

Assignment-11

ELP - 718 Telecom Software Laboratory

Murali Krishnan K H

2019JTM2680

2019-2021

A report presented for the assignment on
Python with KIVY



**Bharti School Of
Telecommunication Technology and Management
IIT Delhi
India
October 22, 2019**

Contents

1 Problem Statement-1	3
1.1 Problem Statement	3
1.2 Algorithm and Implementation	5
1.3 Program Structure	6
1.5 Screenshots	8
A Appendix	13
A.1 Code for PS	13

List of Figures

1	Main menu	8
2	Employee login menu	8
3	Employee option menu	9
4	Employee update	9
5	Employee view	10
6	Customer menu	10
7	Customer bank menu	11
8	Customer transaction details	11
9	Customer forgot password	12
10	New Customer	12

1 Problem Statement-1

1.1 Problem Statement

RBI put efforts to reduce the wastage of resources. NITI Aayog proposed a solution that instead of maintaining separate accounts in each bank, only one account related to all banks must be maintained by a single user is proposed. Bank gives the project to a software developer, who is in this case you, expecting a final end product which is a single portal interface with options equipped as mentioned below.

Requirements:

- The project must have the capability to handle large sets of data.
- A simple method to enroll in various banks with a single account number is the core idea behind this project.
- Account details like account number and the name of the account holder remain same across all the banks but the details of the bank like IFSC Code, Bank name, MICR code etc. changes according to the bank being considered. (Again, the details of the banks and the account details to be entered into the database can be of user choice).
- A person entering into the portal should be given options to choose an operation that is to be performed related to a particular bank out of all multiple existing banks(only one account pertaining to a single type of bank).

For example, withdrawal is one such operation and that should be specific to each and every bank.

- All details must be shared between the various accounts but some confidential details cannot be shared. They are as follows: (Two of 'n' such confidential parameters have been mentioned below. User should consider these three parameters at least while implementation and allow maximum number of parameters to his own discretion)
 - Amount in a particular bank account
 - Transition history of that bank account

- As Project is in the starting phase, we assume any account holder has only accounts in three different banks namely SBI, HDFC, ICICI

UI Requirements:

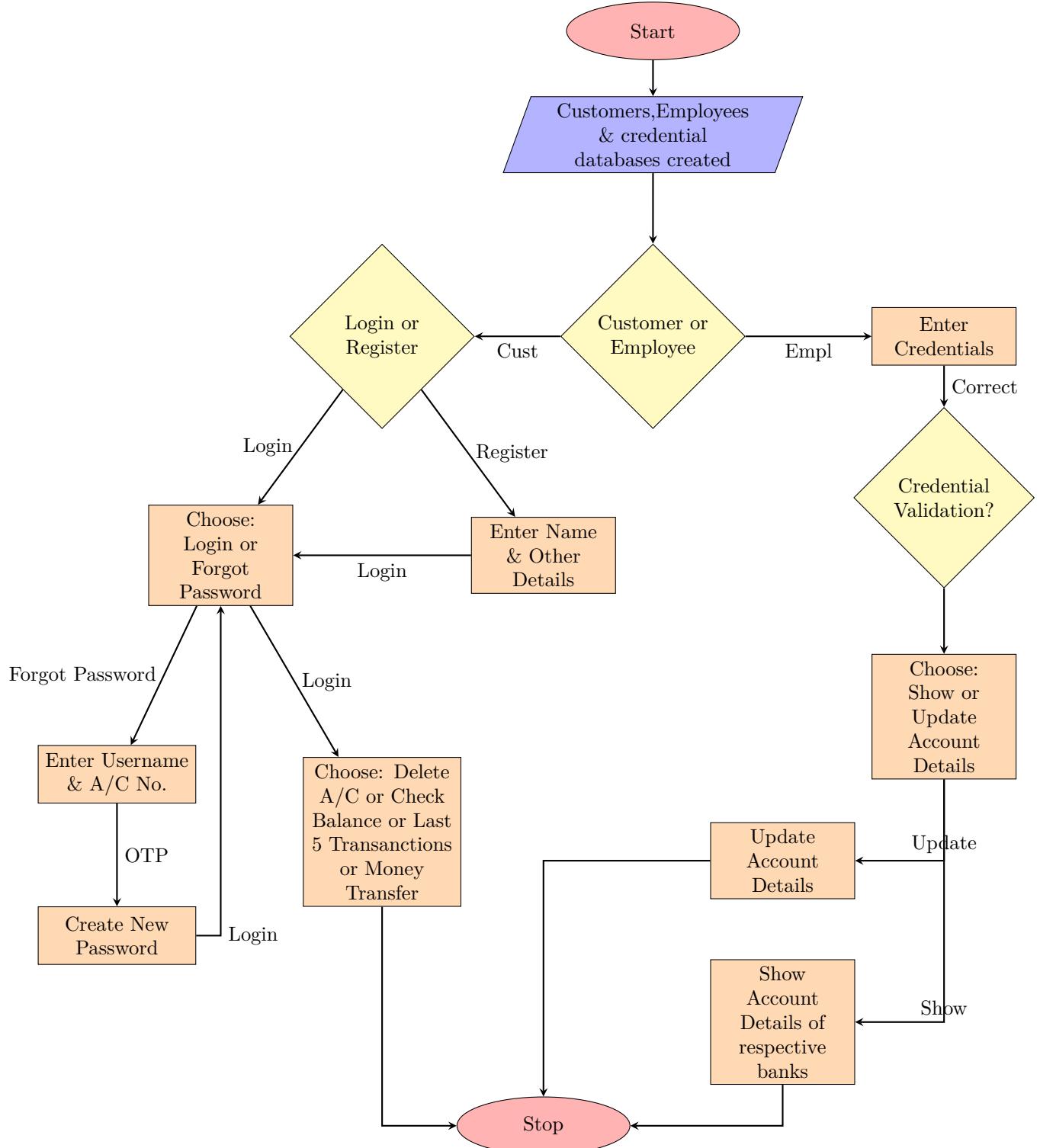
- First screen should be attractive with an appropriate logo and background image for the unified bank portal providing two options for employee and customer.
- Customer screen, if selected:
 - * Should give two widgets namely login, create new account
 - * **Login**, if selected
 - Should ask for username and password
 - Implement forgot password
 - Take username, account number and generate a 4-digit OTP after submitting the entered details and send the generated OTP to telegram bot.
 - Enter this OTP and after matched, a window asking for new password and confirm password should pop-up
 - After entering these details, the password of the particular customer should get updated in the database and the window should go back to the customer login screen.
 - After entering correct credentials, four options must be displayed, But first customer must select the **BANK**

- * Delete Account from Bank
 - Ask for Bank
 - Password
 - Random 4 digit system generated code received on telegram bot
 - Confirmation message on telegram bot “Account Deleted from <Bank name>”
- Check current balance
- Last five transactions
- Money transfer
 - * Account number to be transferred to must be entered and after entering, name and mobile number related to that account must be displayed.
 - * Then amount must be entered.
 - * Message should go to telegram bot displaying the amount sent and the account number to which the money has been sent with **Time and Date**
- After selecting the “New User Registration” following details must be shown on the screen
 - User must enter following details for successful registration
 - * Name
 - * Date of Birth
 - * Address
 - * Country
 - * Bank (To which the customer want to be registered to)
 - * Mobile Number
 - After click on the submit button a 7 digit number must be generated and to be sent to the User Telegram Bot And a pop window generated with message “Successfully Registered” and direct the user to login screen
 - * eg. You are successfully registered for the <Bank Name> with account number “10 digit-number”.
 - EG. “**2345362768**”
- “Employee login screen”
 - User-Id (Must be in the form of “**SBI234**” and employee already registered in the bank)
 - Password
 - After successful login, An employee can be able to perform the following tasks
 - * Update account details of customer, based on the details filled up by the customer
 - The following details can be updated
 - Address
 - Mobile Number
 - Date of Birth
 - When any data related to customer gets updated, an ”**Acknowledge message** must be sent to customer via telegram bot with **Time and Date**
 - * Employee can see the account details(Details related to employee of one of the multiple banks) of all customers.

