name | varchar(255) |
| graduated\_year | int(11) |
| residence\_type | enum('Own','Rent','Company Provided','PG') |
| present\_address | text |
| permanent\_address | text |
| permanent\_pincode | int(11) |
| present\_landmark | varchar(255) |
| permanent\_landmark | varchar(255) |
| city\_id | int(11) |
| state\_id | int(11) |
| country\_id | int(11) |
| present\_pincode | varchar(20) |
| home\_phone | varchar(100) |
| permanent\_mobile\_number | varchar(255) |
| permanent\_home\_phone | varchar(25) |
| permanent\_state\_id | int(11) |
| permanent\_city | varchar(250) |
| alternate\_email\_address | varchar(255) |
| num\_of\_years\_curr\_address | varchar(100) |
| num\_of\_months\_curr\_address | varchar(100) |
| people\_at\_permanent\_address | enum('Parents','Siblings','Spouse','Rented','Others') |
| pan\_number | varchar(100) |
| aadhar\_number | varchar(20) |
| ref\_relation1\_id | int(11) |
| ref\_details1 | varchar(250) |
| ref\_name1 | varchar(255) |
| ref\_address1 | text |
| ref\_phone1 | varchar(255) |
| ref\_mobile1 | varchar(255) |
| ref\_state1\_id | int(11) |
| ref\_city1\_id | varchar(250) |
| ref\_pincode1 | int(11) |
| ref\_landmark1 | varchar(250) |
| ref\_relation2\_id | int(11) |
| ref\_details2 | varchar(250) |
| ref\_name2 | varchar(255) |
| ref\_address2 | text |
| ref\_phone2 | varchar(255) |
| ref\_mobile2 | varchar(255) |
| ref\_state2\_id | int(11) |
| ref\_city2\_id | varchar(250) |
| ref\_pincode2 | int(11) |
| ref\_landmark2 | varchar(250) |
| company\_name | varchar(255) |
| company\_category\_id | int(11) |
| industry\_id | int(11) |
| business\_ownership\_type | enum('Lease','Owner','Pagadi','Parental','Relatives','Rental') |
| business\_incorporation\_date | date |
| employee\_id | varchar(20) |
| designation | varchar(255) |
| company\_website | varchar(200) |
| office\_address | varchar(255) |
| office\_city\_id | int(11) |
| office\_country\_id | int(11) |
| office\_state\_id | int(11) |
| office\_pincode | varchar(255) |
| office\_landmark | varchar(255) |
| office\_phone | varchar(100) |
| office\_mobile\_number | varchar(100) |
| office\_email\_address | varchar(255) |
| no\_of\_years\_work\_current\_company | int(11) |
| no\_of\_months\_work\_current\_company | int(11) |
| no\_of\_years\_work | int(11) |
| no\_of\_months\_work | int(11) |
| no\_of\_job\_change | int(11) |
| father\_employment | enum('Salaried','Self-Employed', 'Retired/Unemployed','Others') |
| father\_company | varchar(255) |
| father\_company\_address | text |
| father\_company\_state\_id | int(11) |
| father\_company\_city\_id | varchar(255) |
| father\_company\_pincode | int(11) |
| spouse\_employment | enum('Salaried','Self-Employed', 'Retired/Unemployed','Others') |
| spouse\_company | varchar(255) |
| spouse\_company\_address | text |
| spouse\_company\_state\_id | int(11) |
| spouse\_company\_city\_id | varchar(255) |
| spouse\_company\_pincode | int(11) |
| gross\_income | int(11) |
| monthly\_income | int(11) |
| house\_rent | int(11) |
| electricity | int(11) |
| household | int(11) |
| food\_groceries | int(11) |
| school\_fee | int(11) |
| medical\_expenses | int(11) |
| fuel\_expenses | int(11) |
| home\_loan\_emi | int(11) |
| other\_emi | int(11) |
| leisure\_entertinement | int(11) |
| others | int(11) |
| other\_investments | int(11) |
| have\_car | enum('Yes','No') |
| car\_ownership | enum('Self-Financed','Financed','Company Provided') |
| have\_two\_wheeler | enum('Yes','No') |
| two\_wheeler\_ownership | enum('Self-Financed','Financed','Company Provided') |
| have\_house | enum('Yes','No') |
| house\_ownership | enum('Self-Financed','Financed','Company Provided') |
| have\_land | enum('Yes','No') |
| land\_ownership | enum('Self-Financed','Financed') |
| assets\_owned | varchar(255) |
| twitter\_id | varchar(255) |
| show\_twitterid | enum('Yes','No') |
| facebook\_id | varchar(255) |
| show\_facebookid | enum('Yes','No') |
| linkedin\_id | varchar(255) |
| show\_linkedin\_id | enum('Yes','No') |
| referred\_by | varchar(15) |
| register\_status | enum('0','1','2','3','4','5') |
| created\_on | timestamp |
| final\_submission\_on | timestamp |

3.1.1.3 Order of registration

Borrowers:

Preregister – Email verification - Personal – Education - Reference – Professional – Financial – Other

Investors:

Preregister – Email verification - Personal – Professional – Financial – Other

3.1.1.4 Screen Shots

### 3.1.2 User Verification

3.1.2.1 Introduction

Once the user has completed his registration process, he has to go under verification.

