

Dharmsinh Desai University, Nadiad
Faculty of Technology
Department of Computer Engineering



Project Report
On
ONLINE PAYMENT SYSTEM

B.Tech (CE) Sem-IV

Subject :- Software Project (CE-419)

Name:- Nakrani Jenit Anilbhai

Roll No:- CE074

ID:- 19CEUEG098

Name:- Padaliya Ronak Rasikbhai

Roll No:- CE080

ID:- 19CEUEG041

Under the Guidance of

Prof. Brijesh S. Bhatt

Prof. Pinkal Chauhan

Prof. Jigar Pandya

DHARMSINH DESAI UNIVERSITY

College Road, NADIAD-387001 (Gujarat)



Certificate

This is to certify that the term work carried out in the subject of **System Design Practice** and recorded in this report is bonafide work of **Mr. Nakrani Jenit Anilbhai (Roll No.: CE074, Id:- 19CEUEG098)** and **Mr. Padaliya Ronak Rasikbhai (Roll No.: CE080,Id: 19CEUEG041)** of **B.Tech Semester 4th** in the branch of Computer Engineering during the academic year 2020-21.

Prof. Brijesh S. Bhatt
(Project Guide and Associate Professor)
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

Dr. C.K. Bhensdadia
Head of CE Dept.,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad.

Index

<u>1. Abstract</u>	4
<u>2. Introduction</u>	5
<u>3. Software Requirement Specifications</u>	6
<u>4. Design Documents</u>	11
<u>5. Implements Details</u>	18
<u>6. Data Dictionary</u>	22
<u>7. WorkFlow/Layouts</u>	24
<u>8. Conclusion</u>	30
<u>9. Limitations and Future Extensions</u>	31
<u>10. Bibliography</u>	32

1. Abstract

Now a days Payment make through various digital platforms. Make Payment simple and secure using this type of digital platforms. If some unfair occur then we can block our bank account using this web application. Using this we not fear about thief which stolen our money. Now a days it is most popular to make payment.

In such type of application we can make payment through different ways, such as a using mobile number(linked with bank) , using account number , and using QR code scanning.

2. Introduction

1.1 Brief Introduction

Online Payment System , this system is useful for user which have an bank account and want to make payment through digital payment system. In this user have two option to make payment ,(1)through mobile number , (2)through account number.User of this system can make payment using UPI PIN which is necessary for make payment. Safety critical issues in this system to make less possible as make.

To use this system to make payment, make sure that user enter mobile number at time of create account in this is linked with his/her bank account. Without it system not allow user to make payment.

1.2 Technologies/Tools/Platforms used :

→ Django:

Django is a open source python web framework used for rapid development, maintainable, pragmatic ,secure and clean design.

→ Git:

This is platform is used were more than one user can develop Project and share using concurrently.

→ Virtual Environment :

For easy management and migration of the project is used. Virtual Environment provides developer all tools and packages necessary for the development without affecting current operating system configuration.

3. Software Requirement Specifications

Online Payment System

Users:-

- (1) **Customer**:- Which is use this system for making a payments.
- (2) **Admin**:- Which is maintain databases,users and functionality to make system flexible.

Functional Requirements:-

1. Manage Customers

R 1.1 Add Customer(Sign up):-

Description:- If user need to make payment then need to create account

Input:- Enter name,email,mobile number,PIN for login

Output:- Get OTP,after enter OTP make profile

R1.2 Login:-

Description:- User can login for make payments.

Input:- Enter mobile number,PIN

Output:- Login successful

R 1.3 Remove customer(Delete account):-

Description:- If user no long as use this then user can delete his/her account.

Input:- User selection

Output:- Remove account

R 1.4 Logout:-

Description:- User can exit from the system.

Input:- User selection

Output:- logout successful

2. Manage bank account

R2.1 Add bank account:-

Description:- For make payment user need to add bank account and generate UPI PIN for make payments afterwards.

Input:- Enter bank type, Account number, IFSC code, UPI PIN

Output:- Get OTP

R2.2 Change bank account:-

Description:- User can change bank account

Input:- Enter bank type, Account number, IFSC code, UPI PIN

Output:- Get OTP

R2.3 Remove bank account:-

Description:- User remove bank account also.

Input:- Enter UPI PIN

Output:- Remove account successful

3. Manage Payments

R3.1 Make payment

R3.1.1 Using mobile no.:-

Description:- For making payment use mobile number which is linked to bank account

Input:- Enter mobile number, UPI PIN

Output:- Payment status

R3.1.2 Using QR code:-

Description:- For make payment use QR code

Input:- Scan QR code and enter UPI PIN

Output:- Payment status

R3.1.3 Using account number and IFSC code

Description:- For make payment use account number and IFSC code

Input:- Enter account number, IFSC code, UPI PIN

Output:- Payment status

R3.2 Request for payment:-

Description:- User can request for other to make payment

Input:- Enter mobile number

Output:- Send message which is request for payment

R3.3 Cancel payment:-

Description:- User can cancel payment

Input:- User selection

Output:- Cancel payment

4. Manage Different Functionality

R4.1 Change PIN / UPI PIN:-

Description:- Change PIN which is used when login time and also change UPI PIN which used at make payment.

Input:- Enter mobile number

Output:- Get OTP and Change PIN

R4.2 Change profile:-

Description:- Change name, email, mobile number.

Input:- User selection

Output:- Change profile successfully

R4.3 Check balance:-

Description:- Show current balance in the bank account.