1.2 Algorithm and Implementation

1. Start
2. Create tables in the database according to necessity.
3. Create kv file
4. Link kv file
5. Display screen with choice Customer and Employee
6. if choice is customer then
7. Again display a screen with two choice login, forgot password and new user
8. if user is logged in Display options for deleting, transferring money and show balance
9. if choice is newuser then create the account and update it into table
10. if choice is Employee then display employ login page
11. after logged in ,display screen with two choice update and view details
12. if choice is update employee can update customer details
13. if choice is view employ can view customer details
14. Exit

1.3 Program Structure



1.5 Screenshots



Figure 1: Main menu

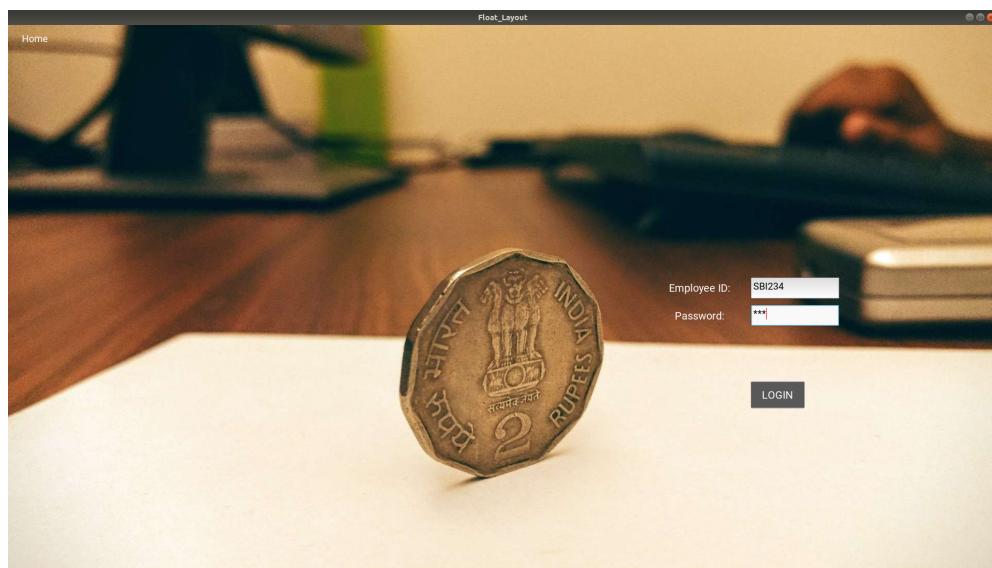


Figure 2: Employee login menu



Figure 3: Employee option menu

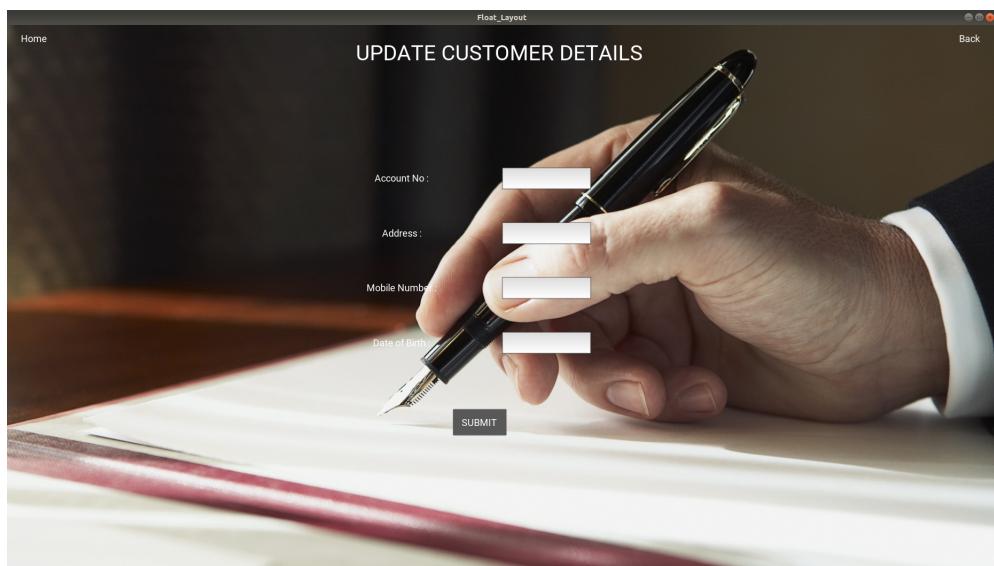


Figure 4: Employee update

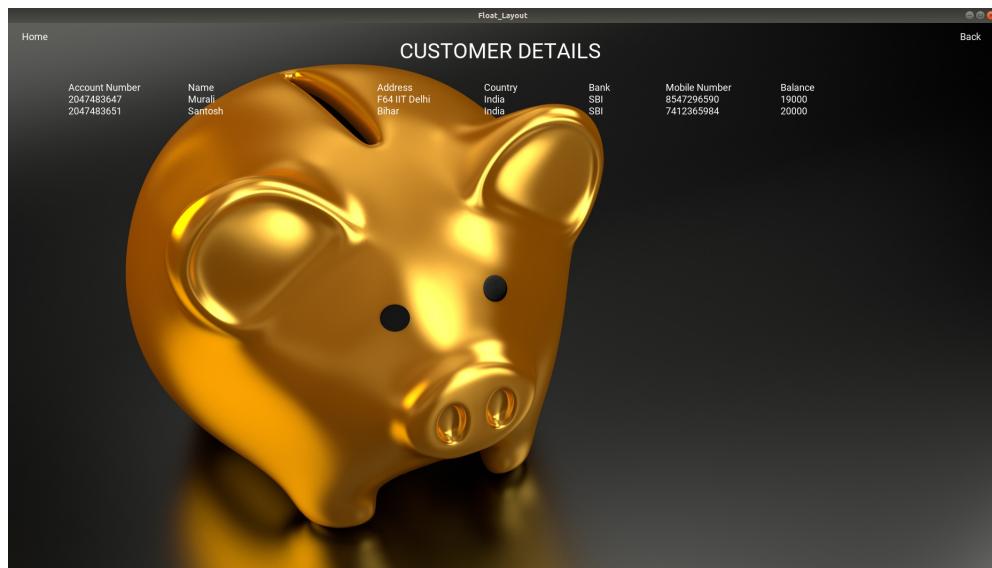


Figure 5: Employee view

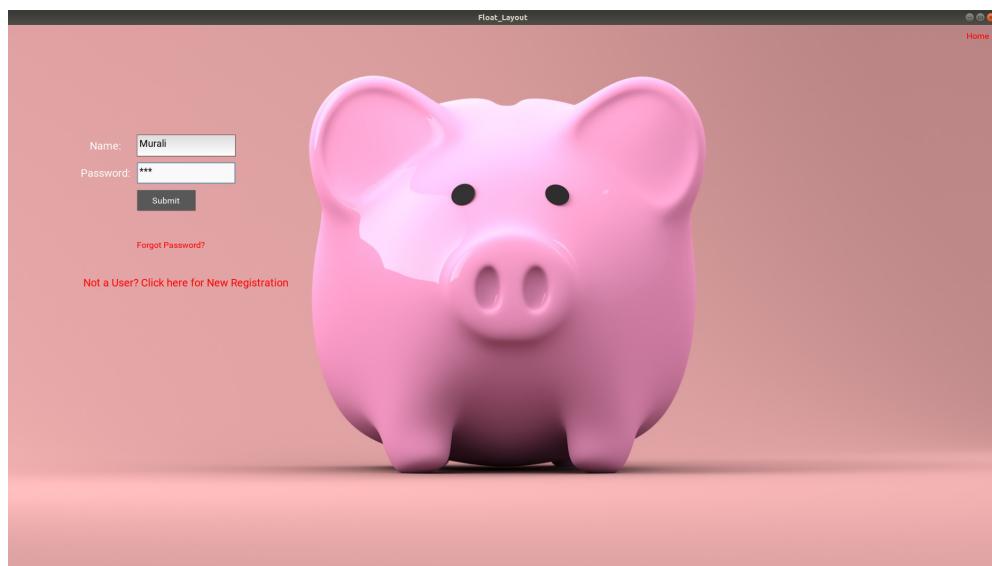


Figure 6: Customer menu



Figure 7: Customer bank menu



Figure 8: Customer transaction details



Figure 9: Customer forgot password

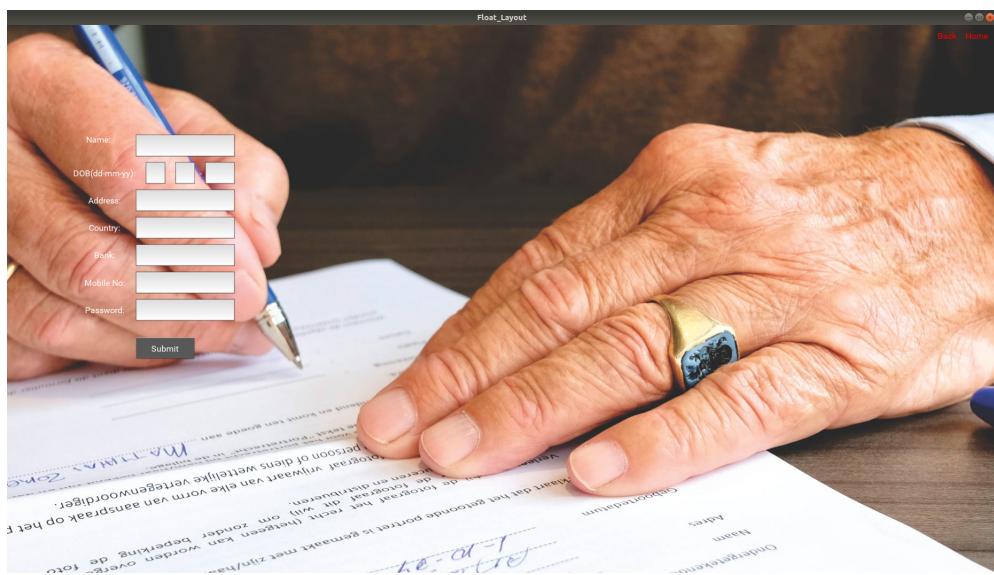


Figure 10: New Customer

A Appendix

A.1 Code for PS

```
1 # Import libary for kivy
2 import kivy
3 from kivy.app import App
4 from kivy.uix.label import Label
5 from kivy.lang import Builder
6 from kivy.uix.screenmanager import ScreenManager, Screen, FadeTransition
7 from kivy.uix.widget import Widget
8 from kivy.graphics import Line
9 from kivy.uix.textinput import TextInput
10 from kivy.uix.gridlayout import GridLayout
11 from kivy.uix.floatlayout import FloatLayout
12 from kivy.core.window import Window
13 from kivy.properties import ObjectProperty
14 import pymysql
15 from getpass import getpass
16 from kivy.uix.floatlayout import FloatLayout
17 from kivy.uix.dropdown import DropDown
18 from kivy.uix.popup import Popup
19 from kivy.uix.button import Button
20 import requests
21 from random import randint
22 def telegram_bot_sendtext(bot_message):
23
24     bot_token = '890171469:AAFFm8cvIAtAunETDdYllpwOsqVZZZnR3VA'
25     bot_chatID = '718923667'
26     send_text = 'https://api.telegram.org/bot' + bot_token + '/sendMessage?chat_id=' + \
27     bot_chatID + '&parse_mode=Markdown&text=' + bot_message
28
29     response = requests.get(send_text)
30
31     return response.json()
32 def isNumber(s) :
33
34     for i in range(len(s)) :
35         if s[i].isdigit() != True :
36             return False
37
38     return True
39 def random_with_N_digits(n):
40     range_start = 10** (n-1)
41     range_end = (10**n)-1
42     return randint(range_start , range_end)
43
44 def invalidLogin():
45     pop = Popup(title='Invalid Login',
46                 content=Label(text='Invalid username or password.'),
47                 size_hint=(None, None), size=(400, 400))
48     pop.open()
49 def invalidAcc():
50     pop = Popup(title='Invalid Login',
51                 content=Label(text='Invalid username or account number.'),