Tele-agent will call him first for tele-verification. Once it is successful then that particular case will be forwarded to Verification-Agent else to Verification-Manager.

Verification-Agent will do manual verification of Residential address and other details, and then he will collect all the Documents required from the user. Once the verification is done then that particular case will be forwarded to Verification-Manager.

Verification-Manager will review the case and he will either approve or reject the case.

Once the user is approved then his listing will be published.

3.1.2.2 Inputs

The fields that are collected during verification process:

|  |  |
| --- | --- |
| verification\_started\_on | timestamp |
| verified\_on | timestamp |
| final\_approval\_on | timestamp |
| updated\_on | timestamp |
| verification\_state\_id | int(11) |
| rejection\_reason\_id | varchar(11) |
| verification\_status | enum('Pre-Register','Under Process','In Progress', 'Completed','Approved','Rejected') |
| verification\_agent\_id | int(11) |
| affiliate | varchar(100) |
| mailer | varchar(50) |
| status | enum('Active','Inactive','Deleted') |
| remarks | Text |

3.1.2.3 Order of verification

Tele Verification by tele-caller – Background Verification by verification-agent – Final Approval by Verification-manager

3.1.2.4 Screen Shots

### 3.1.3 Listing

3.1.3.1 Introduction

Once the user has been approved, his listing will be published (visible to all other users)

Investors have investor listing and Borrowers have borrowers listing.

Investors’ listing consists of amount willing to invest, returns expected, investment criteria, Groups, Q&A.

Borrowers’ listing consists of loan amount, duration, interest rate expected, list of guarantors (if any), borrowers’ profile, Groups, Q&A.

3.1.3.2 Inputs

The fields that are collected for Listing:

|  |  |
| --- | --- |
| id | int(11) |
| user\_id | int(11) |
| title\_id | int(11) |
| amount | int(11) |
| init\_amount | int(11) |
| interest | float |
| init\_interest | float |
| duration | int(3) |
| applied\_on | timestamp |

3.1.3.3 Screen Shots

### 3.1.4 Groups

3.1.4.1 Introduction

Any Registered user can create a group. Once the group is approved by Admin it will be visible to public.

The user who creates group will act as group leader.

Any Registered user can join Approved Group. Once the user has joined the group , the respected group leader has to approve that member.

Group leader can send requests to other approved members to join his group. The user who has received the request has to accept it to join that group.

3.1.4.2 Inputs

The fields that are collected for Group creation:

|  |  |
| --- | --- |
| id | int(11) |
| leader\_id | int(11) |
| group\_type\_id | int(11) |
| group\_membership | enum('Open','Invitation') |
| group\_name | varchar(255) |
| group\_subtitle | varchar(255) |
| group\_photo | varchar(255) |
| group\_description | text |
| group\_created\_on | timestamp |
| approved\_on | timestamp |
| group\_status | enum('Created','Approved','Rejected','Closed','Deleted') |

The fields that are collected for group join:

|  |  |
| --- | --- |
| id | int(11) |
| group\_id | int(11) |
| user\_id | int(11) |
| group\_relation\_id | int(11) |
| join\_status | enum('Requested','Invited','Approved','Rejected') |
| join\_message | text |
| join\_date | timestamp |
| approved\_on | timestamp |

3.1.4.3.1 Order of group creation

Create Group – Become group leader – Admin approval – Visible to public

3.1.4.3.2 Order of member joins in group

Browse Group – Join group – Group leader accept request

3.1.4.3.2 Order of group leader sends invitations

Select members – send invitations – members accept/reject invitation

3.1.4.4 Screen Shots

### 3.1.5 Q&A

3.1.5.1 Introduction

Once the Borrowers and Investors are approved and their listings are published they can interact with each other only through this section and the complete interaction is visible to all registered users.

This is a one-one interaction. Where one approved investor can ask questions to other approved borrowers and approved borrowers can give reply to the questions asked to them by the approved investors.