Input:- User selection

Output:- Show bank balance

R4.4 Check account status:-

Description:- Check account status for show that it is BLOCK or UNBLOCK

Input:- User selection

Output:- Show account status

5. Manage statistics

R5.1 Show past payments details:-

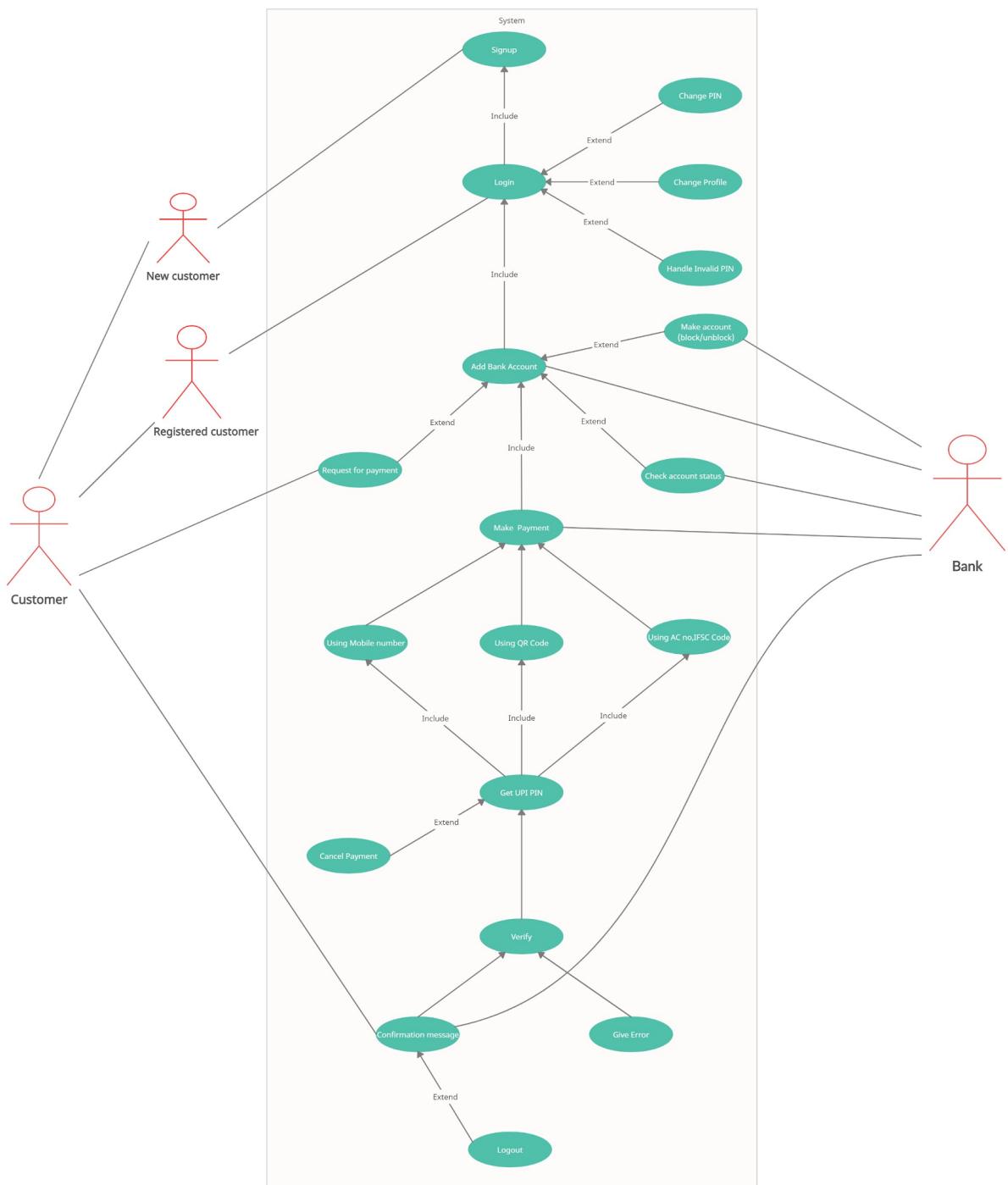
Description:- Show all payments which occurs in a past.