52                 size_hint=(None, None), size=(400, 400))
53     pop.open()
54 def invalidAcc3():
55     pop = Popup(title='Invalid Account',
56                 content=Label(text='Invalid account number.'),
57                 size_hint=(None, None), size=(400, 400))
58     pop.open()
59 def invalidCustomer():
```

```

60     pop = Popup(title='Invalid Login',
61                 content=Label(text='Customer not registered.'),
62                 size_hint=(None, None), size=(400, 400))
63     pop.open()
64     pop.open()
65 def invalidEmployee():
66     pop = Popup(title='Invalid Login',
67                 content=Label(text='Employee not registered.'),
68                 size_hint=(None, None), size=(400, 400))
69     pop.open()
70
71
72 def invalidForm():
73     pop = Popup(title='Invalid Form',
74                 content=Label(text='Please fill in all inputs with valid information.' ),
75                 size_hint=(None, None), size=(400, 400))
76     pop.open()
77 def invalidDOB():
78     pop = Popup(title='Invalid Form',
79                 content=Label(text='Please enter a valid DOB.' ),
80                 size_hint=(None, None), size=(400, 400))
81     pop.open()
82 def validReg(x,y):
83     pop = Popup(title='Valid Registration',
84                 content=Label(text='You have registered succesfully and your account number
85                 and registration ID is ' + str(x) + str(y)),
86                 size_hint=(None, None), size=(800, 800))
87     pop.open()
88 def dispDetails(x,y):
89     pop = Popup(title='',
90                 content=Label(text='The Name and mobile number of the beneficiary is ' +
91                               str(x) + str(y)),
92                 size_hint=(None, None), size=(800, 800))
93
94     pop.open()
95 def getotp(x):
96     pop = Popup(title='OTP',
97                 content=Label(text='Your OTP is ' + str(x)),
98                 size_hint=(None, None), size=(400, 400))
99     pop.open()
100 def deletesuccess(x):
101     pop = Popup(title='Success',
102                 content=Label(text='You have deleted the account successfully with
103                 deletion id ' + str(x)),
104                 size_hint=(None, None), size=(400, 400))
105 def updatesuccess():
106     pop = Popup(title='Update',
107                 content=Label(text='Password updated sucessfully.'),
108                 size_hint=(None, None), size=(400, 400))
109 def updatesuccess2():
110     pop = Popup(title='Update',
111                 content=Label(text='Data updated sucessfully.'),
112                 size_hint=(None, None), size=(400, 400))
113     pop.open()
114 def invalidAcc2():
115     pop = Popup(title='Invalid Login',
116                 content=Label(text='Invalid username or account number or OTP'),
117                 size_hint=(None, None), size=(400, 400))
118     pop.open()
119 def invalidBank():
120     pop = Popup(title='Invalid selection',
121                 content=Label(text='Sorry you do not have an account in this bank.')),