3.1.5.2 Inputs

The fields that are collected for Q&A:

|  |  |
| --- | --- |
| id | int(11) |
| lender\_id | int(11) |
| borrower\_id | int(11) |
| conversation | text |
| type | enum('Question','Answer') |
| created\_on | timestamp |
| status | enum('Active','Deleted') |
| deleted\_by | int(11) |
| is\_public | enum('Yes','No') |
| is\_read | enum('Yes','No') |

3.1.5.3 Order of Q&A

Investor asks Borrower Question – Borrower replies to investors question

3.1.5.4 Screen Shots

### 3.1.6 Bids

3.1.6.1 Introduction

This is one of the most important sections of the whole application.

Here the real transaction happens.

Approved borrowers can send request to approved investors. Investors can accept the request to invest or can update the request and accept it.

Or

Approved investors can send offers directly to approved borrowers.

Once request is accepted then it is converted to offer.

Investor can update/withdraw his offer any time.

Actions supported in this feature are: request(B-L), offer(L-B), Accept request, reject request, update request/offer, withdraw offer.

3.1.6.2 Inputs

The fields that are collected for Bid Process are:

|  |  |
| --- | --- |
| id | int(11) |
| borrower\_id | int(11) |
| lender\_id | int(11) |
| bid\_status\_id | int(11) |
| amount | int(11) |
| interest | float |
| updated\_on | timestamp |

3.1.6.3 Order of Bid

Request - Reject/accept/update - offer - update/withdraw

3.1.6.4 Screen Shots

### 3.1.7 Loan Closure

3.1.7.1 Introduction

Once the listing has funded sufficient as per ilend norms, borrower can close his listing.

Borrower has to select the bids which he wants to close and have to click on the accept loan.

Once the loan is accepted and listing is closed. He will not receive any offers or send requests to the investors.

Once the loan accepted, ilend will do the background process for loan disbursal.

3.1.7.2 Inputs

The fields that are collected for Loan closure are:

|  |  |
| --- | --- |
| accepted\_on | timestamp |
| approved\_on | timestamp |
| reactivated\_on | timestamp |
| verification\_agent\_id | int(11) |
| completed\_on | date |
| verification\_state\_id | int(11) |
| withdraw\_reason\_id | int(11) |
| withdraw\_reason | varchar(200) |
| status | enum('Open','Pending','Approved','Rejected', 'Completed','Withdrawn') |

3.1.7.3 Order of loan closure

Listing funded sufficiently – Select bids – accept loan

3.1.7.4 Screen Shots

### 3.1.8 Emi Generation

3.1.8.1 Introduction

Once the loan has been accepted by the borrower ilend starts its background loan disbursal process.

Once the process completed the ilend team will generate the emi table and payment date.

This emi schedule can be seen by both investors and borrowers in their account once it is generated.

3.1.8.2 Inputs

The fields that are collected for Emi generation are:

|  |  |
| --- | --- |
| id | int(11) |
| borrower\_listing\_id | int(11) |
| borrower\_id | int(11) |
| lender\_id | int(11) |
| emi\_no | varchar(11) |
| emi\_amount | float |
| additional\_amount | int(11) |
| principal | int(11) |
| interest | float |
| outstanding\_amount | float |

3.1.8.3 Order of Emi generation

Loan Accepted – Background process – Emi schedule generation

3.1.8.4 Screen Shots

### 3.1.9 Loan disbursal process

3.1.9.1 Introduction

Tele-confirmation (Informs the investors about loan closure and fix appointments with them to collect checks and signed agreements.)

Document Collection (meets the investors and Collects the checks, signed documents and verifies them.)

Once all the docs and checks have collected, i-lend fee is collected from the users, the loan is disbursed.

3.1.9.2 Order of loan disbursal process

Telecheck –Document collection – Loan approval – loan disbursal

3.1.9.3 Screen Shots

### 3.1.10 Emi repayment process

3.1.10.1 Introduction

Emi cheque deposit:

Tele-agent calls Borrowers and intimates the payment date.

Repayment-agent deposits the checks in the investors Account.

Emi cheque status:

Tele-agent calls investors and checks whether the emi has been deposited or not.

If not Cleared then case will be forwarded to recovery agent by Loan-manager

3.1.10.2 Inputs

The fields that are collected for Emi repayment process are:

|  |  |
| --- | --- |
| charges | float |
| paid\_amount | float |
| payment\_mode | varchar(25) |
| account\_no | varchar(25) |
| bank\_name | varchar(100) |
| branch | varchar(100) |
| check\_no | varchar(10) |
| investor\_account\_no | varchar(25) |
| investor\_bank\_name | varchar(100) |
| investor\_branch | varchar(100) |
| payment\_date | date |
| paid\_date | date |
| status | varchar(255) |

3.1.9.3 Order of Emi repayment process

Intimate borrowers – depositing cheques – Verifying with investors

3.1.9.4 Screen Shots