Input:- User selection

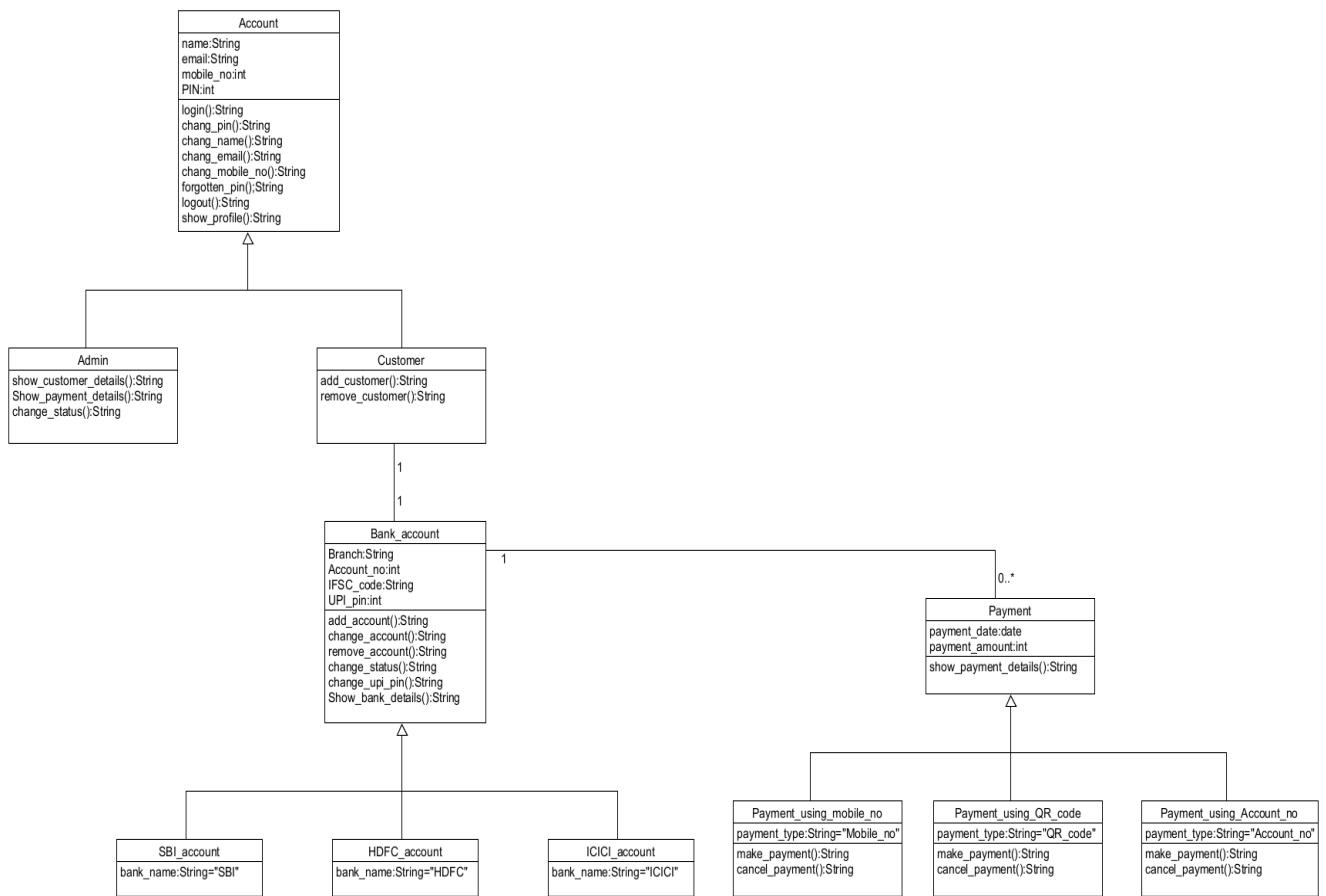
Output:- Show all payment which is done

4. Design Documents

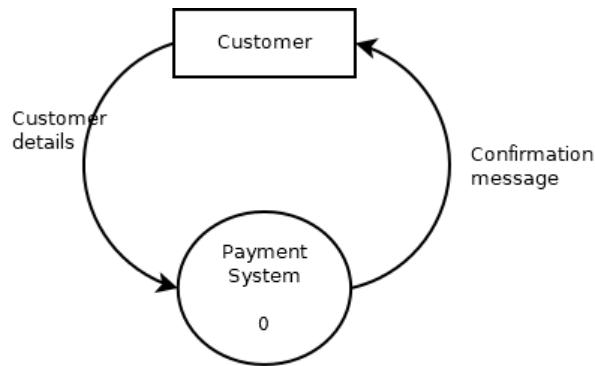
(1) Use Case Diagram :-



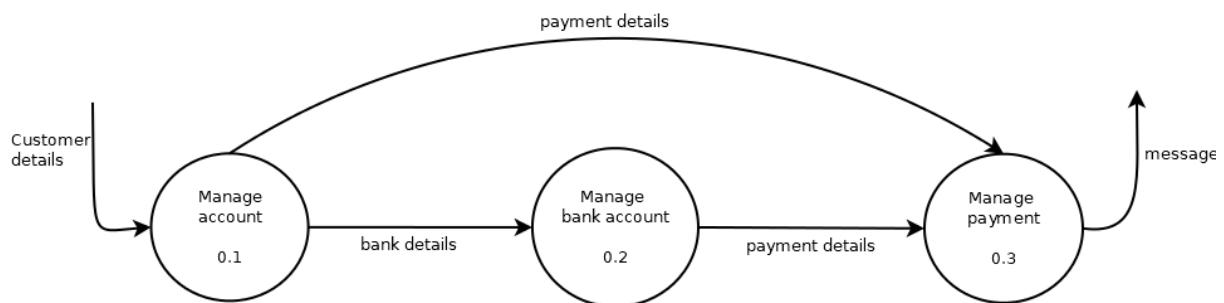
(2) Class Diagram:-



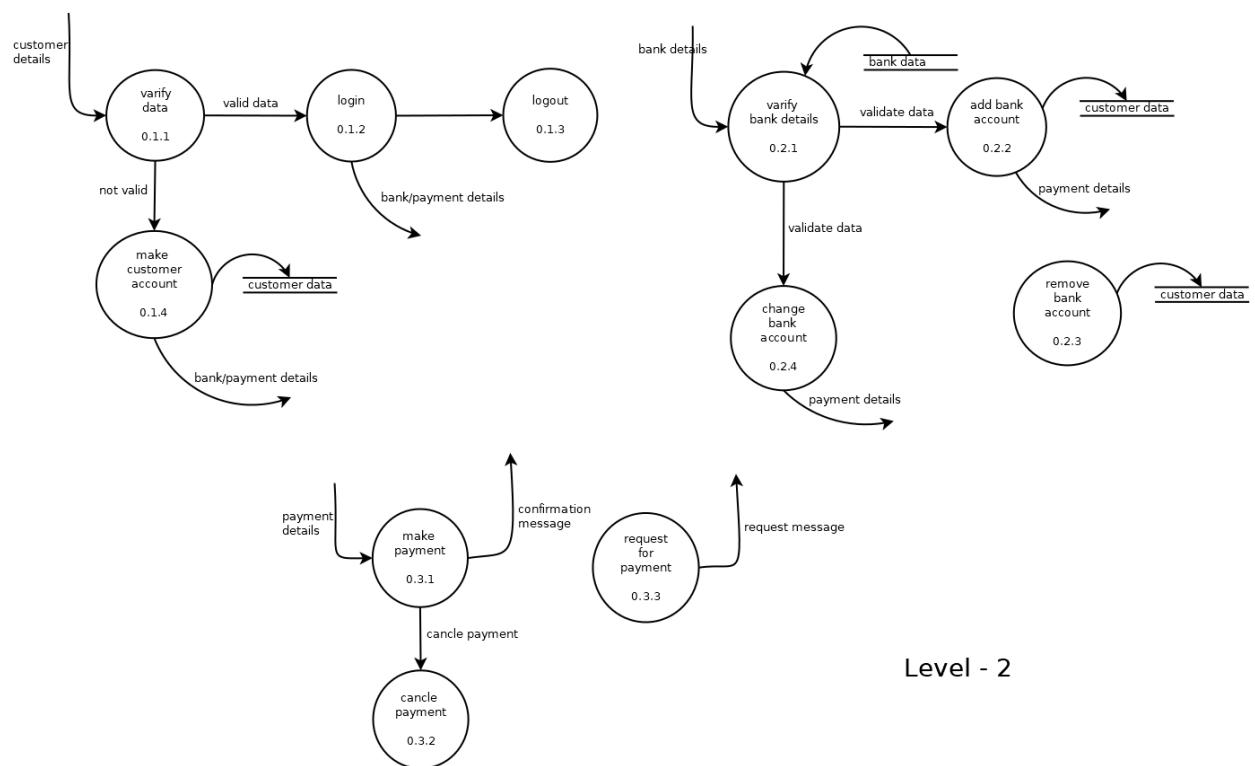
(3) Data Flow Diagram:-

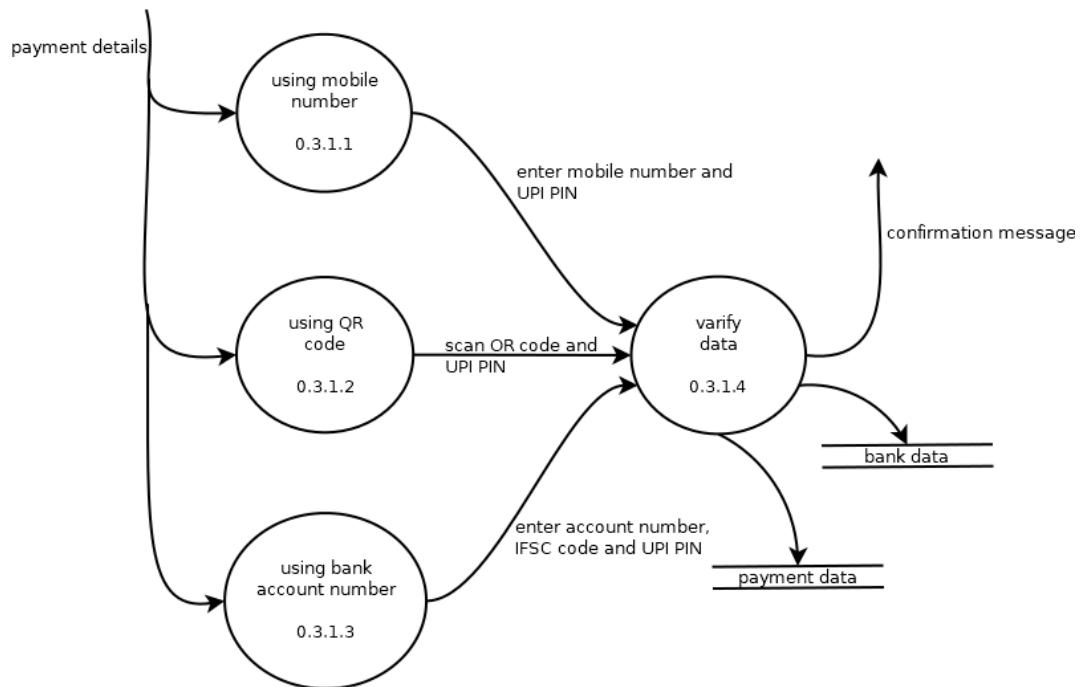


Level - 0



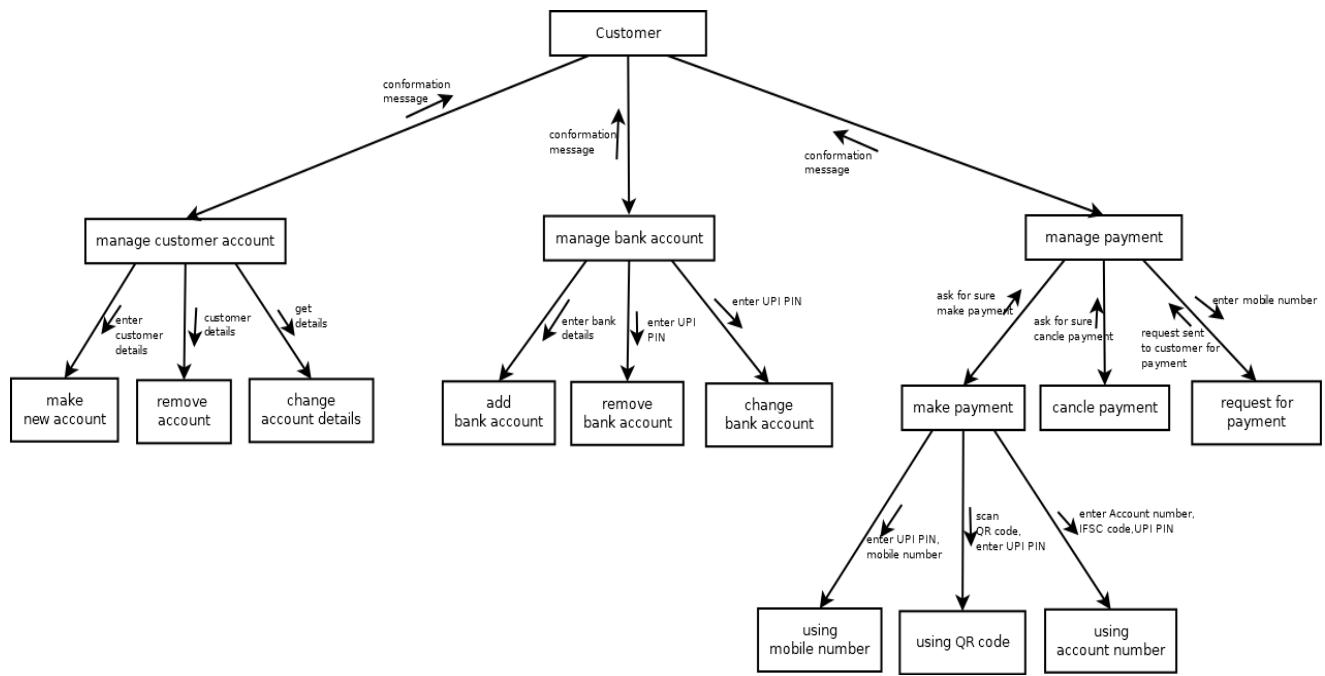