```

```

122                     size_hint=(None, None), size=(400, 400))
123         pop.open()
124     def dispBalance(x):
125         pop = Popup(title='Balance',
126                     content=Label(text='Your available balance is: ' + str(x)),
127                     size_hint=(None, None), size=(400, 400))
128         pop.open()
129     def invalidFund():
130         pop = Popup(title='Balance',
131                     content=Label(text='Your account balance is insufficient for this
132 transaction'),
133                     size_hint=(None, None), size=(400, 400))
134         pop.open()
135     def invalidFund2():
136         pop = Popup(title='Balance',
137                     content=Label(text='Amount entered is invalid'),
138                     size_hint=(None, None), size=(400, 400))
139         pop.open()
140     def transsuccess():
141         pop = Popup(title='Transaction',
142                     content=Label(text='Transaction completed successfully.'),
143                     size_hint=(None, None), size=(400, 400))
144         pop.open()
145     def cantdelete():
146         pop = Popup(title='Restricted',
147                     content=Label(text='You cannot delete the bank you have logged in.'),
148                     size_hint=(None, None), size=(400, 400))
149         pop.open()
150     def nobank():
151         pop = Popup(title='Invalid',
152                     content=Label(text='You do not have account in this bank.'),
153                     size_hint=(None, None), size=(400, 400))
154         pop.open()
155     def nameoccupied():
156         pop = Popup(title='Invalid',
157                     content=Label(text='Username already occupied.'),
158                     size_hint=(None, None), size=(400, 400))
159         pop.open()
160
161 #Establishing connection with user and database
162 mydb = pymysql.connect(
163     host="localhost",
164     user="user",
165     passwd="password",
166     database="ps"
167 )
168
169 #Setting up the cursorfrom kivy.uix.floatlayout import FloatLayout
170 mycursor = mydb.cursor()
171
172 class Widgets(Widget):
173     pass
174
175 class MainScreen(Screen):
176     pass
177 #####Employee's login verification screen#####
178 class EmployeeScreen(Screen):
179     nam = ObjectProperty(None)
180     pas = ObjectProperty(None)
181     def btn(self):
182         global s
183         s = self.nam.text
184         p = self.pas.text

```

```

186     mycursor.execute("SELECT * FROM emp")
187     myr = mycursor.fetchall()
188     if len(s) == 0 or len(p)==0:
189         invalidForm()
190     else:
191         if s in str(myr):
192             mycursor.execute("select emp_pass from emp where emp_id = %s ",s)
193             myr1 = list(mycursor.fetchall())
194             v =str(myr1[0][0])
195             if p == v:
196                 presentation.current = "emp1"
197             elif len(p) == 0 :
198                 invalidLogin()
199             else:
200                 invalidLogin()
201             else:
202                 invalidEmployee()
203
204             self.nam.text = ""
205             self.pas.text = ""
206 class EmployeeScreen1(Screen):
207     pass
208 #####Employee screen to update details#####
209 class EmployeeScreen2(Screen):
210     euac = ObjectProperty(None)
211     euaddress = ObjectProperty(None)
212     eumob = ObjectProperty(None)
213     eudob = ObjectProperty(None)
214     def btn1(self):
215         eux=self.euac.text
216         euy=self.euaddress.text
217         euz=self.eumob.text
218         euzz=self.eudob.text
219
220         if len(eux)== 0 or len(euy)==0 or len(euz)==0 or len(euzz)==0:
221             invalidForm()
222         else:
223             mycursor.execute("select Cust_acc from Customer where Cust_acc = %s", eux)
224             myr = mycursor.fetchall()
225             var1 = myr[0][0]
226             day, month, year = euzz.split('-')
227             sql1 ="update Customer set Cust_add = %s ,Cust_mob = %s , Cust_dd= %s , Cust_mm= %s
228             , Cust_yy= %s where Cust_acc = %s"
229             sql2 = (euy,euz,day,month,year,var1)
230             mycursor.execute(sql1,sql2)
231             mydb.commit()
232             updatesuccess2()
233             presentation.current="emp1"
234 #####Employee screen to print details#####
235 class EmployeeScreen3(Screen):
236     l1 = ObjectProperty(None)
237     l2 = ObjectProperty(None)
238     l3 = ObjectProperty(None)
239     l4 = ObjectProperty(None)
240     l5 = ObjectProperty(None)
241     l6 = ObjectProperty(None)
242     l7 = ObjectProperty(None)
243     l8 = ObjectProperty(None)
244
245     def on_enter(self, *args):
246         mycursor.execute("select emp_bank from emp where emp_id = %s ",s)
247         myr = mycursor.fetchall()
248         x=myr[0][0]
249         mycursor.execute("select count(*) from Customer where Cust_bank =%s ",x)

```

```

250     myr = mycursor.fetchall()
251     y=myr[0][0]
252     mycursor.execute("select Cust_acc from Customer where Cust_bank = %s",x)
253     myr = list(mycursor.fetchall())
254     self.l1.text="Account Number \n"
255     for m in range(y):
256         self.l1.text=self.l1.text+str(myr[m][0])+'\n'
257
258     mycursor.execute("select Cust_name from Customer where Cust_bank = %s",x)
259     myr = list(mycursor.fetchall())
260     self.l2.text="Name \n"
261     for m in range(y):
262         self.l2.text=self.l2.text+str(myr[m][0])+'\n'
263
264     mycursor.execute("select Cust_addr from Customer where Cust_bank = %s",x)
265     myr = list(mycursor.fetchall())
266     self.l4.text="Address \n"
267     for m in range(y):
268         self.l4.text=self.l4.text+str(myr[m][0])+'\n'
269
270     mycursor.execute("select Cust_coun from Customer where Cust_bank = %s",x)
271     myr = list(mycursor.fetchall())
272     self.l5.text="Country \n"
273     for m in range(y):
274         self.l5.text=self.l5.text+str(myr[m][0])+'\n'
275
276     mycursor.execute("select Cust_bank from Customer where Cust_bank = %s",x)
277     myr = list(mycursor.fetchall())
278     self.l6.text="Bank \n"
279     for m in range(y):
280         self.l6.text=self.l6.text+str(myr[m][0])+'\n'
281
282     mycursor.execute("select Cust_mob from Customer where Cust_bank = %s",x)
283     myr = list(mycursor.fetchall())
284     self.l7.text="Mobile Number \n"
285     for m in range(y):
286         self.l7.text=self.l7.text+str(myr[m][0])+'\n'
287
288     mycursor.execute("select Cust_bal from Custbank where Cust_bank = %s",x)
289     myr = list(mycursor.fetchall())
290     self.l8.text="Balance \n"
291     for m in range(y):
292         self.l8.text=self.l8.text+str(myr[m][0])+'\n'
293 #####Customer main screen#####
294 class CustomerScreen(Screen):
295     nam = ObjectProperty(None)
296     email = ObjectProperty(None)
297     def btn(self):
298         global acco
299         s = self.nam.text
300         p = self.pas.text
301         mycursor.execute("SELECT * FROM Customer")
302         myresult = mycursor.fetchall()
303         if len(s) == 0 or len(p)==0:
304             invalidForm()
305         else:
306             if s in str(myresult):
307                 mycursor.execute("select Cust_pass from Customer where Cust_name= %s ",s)
308                 myr1 = list(mycursor.fetchall())
309                 v =str(myr1[0][0])
310                 if p == v:
311                     mycursor.execute("select Cust_acc from Customer where Cust_name= %s ",s)
312                     myr2 = list(mycursor.fetchall())
313                     acco =str(myr2[0][0])
314                     presentation.current="Customer1"

