**User [ Borrower - Lenders]**

email (unique)

password

Full Name

mobile (unique)

profile picture

User\_ID

**Model** : User\_Table

**View :** logn in, signup , profile page , edit profit page , Search ,my network

**Controller:**

Sign up :

1. Check email is in valid format
2. Check password match [8 characters]
3. Terms and conditions Check box is checked
4. Registration button pressed
   1. Check if email already exist in user table
      1. If new : add data to the user table(name ,email , password )
      2. If found : alert user already exist (sing in - forget password option)
5. Confirmation/activation email sent to user email
6. return to sign in page to login

log in :

1. check if data (email , password ) already exist in user table and match in the same record
2. Load user dash board page
3. If No match , ask user to renter data , forget password option
   1. Send rest password email to user , user set a new password

profile page :

1. profile view , Retrieve data [name , profile picture] from user table with that logged in email and showing it
2. Retrieve User Rate -> average rate , Reviews received [ rates , reviews table]

Edit profit page:

1. Retrieve data from user table with that logged in email and showing it in a form format
2. User can editing / over right retrieved data [email , password ,Full Name,mobile full Address ,profile picture]
3. Save / update data in users table

Search :

1. enter email -mobile
2. return users data [name , profile pic, rate] that match

**Ratings and reviews**

**Model:**

users\_rate\_Table

Rate

User\_ID

users\_review\_Table

Review

User\_ID

(foreign key user id one to many]

(one user have many rate , many reviews )

**View:** profile page , loan request page , search

**Controller**

add rating( many to one )(many ratings for one user )

1. Accept input 1 to 5
2. insert it in rating table with profile User\_id

Add review( many to one )(many reviews for one user )

1. Accept text input
2. insert it in review table with profile User\_id

Average rate calculator (one to one) (user have 1 average rate )

1. Retrieve all record for current user , using User\_id
2. Calculate average rate from returned records rating

**Loan Request**

**Model:**

**Loans\_Tables**

User\_ID

Collection\_ID

Description

Amount

Installment period

Interest amount

Total amount to be paid

Installment Paybacks amount

(foreign key user id one to many, Collection\_ID one to one]

**View:** loan request form , loan calculations

**Controller**

Create loan

Insert Need , Amount , period ,Interest ,Total amount, Paybacks amount

**Cash Collection**

**Model:** **cash\_Collection**

Loan\_ID -> access borrower id

Lenders (user id)

amount

Status (notification - pending - confirmed)

(foreign keys Loan\_ID one to one]

**View:** dash board

**Controller**

cash notification [borrower]

list of amount and lenders name in which status is notification - pending

Can be accepted , if Accept ->change status to pending from notification

declined by borrower

Pending cash[lender]

Pending request shown on lender dashboard

Show borrower - lender Contact and , preferred payment method details

Both parties arrange a way to complete transaction [ payment method, contract ]

Realize cash

User confirms real cash received , change status to confirmed

Increase actual cash collected amount , decrease left amount

If amount collected close -stop lending option for that loan

transfer record from Cash Collection to transactions

Add record in transactions table with amount received ,lender id ,borrower id,date

Create a payback table for that transaction

**Notification system**

User\_id

**Payback deadlines**

payback amount (loan\_id)

Lender (User\_id)

Borrower(User\_id)

Deadline(date)

View dashboard

Show next payment amount and to whom

**My Network list**

User\_id

User\_id

Friendship (accept - decline)

Add to network option (friend request)

Subset of User

**For online payment and cash delivery**

**Wallet**

Wallet ID (foreign key user id , one to one)

Money [deposited + payback + rewards ]

**Model** Wallet\_table

**View** wallet view

**Controller**

Deduct money

1. Auto paid payback: check next payback day , if payback due 5 days notify user to deposit amount need if wallet > amount , if today = payback day deduct payback amount from wallet -> split it on lender -> deposit in lender wallet
2. Withdraw : enter amount , amount <= money , select desired method and fill in required details for it , deduct amount , notify admin to process it
3. Publishing fees

Add money

1. Deposit
2. Received payback
3. Rewards