Level - 1





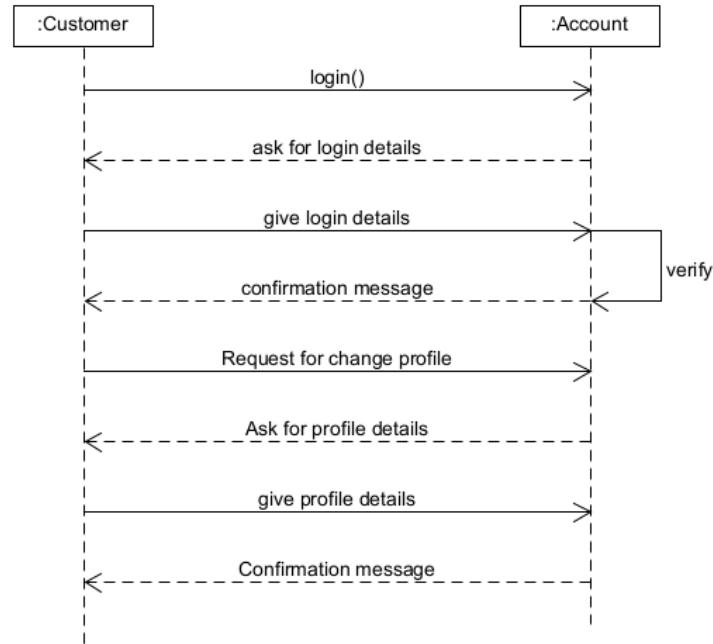
Level - 3

(4) Structure Chart:-

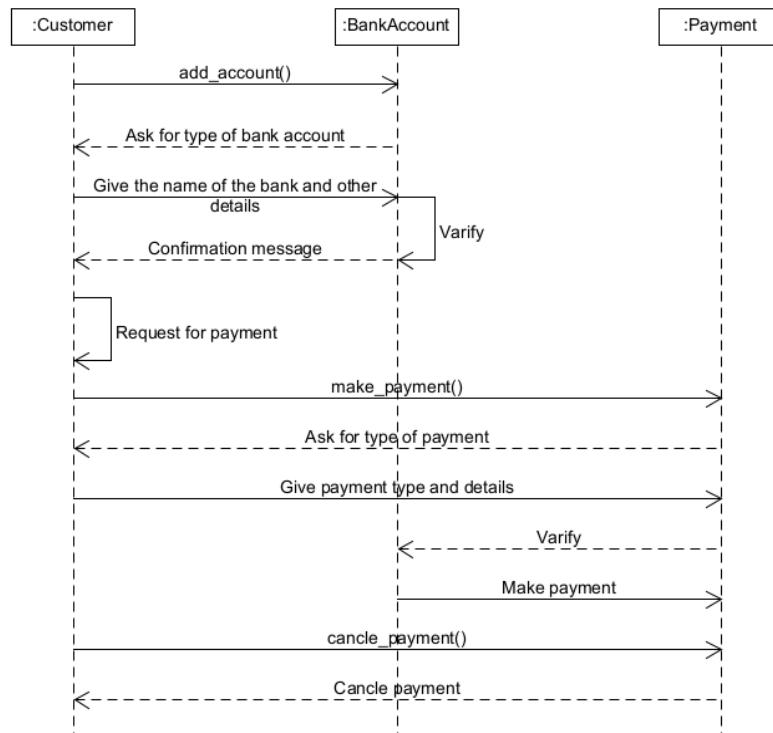


(5) Sequence Diagram:-

→ Login

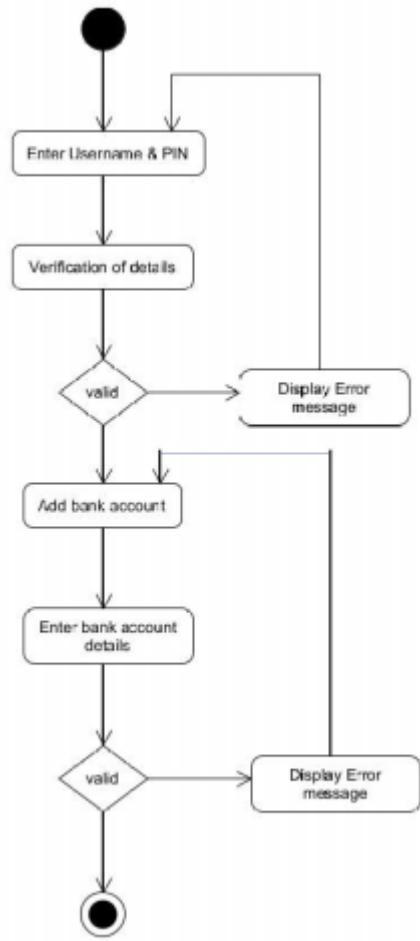


→ Make Payment

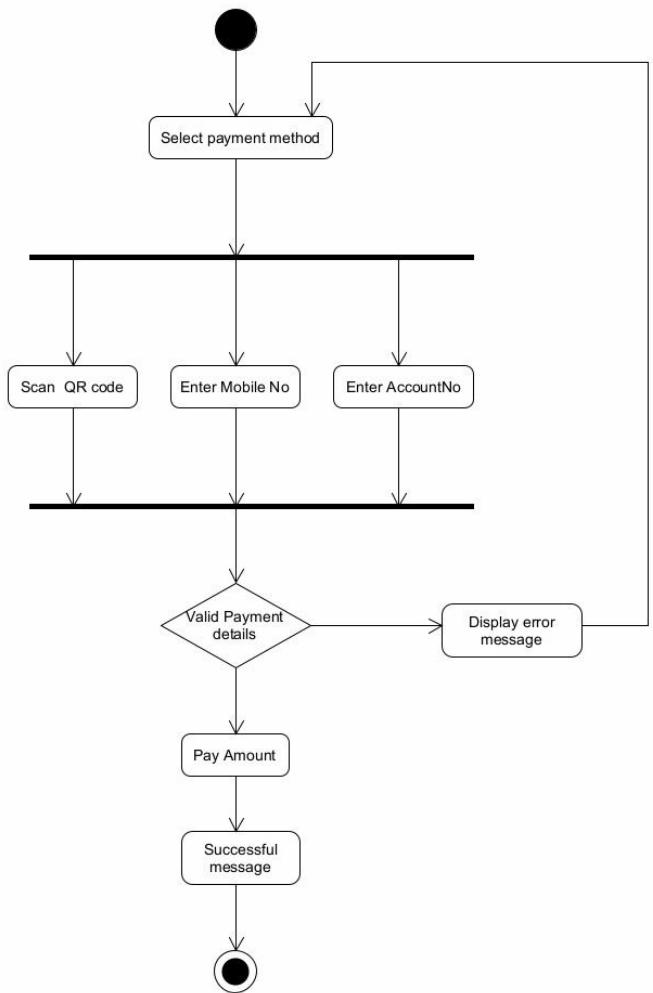


(6) Activity Diagram:-

Add bank account



Make Payment



5. Implements Details

4.1 Description of module:-

Customer Module:-

In this module we manage basic functionality which is used in every type of system. Like sign up, login, forgot password, remove bank account, logout, etc...

Bank Module:-

In this module manage bank account.some operation on bank account is like add bank account, change bank account, remove bank account, black/unblock account, etc...

Payment Module:-

In this manage payments using mobile number and account number. We cancel payment also.