```

```

315         elif len(p) == 0 :
316             invalidLogin()
317         else:
318             invalidLogin()
319     else:
320         invalidCustomer()
321
322     self.nam.text = ""
323     self.pas.text = ""
324
325
326 #####Customer screen to execute bank selection#####
327 class MenuScreen(Screen):
328     def sbibank(self):
329         global ban
330         ban = "SBI"
331         mycursor.execute("SELECT Cust_bank FROM Custbank where Cust_acc=%s ",acco)
332         myresult = list(mycursor.fetchall())
333         q2 = myresult[0][0]
334         q = "SBI"
335         if q == q2:
336             presentation.current="Customer2"
337         else:
338             invalidBank()
339     def icicibank(self):
340         global ban
341         ban = "ICICI"
342         mycursor.execute("SELECT Cust_bank FROM Custbank where Cust_acc=%s ",acco)
343         myresult = list(mycursor.fetchall())
344         q2 = myresult[0][0]
345         q = "ICICI"
346         if q == q2:
347             presentation.current="Customer2"
348         else:
349             invalidBank()
350     def hdfcbank(self):
351         global ban
352         ban = "HDFC"
353         mycursor.execute("SELECT Cust_bank FROM Custbank where Cust_acc=%s ",acco)
354         myresult = list(mycursor.fetchall())
355         q2 = myresult[0][0]
356         q = "HDFC"
357         if q == q2:
358             presentation.current="Customer2"
359         else:
360             invalidBank()
361 #####Screen to display balance and last 5 transactions#####
362 class MenuScreen2(Screen):
363     bal = ObjectProperty(None)
364     def balance(self):
365         sql3 = "SELECT Cust_bal FROM Custbank where Cust_acc=%s and Cust_bank=%s"
366         val3 = (acco,ban)
367         mycursor.execute(sql3 ,val3)
368         myresult = list(mycursor.fetchall())
369         h = myresult[0][0]
370         dispBalance(h)
371 class transactionScreen(Screen):
372     l1 = ObjectProperty(None)
373     l2 = ObjectProperty(None)
374     l3 = ObjectProperty(None)
375     l4 = ObjectProperty(None)
376     l5 = ObjectProperty(None)
377     l6 = ObjectProperty(None)
378     def trans(self):
379

```

```

380     mycursor.execute("select Cust_id from history where Cust_acc = %s ORDER BY Cust_id
DESC LIMIT 5",acco)
381     myr3 = list(mycursor.fetchall())
382     y=len(myr3)
383     print(myr3[0][0])
384     self.l1.text="Cust_id \n"
385     for m in range(y):
386         self.l1.text+=self.l1.text+str(myr3[m][0])+'\n'
387
388     mycursor.execute("select Cust_acc from history where Cust_acc = %s ORDER BY Cust_id
DESC LIMIT 5",acco)
389     myr = list(mycursor.fetchall())
390     self.l2.text="Account no \n"
391     for m in range(y):
392         self.l2.text+=self.l2.text+str(myr[m][0])+'\n'
393
394     mycursor.execute("select Cust_state from history where Cust_acc = %s ORDER BY
Cust_id DESC LIMIT 5",acco)
395     myr = list(mycursor.fetchall())
396     self.l3.text="c/d \n"
397     for m in range(y):
398         self.l3.text+=self.l3.text+str(myr[m][0])+'\n'
399
400     mycursor.execute("select Cust_amt from history where Cust_acc = %s ORDER BY Cust_id
DESC LIMIT 5",acco)
401     myr = list(mycursor.fetchall())
402     self.l4.text="Amount \n"
403     for m in range(y):
404         self.l4.text+=self.l4.text+str(myr[m][0])+'\n'
405
406     mycursor.execute("select Cust_bank from history where Cust_acc = %s ORDER BY Cust_id
DESC LIMIT 5",acco)
407     myr = list(mycursor.fetchall())
408     self.l5.text="Bank \n"
409     for m in range(y):
410         self.l5.text+=self.l5.text+str(myr[m][0])+'\n'
411
412     mycursor.execute("select Cust_ben from history where Cust_acc = %s ORDER BY Cust_id
DESC LIMIT 5",acco)
413     myr = list(mycursor.fetchall())
414     self.l6.text="To/From \n"
415     for m in range(y):
416         self.l6.text+=self.l6.text+str(myr[m][0])+'\n'
417 #####SCreen for money transfer#####
418 class transferScreen(Screen):
419     acn :ObjectProperty(None)
420     def ver_acc(self):
421         a = self.acn.text
422         global benacco
423         benacco = a
424         mycursor.execute("SELECT * FROM Custbank")
425         myresult = mycursor.fetchall()
426         if len(a) == 0:
427             invalidForm()
428         else:
429             if a in str(myresult) and a != acco:
430                 mycursor.execute("SELECT Cust_name FROM Customer where Cust_acc=%s ",a)
431                 myresult = list(mycursor.fetchall())
432                 l1 = myresult [0][0]
433                 l1 = l1 + " , "
434                 mycursor.execute("SELECT Cust_mob FROM Customer where Cust_acc=%s ",a)
435                 myresult = list(mycursor.fetchall())
436                 l2 = myresult [0][0]
437                 dispDetails(l1,l2)
438                 presentation.current="transfer2"

```

```

439     else:
440         invalidAcc3()
441
442
443 class transferScreen2(Screen):
444     amo = ObjectProperty(None)
445     def sendmoney(self):
446         amt = self.amo.text
447         if len(amt) == 0:
448             invalidForm()
449         elif isNumber(amt):
450             sql3 = "SELECT Cust_bal FROM Custbank where Cust_acc=%s and Cust_bank=%s"
451             val3 = (acco,ban)
452             mycursor.execute(sql3, val3)
453             myresult = list(mycursor.fetchall())
454             h = myresult[0][0]
455             x = int(amt)
456             y = int(h)
457             if x < y:
458                 g1 = "debited"
459                 g2 = "credited"
460                 sql = "INSERT INTO history (Cust_acc, Cust_state, Cust_amt, Cust_bank,
461 Cust_ben) VALUES (%s,%s, %s,%s,%s)"
462                 val = (acco,g1,amt,ban,benacco)
463                 mycursor.execute(sql, val)
464                 mydb.commit()
465                 mycursor.execute("SELECT Cust_bank FROM Custbank where Cust_acc=%s ",benacco
466 )
467                 myresult = list(mycursor.fetchall())
468                 ban2 = myresult[0][0]
469                 sql = "INSERT INTO history (Cust_acc, Cust_state, Cust_amt, Cust_bank,
470 Cust_ben) VALUES (%s,%s, %s,%s,%s)"
471                 val = (benacco,g2,amt,ban2,acco)
472                 mycursor.execute(sql, val)
473                 mydb.commit()
474                 z = y-x
475                 z = str(z)
476                 sql3 = "SELECT Cust_bal FROM Custbank where Cust_acc=%s and Cust_bank=%s"
477                 val3 = (benacco,ban2)
478                 mycursor.execute(sql3, val3)
479                 myresult = list(mycursor.fetchall())
480                 h1 = myresult[0][0]
481                 y1 = int(h1)
482                 z1 = x+y1
483                 z1 = str(z1)
484                 sql1 = "UPDATE Custbank SET Cust_bal = %s where Cust_acc = %s"
485                 val1 = (z,acco)
486                 mycursor.execute(sql1, val1)
487                 mydb.commit()
488                 sql1 = "UPDATE Custbank SET Cust_bal = %s where Cust_acc = %s"
489                 val1 = (z1,benacco)
490                 mycursor.execute(sql1, val1)
491                 mydb.commit()
492                 transsuccess()
493                 presentation.current="Customer2"
494             else:
495                 invalidFund()
496         else:
497             invalidFund2()
498 ######Screen to delete account#####
499 class deleteScreen(Screen):
500     def sbibank(self):
501         mycursor.execute("SELECT Cust_bank FROM Custbank where Cust_acc=%s ",acco)
502         myresult = list(mycursor.fetchall())
503         x="SBI"