4.2 Major Functionality:-

Registration:-

```

14 def registration(request):
15     if request.method == "POST":
16         username = request.POST['username']
17         password = request.POST['password']
18         password2 = request.POST['password2']
19         email = request.POST['email']
20         first_name = request.POST['firstname']
21         last_name = request.POST['lastname']
22         mobile=request.POST['mobile']
23         # check the data is entered or not by the user
24         if username != '' and password != '' and password2 != '' and email != '' and first_name != '' and last_name != '':
25             # check password and repassword are same or not
26             if password == password2:
27                 if Users_info.objects.filter(username=username).exists():
28                     messages.error(request,"User is already Exists!!!")
29                     return render(request,'Login/registration.html')
30                 elif Users_info.objects.filter(email=email).exists():
31                     messages.error(request,"Email is already Exists!!!")
32                     return render(request,'Login/registration.html')
33                 elif Users_info.objects.filter(mobile=mobile).exists():
34                     messages.error(request,"Mobile NO is already Exists!!!")
35                     return render(request,'Login/registration.html')
36                 else:
37                     u = Users_info(username=username, password=password, email=email, first_name=first_name, last_name=last_name, mobile=mobile)
38                     u.save()
39                     messages.success(request,"Successfully Created account")
40                     return redirect(' /Login ')
41                 # if passwords are not match then again render register page
42             else:
43                 messages.error(request, "password does not match")
44                 return redirect('registration')
45         else:
46             messages.error(request, "please enter the data")
47             messages.error(request, "all fields are must required")
48             return render(request, 'Login/registration.html')
49     else:
50         return render(request, 'Login/registration.html')
```

Add Bank Account:-

```

9 def bank_form(request):
10     if request.method=="POST":
11         mobile=request.POST['mobile']
12         acno=request.POST['acno']
13         check=request.session['user_username']
14         if mobile == Users_info.objects.filter(username=check).first().mobile:
15             if Bank.objects.filter(acno=acno).exists():
16                 if mobile == Bank.objects.filter(acno=acno).first().mobile:
17                     if len(list(Bank.objects.raw(f'SELECT * FROM addbankaccount_bank where mobile={mobile}')))!=1:
18                         b = Bank.objects.filter(acno=acno).first()
19                         b.status=1
20                         u = Users_info.objects.filter(username=check).first()
21                         u.status=1
22                         b.username = Users_info.objects.get(username =check)
23                         u.save()
24                         b.save()
25                         request.session['user_status']=1
26                         request.session['bank_status']=1
27                         request.session['bank_balance']=b.balance
28                         #request.session["bank_block"]=b.block
29                         messages.success(request,"Sucessfully connected")
30                         return redirect('http://127.0.0.1:8000/upi')
31                     else:
32                         messages.error(request,"Your mobile nummber is linked with mutiple account")
33                         return render(request,'addbankaccount/bank_details.html')
34                 else:
35                     messages.error(request,"Your mobile number is not linked with Bank")
36                     return render(request,"addbankaccount/bank_details.html")
37             else:
38                 messages.error(request,"Invalid account number")
39                 return render(request,"addbankaccount/bank_details.html")
40         else:
41             messages.error(request,"You have to change your mobile number")
42             return render(request,"addbankaccount/bank_details.html")
43     else:
44         return render(request,"addbankaccount/bank_details.html")

```

Make payment using mobile number:-

```

74 def mobile_no(request):
75     if request.method == "POST":
76         acno=request.POST['acno']
77         amount=request.POST['amount']
78         purpose=request.POST['purpose']
79         receiver_mobile_no=request.POST['receiver_mobile_no']
80         upi=request.POST['upi']
81         username=request.session['user_username']
82         # receiver_ac_no=int(receiver_mobile_no)
83         upi=str(upi)
84         amount=int(amount)
85         if Bank.objects.filter(mobile=receiver_mobile_no).first().block is None:
86             if acno == Bank.objects.filter(username=username).first().acno:
87                 if receiver_mobile_no == Bank.objects.filter(mobile=receiver_mobile_no).first().mobile:
88                     if Bank.objects.filter(mobile=receiver_mobile_no).first().username is not None:
89                         if upi == Users_info.objects.filter(username=username).first().upi:
90                             if amount <= Bank.objects.filter(username=username).first().balance-100:
91                                 current_b = Bank.objects.filter(username=username).first()
92                                 current_b.balance = current_b.balance - amount
93                                 request.session['bank_balance']=current_b.balance
94                                 receiver_b = Bank.objects.filter(mobile=receiver_mobile_no).first()
95                                 receiver_b.balance = receiver_b.balance + amount
96                                 current_b.save()
97                                 receiver_b.save()
98                                 t = Transaction()
99                                 t.sender_name = u.first_name
100                                t.sender_mobile = u.mobile
101                                t.sender_acno = acno
102                                t.sender_amount = amount
103                                t.sender_purpose = purpose
104                                t.sender_username=username
105                                t.receiver_name = Bank.objects.filter(mobile=receiver_mobile_no).first().first_name
106                                t.receiver_mobile = receiver_mobile_no
107                                t.receiver_acno = Bank.objects.filter(mobile=receiver_mobile_no).first().acno
108                                t.payment_type = "mobilen"
109                                t.save()
110                                messages.success(request, "Successfully payment Done")
111                                return redirect('http://127.0.0.1:8000/login/welcome/')
112                            else:
113                                messages.error(request, "Your balance is too low")
114                                return render(request, 'make_payment/mobile_no.html')
115                        else:
116                            messages.error(request, "upi is invalid")
117                            return render(request, 'make_payment/mobile_no.html')
118                    else:
119                        messages.error(request, "User is not registered")
120                        return render(request,'make_payment/mobile_no.html')
121                else:
122                    messages.error(request, "receiver mobile no is invalid")
123                    return render(request, 'make_payment/mobile_no.html')
124            else:
125                messages.error(request, "your acno is invalid")
126                return render(request, 'make_payment/mobile_no.html')
127        else:
128            messages.error(request,"Receiver account is blocked. So you can not make payment.")
129            return render(request,'make_payment/mobile_no.html')
130    else:
131        return render(request, 'make_payment/mobile_no.html')