```

```

501     l = len(myresult)
502     if ban=="SBI":
503         cantdelete()
504     else:
505         for m in range(1):
506             if x == myresult[m][0]:
507                 sql = "DELETE FROM Custbank where Cust_acc=%s AND Cust_bank=%s"
508                 val = (acco,x)
509                 mycursor.execute(sql ,val)
510                 mydb.commit()
511                 ra = random_with_N_digits(4)
512                 deletesuccess(ra)
513                 presentation.current="Customer2"
514
515             else:
516                 nobank()
517
518
519
520     def icicibank(self):
521         mycursor.execute("SELECT Cust_bank FROM Custbank where Cust_acc=%s ",acco)
522         myresult = list(mycursor.fetchall())
523         x="ICICI"
524         l = len(myresult)
525         if ban=="ICICI":
526             cantdelete()
527         else:
528             for m in range(1):
529                 if x == myresult[m][0]:
530                     sql = "DELETE FROM Custbank where Cust_acc=%s AND Cust_bank=%s"
531                     val = (acco,x)
532                     mycursor.execute(sql ,val)
533                     mydb.commit()
534                     ra = random_with_N_digits(4)
535                     deletesuccess(ra)
536                     presentation.current="Customer2"
537
538             else:
539                 nobank()
540
541
542     def hdfcbank(self):
543         mycursor.execute("SELECT Cust_bank FROM Custbank where Cust_acc=%s ",acco)
544         myresult = list(mycursor.fetchall())
545         x="HDFC"
546         l = len(myresult)
547         if ban=="HDFC":
548             cantdelete()
549         else:
550             for m in range(1):
551                 if x == myresult[m][0]:
552                     sql = "DELETE FROM Custbank where Cust_acc=%s AND Cust_bank=%s"
553                     val = (acco,x)
554                     mycursor.execute(sql ,val)
555                     mydb.commit()
556                     ra = random_with_N_digits(4)
557                     deletesuccess(ra)
558                     presentation.current="Customer2"
559
560             else:
561                 nobank()
562 #####Screen to create new user#####
563 class CreateScreen(Screen):
564     nam1 = ObjectProperty(None)
565     dd = ObjectProperty(None)

```

```

566 month = ObjectProperty(None)
567 yy = ObjectProperty(None)
568 add = ObjectProperty(None)
569 country = ObjectProperty(None)
570 bank = ObjectProperty(None)
571 mob = ObjectProperty(None)
572 #
573 # dd =int(dd)
574 # month = int(month)
575 # yy =int(yy)
576 def fetch(self):
577     n = self.nam.text
578     d = self.dd.text
579     month = self.m.text
580     y = self.yy.text
581     a = self.add.text
582     c = self.country.text
583     b = self.bank.text
584     mo = self.mob.text
585     p = self.pas.text
586     mycursor.execute("SELECT Cust_name FROM Customer")
587     myresult = list(mycursor.fetchall())
588     print(myresult)
589     l = len(myresult)
590     if len(n) == 0 or len(d) == 0 or len(month) == 0 or len(y) == 0 or len(a) == 0 or
591     len(c) == 0 or len(b) == 0 or len(mo) == 0 or len(p) == 0:
592         invalidForm()
593     d1 = int(d)
594     month1 = int(month)
595     y1 = int(y)
596     if d < 1 or d > 31 or month < 1 or month > 12 or y < 1900 or y > 2019:
597         invalidDOB()
598     else:
599         for m in range(l):
600             if n == myresult[m][0]:
601                 nameoccupied()
602             else:
603                 sql = "INSERT INTO Customer (Cust_name, Cust_dd, Cust_mm, Cust_yy,
604 Cust_add, Cust_coun, Cust_bank, Cust_mob, Cust_pass) VALUES (%s,%s, %s,%s,%s,%s,%s,%s,%s,%s)"
605                 val = (n, d, month, y, a, c, b, mo, p)
606                 mycursor.execute(sql, val)
607                 mydb.commit()
608                 mycursor.execute("SELECT Cust_acc FROM Customer where Cust_name=%s",n)
609                 myresult = mycursor.fetchall()
610                 ver_acc = myresult[0][0]
611                 bal = '20000'
612                 sql = "INSERT INTO Custbank (Cust_acc, Cust_bank, Cust_mob, Cust_pass,
613 Cust_bal) VALUES (%s,%s, %s,%s,%s)"
614                 val = (ver_acc , b, mo,p,bal)
615                 mycursor.execute(sql, val)
616                 mydb.commit()
617                 ra = random_with_N_digits(7)
618                 ra2 = ", "+str(ra)
619                 #
620                 #
621                 ra = str(ra)
622                 telegram_bot_sendtext(ra)
623                 validReg(ver_acc ,ra2)
624                 presentation.current = "customer"
625                 self.nam.text =""
626                 self.dd.text =""
627                 self.m.text =""
628                 self.yy.text =""
629                 self.add.text =""
630                 self.country.text =""
631                 self.bank.text =""
632                 self.mob.text =""

```

```

627             self.pas.text = ""
628 #####Screen for updating password#####
629 class PassupdateScreen(Screen):
630     def verify(self):
631         s = self.nam2.text
632         p = self.ac.text
633         global o
634         mycursor.execute("SELECT * FROM Customer")
635         myresult = mycursor.fetchall()
636         if len(s) == 0 or len(p)==0:
637             invalidForm()
638         else:
639             if s in str(myresult):
640                 mycursor.execute("select Cust_acc from Customer where Cust_name= %s ",s)
641                 myr1 = list(mycursor.fetchall())
642                 v =str(myr1[0][0])
643                 if p == v:
644
645                     otp = random_with_N_digits(4)
646                     o = otp
647                     getotp(o)
648                     presentation.current="Passupdate2"
649                     elif len(p) == 0 :
650                         invalidLogin()
651                     else:
652                         invalidAcc()
653                     else:
654                         invalidCustomer()
655
656             self.nam2.text = ""
657             self.ac.text = ""
658
659
660
661 class PassupdateScreen2(Screen):
662     def fetch(self):
663         s = self.nam2.text
664         p = self.ac.text
665         ot = self.otp.text
666         newp = self.pas1.text
667         mycursor.execute("SELECT * FROM Customer")
668         myresult = mycursor.fetchall()
669         if len(s) == 0 or len(p)==0 or len(ot)==0:
670             invalidForm()
671         else:
672             if s in str(myresult):
673                 mycursor.execute("select Cust_acc from Customer where Cust_name= %s ",s)
674                 myr1 = list(mycursor.fetchall())
675                 v =str(myr1[0][0])
676                 oc = str(o)
677                 if p == v and ot==oc:
678                     print(o)
679                     sql1 = "UPDATE Customer SET Cust_pass = %s where Cust_name = %s"
680                     val1 = (newp,s)
681                     mycursor.execute(sql1 ,val1)
682                     mydb.commit()
683                     updatesuccess()
684                     presentation.current="customer"
685                     elif ot!=oc:
686                         invalidAcc2()
687                     elif len(p) == 0 or len(ot)==0 :
688                         invalidLogin()
689                     else:
690                         invalidCustomer()
691

```

```

692     self.nam2.text = ""
693     self.ac.text = ""
694     self.otp.text = ""
695     self.pas1.text = ""
696
697
698
699 class ScreenManagement(ScreenManager):
700     pass
701
702 #####Connector of kivy#####
703 presentation = Builder.load_file("my.kv")
704
705 class Float_LayoutApp(App):
706     def build(self):
707         return presentation
708 #####MAIN function#####
709 if __name__ == "__main__":
710     #Window.size = (1366, 768)
711     #Window.fullscreen = True
712     Float_LayoutApp().run()
713
714 #####KIVY CODE#####
715 #: import FadeTransition kivy.uix.screenmanager.FadeTransition
716
717 <Button>:
718     font_size: 18
719     size_hint: 0.055, 0.050
720     #font_size: (root.width**2 + root.height**2) / 13**4
721
722 <TextInput>:
723     font_size: 18
724     size_hint: 0.09, 0.040
725
726 <Label>:
727     font_size: 18
728     #color: 0,0,0,1
729
730 ScreenManagement:
731     transition: FadeTransition()
732     MainScreen:
733         EmployeeScreen:
734             EmployeeScreen1:
735             EmployeeScreen2:
736             EmployeeScreen3:
737             CustomerScreen:
738             MenuScreen:
739             MenuScreen2:
740             CreateScreen:
741             PassupdateScreen:
742             PassupdateScreen2:
743             transactionScreen:
744             transferScreen:
745             transferScreen2:
746             deleteScreen:
747 <MainScreen>:
748     name: "main"
749     Image:
750         source: '1.jpg'
751         allow_stretch: True
752         keep_ratio: True
753         size_hint_y: None
754         size_hint_x: None
755         width: self.parent.width
756         height: self.parent.width / self.image_ratio

```

```

757     Image:
758         source: 'RBI.jpeg'
759         allow_stretch: True
760         keep_ratio: True
761         size_hint_y: 0.05
762         size_hint_x: 0.05
763         width: self.parent.width
764         height: self.parent.width/self.image_ratio
765
766     Button:
767         on_release: app.root.current = "main"
768         text: "UNIFIED BANKING ONLINE"
769         font_size: 50
770         color: 10,100,0,1
771         pos_hint: {"right": 0.8, "top":0.95 }
772         #size_hint: 0.04,0.04
773         background_color: (0, 0, 0, 0)
774
775     Button:
776         on_release: app.root.current = "emp"
777         text: "EMPLOYEE"
778         font_size: 30
779         color: 10,100,0,1
780         pos_hint: {"right": 0.94, "y":0.8 }
781         #size_hint: 0.04,0.04
782         background_color: (0, 0, 0, 0)
783
784     Button:
785         on_release: app.root.current = "customer"
786         text: "CUSTOMER"
787         font_size: 30
788         color: 10,100,0,1
789         pos_hint: {"right": 0.94, 'top':0.75}
790         #size_hint: 0.04,0.04
791         background_color: (0, 0, 0, 0)
792 #####
793 <EmployeeScreen>:
794     name:"emp"
795     nam: nam
796     pas: pas
797     Image:
798         source: 'logo2.jpg'
799         width: dp(1900)
800         height: dp(1500)
801         allow_stretch: True
802         keep_ratio: False
803
804     Button:
805         on_release: app.root.current = "main"
806         text: "Home"
807         pos_hint: {"left": 1, 'top':1}
808         background_color: 0,0,0,0
809
810     Label:
811         text:"Employee ID: "
812         font_size: 20
813         pos_hint: {"x":0.2 , "top":1.02}
814     Label:
815         text:"Password: "
816         font_size: 20
817         pos_hint: {"x":0.2 , "top":0.97}
818     TextInput:
819         id: nam
820         multiline: False
821         pos_hint: {"x": 0.75 , "y":0.5}

```

```

822 TextInput:
823     id: pas
824     multiline: False
825     password: True
826     pos_hint: {"x": 0.75, "y":0.45}
827
828 Button:
829     pos_hint:{ "x":0.75 , "y":0.3}
830     text: "LOGIN"
831     font_size: 20
832     on_release:
833         root.manager.transition.direction = "up"
834         root.btn()
835
836 #####<EmployeeScreen1>:
837     name: "emp1"
838
839 Image:
840     source: 'logo2.jpg'
841     width: dp(1900)
842     height: dp(1500)
843     allow_stretch: True
844     keep_ratio: False
845
846 Label:
847     text: "EMPLOYEE PORTAL "
848     pos_hint: {"x":0, "y":0.45}
849     font_size: 40
850     color: 1,1,1,1
851 Button:
852     on_release: app.root.current = "main"
853     text: "Home"
854     pos_hint: {"left": 1, 'top':1}
855     background_color: (0, 0, 0, 0)
856
857 Button:
858     on_release: app.root.current = "emp"
859     text: "Back"
860     pos_hint: {"right": 1, 'top':1}
861     background_color: (0, 0, 0, 0)
862
863 Button:
864     on_release: app.root.current = "emp2"
865     text: "UPDATE CUSTOMER DETAILS"
866     pos_hint: {"x": 0.1 , 'top ':0.7 }
867     font_size: 20
868     background_color: (0, 0, 0, 0)
869
870 Button:
871     on_release:
872         app.root.current = "emp3"
873         root.on_enter()
874         text: "VIEW CUSTOMER DETAILS"
875         pos_hint: {"x": 0.1 , 'top ':0.6 }
876         font_size: 20
877         background_color: (0, 0, 0, 0)
878 #####
879
880 <EmployeeScreen2>:
881     name: "emp2"
882     euac: euac
883     euaddress: euaddress
884     eumob: eumob
885     eudob: eudob
886     Image:

```

```

887     source: '3.jpg'
888     width: dp(1900)
889     height: dp(1500)
890     allow_stretch: True
891     keep_ratio: False
892 Label:
893     text: "UPDATE CUSTOMER DETAILS "
894     pos_hint: {"x":0, "y":0.45}
895     font_size: 40
896     color: 1,1,1,1
897 Button:
898     on_release: app.root.current = "main"
899     text: "Home"
900     pos_hint: {"left": 1, 'top':1}
901     background_color: (0, 0, 0, 0)
902 Button:
903     on_release: app.root.current = "emp1"
904     text: "Back"
905     pos_hint: {"right": 1, 'top':1}
906     background_color: (0, 0, 0, 0)
907 Label:
908     text: "Account No : "
909     pos_hint: {"x":-0.1, "top":1.22}
910 Label:
911     text: "Address : "
912     pos_hint: {"x":-0.1, "top":1.12}
913 Label:
914     text: "Mobile Number : "
915     pos_hint: {"x":-0.1, "top":1.02}
916 Label:
917     text: "Date of Birth : "
918     pos_hint: {"x":-0.1, "top":0.92}
919 TextInput:
920     id: euac
921     multiline: False
922     pos_hint: {"x": 0.5, "y":0.7}
923
924 TextInput:
925     id: euaddress
926     multiline: False
927     pos_hint: {"x": 0.5, "y":0.6}
928 TextInput:
929     id: eumob
930     multiline: False
931     pos_hint: {"x": 0.5, "y":0.5}
932 TextInput:
933     id: eudob
934     multiline: False
935     pos_hint: {"x": 0.5, "y":0.4}
936
937 Button:
938     text: "SUBMIT"
939     pos_hint: {"x": 0.45, 'y':0.25}
940     on_release:
941         root.manager.transition.direction = "up"
942         root.btn1()
943 #####
944
945 <EmployeeScreen3>:
946     name: "emp3"
947     l1:l1
948     l2:l2
949     l4:l4
950     l5:l5
951     l6:l6