```

Make payment using account number:-

```

17 def acno(request):
18     if request.method == "POST":
19         acno=request.POST['acno']
20         amount=request.POST['amount']
21         purpose=request.POST['purpose']
22         receiver_ac_no=request.POST['receiver_ac_no']
23         upi=request.POST['upi']
24         username=request.session['user_username']
25         receiver_ac_no=int(receiver_ac_no)
26         upi=str(upi)
27         amount=int(amount)
28         if Bank.objects.filter(username=username).first().block is None:
29             if acno == Bank.objects.filter(username=username).first().acno:
30                 if receiver_ac_no == int(Bank.objects.filter(acno=receiver_ac_no).first().acno):
31                     if upi == Users_info.objects.filter(username=username).first().upi:
32                         if amount <= Bank.objects.filter(username=username).first().balance-100:
33                             current_b=Bank.objects.filter(username=username).first()
34                             current_b.balance=current_b.balance-amount
35                             request.session['bank_balance']=current_b.balance
36                             receiver_b=Bank.objects.filter(acno=receiver_ac_no).first()
37                             receiver_b.balance=receiver_b.balance+amount
38                             receiver_b.save()
39                             receiver_b.save()
40                             t=Transaction()
41                             u=Users_info.objects.filter(username=username).first()
42                             t.sender_name=u.first_name
43                             t.sender_mobile=u.mobile
44                             t.sender_acno=acno
45                             t.sender_amount=amount
46                             t.sender_purpose=purpose
47                             t.sender_username=username
48                             t.receiver_name=Bank.objects.filter(acno=receiver_ac_no).first().first_name
49                             t.receiver_mobile=Bank.objects.filter(acno=receiver_ac_no).first().mobile
50                             t.receiver_acno=receiver_ac_no
51                             t.payment_type="acno"
52                             t.save()
53                             messages.success(request,"Successfully payment Done")
54                             return redirect('http://127.0.0.1:8000/Login/welcome')
55                         else:
56                             messages.error(request,"Your balance is too low")
57                             return render(request, 'make_payment/acno.html')
58                     else:
59                         messages.error(request,"upi is invalid")
60                         return render(request,'make_payment/acno.html')
61                 else:
62                     messages.error(request,"receiver account number is invalid")
63                     return render(request,'make_payment/acno.html')
64             else:
65                 messages.error(request,"your account number is invalid")
66                 return render(request,'make_payment/acno.html')
67         else:
68             messages.error(request,"Receiver account is blocked. So you can not make payment.")
69             return render(request,'make_payment/acno.html')
70     else:
71         return render(request,'make_payment/acno.html')
72

```

Block bank account:-

```

158 def block_bank_account(request):
159     if request.session.get('user_status') is not None:
160         if request.session.get('bank_block') is None:
161             if request.method=="POST":
162                 mobile=request.POST['mobile']
163                 acno=request.POST['acno']
164                 password=request.POST['password']
165                 check_username=request.session['user_username']
166                 if mobile == Users_info.objects.filter(username=check_username).first().mobile:
167                     if acno == Bank.objects.filter(acno=acno).first().acno:
168                         if password == Users_info.objects.filter(username=check_username).first().password:
169                             b = Bank.objects.filter(acno=acno).first()
170                             b.block=1
171                             b.save()
172                             request.session['bank_block']=1
173                             messages.success(request,"Successfully Bank account Blocked")
174                             return render(request,"Login/welcome.html")
175                         else:
176                             messages.error(request,"Password is Incorrect")
177                             return render(request,"change/block_bank_account.html")
178                     else:
179                         messages.error(request,"Account number is invalid")
180                         return render(request,"change/block_bank_account.html")
181                 else:
182                     messages.error(request,"Mobile number is Invalid")
183                     return render(request,"change/block_bank_account.html")
184             else:
185                 return render(request,"change/block_bank_account.html")
186         else:
187             return render(request,"change/block_bank_account_allready_block.html")
188     else:
189         return render(request,'change/block_bank_account_not_add_bank_account.html')
190

```

Delete User Account:-

```

30 def deleteAccount(request):
31     if request.method=="POST":
32         username=request.POST['username']
33         mobile=request.POST['mobile']
34         password=request.POST['password']
35         uname=request.session['user_username']
36         if username == uname:
37             u=Users_info.objects.filter(username=uname).first()
38             if mobile == u.mobile:
39                 if password == u.password:
40                     if request.session.get("user_status") is not None:
41                         ac=Bank.objects.filter(username=uname).first().acno
42                         b=Bank.objects.filter(acno=ac).first()
43                         b.username=None
44                         b.status=None
45                         b.save()
46
47                     #if u.status is not None:
48                     #    del request.session['bank_block']
49                     u.delete()
50                     while Transaction.objects.filter(sender_username=uname).first() is not None:
51                         t=Transaction.objects.filter(sender_username=uname).first()
52                         t.delete()
53                     del request.session['user_username']
54                     del request.session['user_password']
55                     del request.session['user_mobile']
56                     del request.session['user_status']
57                     del request.session['user_upi']
58                     del request.session['user_upi_staus']
59                     del request.session['user_email']
60
61                     messages.info(request,"Successfully deleted")
62                     return redirect('.../Login')
63                 else:
64                     messages.error(request,"Password is Invalid !")
65                     return render(request,'Profile/deleteAccount.html')
66             else:
67                 messages.error(request,"Mobile Number is Invalid !")
68                 return render(request,'Profile/deleteAccount.html')
69         else:
70             messages.error(request,"Username is Invalid !")
71             return render(request,'Profile/deleteAccount.html')
72     else:
73         return render(request,'Profile/deleteAccount.html')

```

6. Data Dictionary

login_users_info

Name field	Data type	Length	Description
first_name	varchar	100	
last_name	varchar	100	
email	varchar	50	
mobile	varchar	10	
username	varchar	50	Primary key
password	varchar	500	
status	varchar	10	default=NULL
upi	varchar	10	default=NULL
upi_staus	varchar	10	default=NULL

Addbankaccount

Name field	Data type	Length	Description
acno	varchar	14	Primary key
first_name	varchar	100	
last_name	varchar	100	
mobile	varchar	10	
balance	int	11	
status	int	11	default=NULL
username_id	varchar	50	default=NULL, foreign key

			from login_users_info
block	varchar	1	default=NULL

make_payment_transaction

Name field	Data type	Length	Description
Transaction_id	int	11	Primary key, Auto increment
data	date		
sender_name	varchar	100	
sender_mobile	varchar	10	
sender_acno	varchar	14	
sender_amount	int	11	
sender_purpose	mediumtext		
receiver_name	varchar	100	
receiver_mobile	varchar	10	
receiver_acno	varchar	14	
payment_type	varchar	10	
sender_username	varchar	100	

7. Work Flow/Layouts

Screen Shots:-

Dashboard



Login

Registration

Forgot Password

127.0.0.1:8000/Loginforgot/

The screenshot shows the 'Forgot Password' page of the Online Payment System. At the top, it says 'Online Payment System' and 'You can do your payment using this System..'. Below that is the heading 'Forgot Password'. There are three input fields: 'Username:', 'Mobile No:', and 'New Password:', each with a corresponding placeholder. Below the fields are a 'Submit' button and a 'Go to Login' link.

Home Page

127.0.0.1:8000/Login/welcome/

The screenshot shows the 'Welcome' page of the Online Payment System. The header includes links for 'Online Payment System', 'Make Payment', 'Transaction History', 'Bank Account', 'UPI', 'Profile', and 'Logout'. The main area features a large blue background image of a person using a laptop and a smartphone, with the text 'Welcome,ronakpadaliya' above them. To the right, the words 'ONLINE PAYMENT SYSTEM' are displayed in large white capital letters.

Add Bank account

127.0.0.1:8000/addbankaccount/

The screenshot shows the 'Add bank account' page. The header links are identical to the home page. The main content area has a heading 'Add bank account' with the sub-instruction 'You have to do add your bank account to do your payment..'. Below this is a form with two input fields: 'Mobile No:' and 'Account No:', both with validation messages ('Digit must be 10' and 'Digit must be 14'). There is a 'Add' button and a link 'For change Mobile no'.

Change Mobile Number

Change Mobile Number

You are changing your mobile number...

Change Mobile Number

Enter OLD Mobile No :

Enter New Mobile NO :

change

Make UPI Pin for payment

UPI Form

You have to create UPI for secure Payment.....

UPI

Enter mobile number : Digit must be 10

Enter account number : Digit must be 14

Enter username : Minimum length is 6

Enter Password : Digit must be 8

Enter UPI PIN : UPI PIN Size is 4 in digit

change

Change Bank Account

Change bank account

You are changing your bank account.....

Change Bank Account

Mobile No :

OLD Account No :

NEW Account No :

Add

For change [Mobile no](#)

Remove Bank Account

127.0.0.1:8000/changeremove_bank_account/

Online Payment System Make Payment Transaction History Bank Account ▾ UPI ▾

Profile Logout

Remove bank account
You are removing your bank account.....

Remove Bank Account

Mobile No :

Account No :

Password :

Remove

Block Bank Account

127.0.0.1:8000/changeblock_bank_account/

Online Payment System Make Payment Transaction History Bank Account ▾ UPI ▾

Profile Logout

Block bank account
You are block your bank account.....

Block Bank Account

Mobile No :

Account No :

Password :

Block

Unblock Bank Account

127.0.0.1:8000/changebank_account_unblock/

Online Payment System Make Payment Transaction History Bank Account ▾ UPI ▾

Profile Logout

Unblock bank account
You are unblock your bank account.....

Unblock Bank Account

Mobile No :

Account No :

Password :

Unblock

Change UPI Pin

Change UPI PIN

You are changing your UPI PIN.....

Change UPI PIN

Enter Username :

Enter Password :

Enter OLD UPI PIN :

Enter NEW UPI PIN :

Change

Make Payment

Make Payment

You have to do add your bank account to do your payment..

Select Payment Method

1.Mobile number

2.Using Account Number

Make payment using Mobile Number

Payment using Mobile No

You are do your payment using Mobile No..

Payment

Enter Your Account No : Enter Your acno

Enter amount : Enter Amount

Enter purpose : Enter Your purpose

Enter receiver Mobile No: Enter receiver Mobile NO

Enter UPI PIN : Enter UPI

Pay

Make payment using Account Number

127.0.0.1:8000/make_paymentacno/

Online Payment System Make Payment Transaction History Bank Account UPI Profile Logout

Payment using ACNO

You are do your payment using ACNO..

Payment

Enter Your Account No :

Enter amount :

Enter purpose :

Enter receiver account No :

Enter UPI PIN :

Pay

Profile

127.0.0.1:8000/Profileprofile/

Homepage

User Profile

First Name	ronak
Last Name	padaliya
Email	ronakpadaliya77@gmail.com
Username	ronakpadaliya
Mobile No	932800556

Update Profile **Delete Account**

Bank Details

Firstname	ronak
Lastname	padaliya
Account No	12345678901234
Mobile No	932800556
Balance	17630

Delete user's application account

127.0.0.1:8000/Profiledeleteaccount/

Online Payment System Make Payment Transaction History Bank Account UPI Profile Logout

Delete Account

You are delete Your account From this System.....

Delete Account

UserName :

Mobile No :

Password :

Delete Account

8. Conclusion

Hear we can conclude that this system is easy to use , secure and better than cash payment. Whenever you want to make payment and you have not enough cash then this is useful for that user.

- In this system if user have more bank account then if user is add bank account and try to add more bank account then user not allowed to do this. First user need to remove bank account then after add other bank account.
- If user already added bank account then if user delete his/her account from this web application then bank data as it is present but transaction history and user data will be lost. User need to again register for use it.
- If user block his/her bank account then this application not allow to user to make payment, change bank account, remove bank account, change upi, etc...
- User can update his/her profile from application profile.

9. Limitations and Future Extensions

Limitations:

- If user want to [change username](#) then this is not possible.
- If user want to [change mobile number](#) then first user need to remove bank account if added then after go again with [add bank account](#) and in this window below option is for change mobile number.
- If user want to [update any kind of bank data](#) then it is not under for this application because it is by default provided by admin of this user.
- If user forgot his/her password then for change password user need to logout then select [forgot password](#) option then it can be change.

Future Extensions:

- [All limitations](#) can be implement in afterward version of this web application.
- Hear in this web application we can add [QR Code payment](#).
- In this we can verify user using [OTP authentication](#) when user make account, add/change/remove/block/unblock bank account, delete application account and use other functionality.
- User can [send request](#) to people who can make payment for user.

10. Bibliography

Reference use for develop web application:

Stackoverflow : <https://stackoverflow.com/>

Django official documentation :- <https://docs.djangoproject.com/en/3.1/>

W3Schools :- <https://www.w3schools.com/>

And many others...

Tools user for develop web application:

Django framework : <https://www.djangoproject.com/>

Visual Studio Code : <https://code.visualstudio.com/>

Git Hub : <https://github.com/>