```

```

952     17 : 17
953     18 : 18
954     Image:
955         source: '7.jpg'
956         width: dp(1900)
957         height: dp(1500)
958         allow_stretch: True
959         keep_ratio: False
960     Button:
961         on_release: app.root.current = "main"
962         text: "Home"
963         pos_hint: {"left": 1, 'top':1}
964         background_color: (0, 0, 0, 0)
965
966     Button:
967         on_release: app.root.current = "emp1"
968         text: "Back"
969         pos_hint: {"right": 1, 'top':1}
970         background_color: (0, 0, 0, 0)
971
972     Label:
973         text: "CUSTOMER DETAILS "
974         pos_hint: {"x":0, "y":0.45}
975         font_size: 40
976         color: 1,1,1,1
977
978     Label:
979         id:11
980         text: "Account Number "
981         pos_hint: {"x":-0.4, "y":0.35}
982     Label:
983         id:12
984         text: "Name "
985         pos_hint: {"x":-0.3, "y":0.35}
986
987     Label:
988         id:14
989         text: "Address "
990         pos_hint: {"x":-0.1, "y":0.35}
991     Label:
992         id:15
993         text: "Country "
994         pos_hint: {"x":0, "y":0.35}
995     Label:
996         id:16
997         text: "Bank "
998         pos_hint: {"x":0.1, "y":0.35}
999     Label:
1000         id:17
1001         text: "Mobile "
1002         pos_hint: {"x":0.2, "y":0.35}
1003     Label:
1004         id:18
1005         text: "Balance "
1006         pos_hint: {"x":0.3, "y":0.35}
1007
1008 #####
1009
1010
1011
1012
1013 <CustomerScreen>:
1014     name: "customer"
1015     nam: nam
1016     pas: pas

```

```

1017     Image:
1018         source: '4.jpg'
1019         allow_stretch: True
1020         keep_ratio: True
1021             size_hint_y: None
1022             size_hint_x: None
1023             width: self.parent.width
1024             height: self.parent.width/self.image_ratio
1025
1026     Button:
1027         pos: root.right - self.width, root.top - self.height
1028         on_release: app.root.current = "main"
1029             text: "Home"
1030             size_hint: 0.04,0.04
1031             pos_hint: {"right": 1, 'top':1}
1032             font_size: 16
1033             color: 1,0,0,1
1034             background_color: (53, 14,25,0)
1035
1036     Label:
1037         text: "Name: "
1038         font_size: 20
1039         pos_hint: {"x": 0.08 , 'top ':0.8 }
1040         size_hint: (0.04 ,0.04)
1041
1042     TextInput:
1043         id: nam
1044         multiline:False
1045         pos_hint: {"x": 0.13 , 'top ':0.8 }
1046         size_hint: (0.1 ,0.04)
1047
1048     Label:
1049         text: "Password: "
1050         font_size: 20
1051         pos_hint: {"x": 0.08 , 'top ':0.75 }
1052         size_hint: (0.04 ,0.04)
1053
1054     TextInput:
1055         id: pas
1056         password: True
1057         multiline: False
1058         pos_hint: {"x": 0.13 , 'top ':0.75 }
1059         size_hint: (0.1 ,0.04)
1060
1061     Button:
1062         text:"Submit"
1063         font_size: 16
1064         pos_hint: {"x": 0.13 , 'top ':0.7 }
1065         size_hint: (0.06 ,0.04)
1066         on_release:
1067             root.btn()
1068
1069     Button:
1070         text:"Not a User? Click here for New Registration"
1071         font_size: 20
1072         color: 1,0,0,1
1073         pos_hint: {"x": 0.13 , 'top ':0.55 }
1074         size_hint: (0.1 ,0.04)
1075         background_color: (0, 0, 0, 0)
1076         on_release:
1077             app.root.current = "Create"
1078
1079     Button:
1080         text:"Forgot Password?"
1081         font_size: self.size
1082         color: 1,0,0,1
1083         pos_hint: {"x": 0.13 , 'top ':0.63 }
1084         size_hint: (0.1 ,0.04)
1085         background_color: (0, 0, 0, 0)

```

```

1082     on_release:
1083         app.root.current = "Passupdate"
1084
1085 <MenuScreen>:
1086     name: "Customer1"
1087     sbi: sbi
1088     icici: icici
1089     hdfc: hdfc
1090
1091     Image:
1092         source: 'indian-rupe.jpg'
1093         allow_stretch: True
1094         keep_ratio: True
1095         size_hint_y: None
1096         size_hint_x: None
1097         width: self.parent.width
1098         height: self.parent.width / self.image_ratio
1099
1100     Button:
1101         id: sbi
1102         on_release:
1103             root.sbibank()
1104         text: "SBI"
1105         font_size: 30
1106         color: 0,0,0,1
1107         pos_hint: {"x": 0.08, "top":0.75 }
1108         #size_hint: 0.04,0.04
1109         background_color: (0, 0, 0, 0)
1110
1111     Button:
1112         id: icici
1113         on_release:
1114             root.icicibank()
1115         text: "ICICI"
1116         font_size: 30
1117         color: 0,0,0,1
1118         pos_hint: {"x": 0.08, "top":0.65 }
1119         #size_hint: 0.04,0.04
1120         background_color: (0, 0, 0, 0)
1121
1122     Button:
1123         on_release:
1124             root.hdfcbank()
1125         id: hdfc
1126         text: "HDFC"
1127         font_size: 30
1128         color: 0,0,0,1
1129         pos_hint: {"x": 0.08 , 'top ':0.55}
1130         #size_hint: 0.04,0.04
1131         background_color: (0, 0, 0, 0)
1132
1133     Button:
1134         pos:root.right - self.width , root.top - self.height
1135         on_release: app.root.current = "main"
1136         text: "Home"
1137         size_hint: 0.04,0.04
1138         pos_hint: {"right": 1, 'top ':1}
1139         font_size: 16
1140         color: 1,0,0,1
1141         background_color: (53, 14,25,0)
1142
1143     Button:
1144         pos:root.right - self.width , root.top - self.height
1145         on_release: app.root.current = "customer"
1146         text: "Back"

```

```

1147         background_color: (53, 14,25,0)
1148 #####<MenuScreen2>:
1149 <MenuScreen2>:
1150     name: "Customer2"
1151     Image:
1152         source: 'indian-rupe.jpg'
1153         allow_stretch: True
1154         keep_ratio: True
1155         size_hint_y: None
1156         size_hint_x: None
1157         width: self.parent.width
1158         height: self.parent.width/self.image_ratio
1159     Button:
1160         on_release: root.balance()
1161         text: "Current Balance"
1162         font_size: 30
1163         color: 0,0,0,1
1164         pos_hint: {"x": 0.08, 'top':0.8}
1165         #size_hint: 0.04,0.04
1166         background_color: (0, 0, 0, 0)
1167     Button:
1168         on_release: app.root.current = "transaction"
1169         text: "Mini Statement"
1170         font_size: 30
1171         color: 0,0,0,1
1172         pos_hint: {"x": 0.08, 'top':0.7}
1173         #size_hint: 0.04,0.04
1174         background_color: (0, 0, 0, 0)
1175     Button:
1176         on_release: app.root.current = "transfer"
1177         text: "Quick Transfer"
1178         font_size: 30
1179         color: 0,0,0,1
1180         pos_hint: {"x": 0.08, 'top':0.6}
1181         #size_hint: 0.04,0.04
1182         background_color: (0, 0, 0, 0)
1183
1184     Button:
1185         on_release: app.root.current = "delete"
1186         text: "Remove Account"
1187         font_size: 30
1188         color: 0,0,0,1
1189         pos_hint: {"x": 0.08, 'top':0.5}
1190         #size_hint: 0.04,0.04
1191         background_color: (0, 0, 0, 0)
1192     Button:
1193         pos:root.right - self.width, root.top - self.height
1194         on_release: app.root.current = "main"
1195         text: "Home"
1196         size_hint: 0.04,0.04
1197         pos_hint: {"right": 1, 'top':1}
1198         font_size: 16
1199         color: 1,0,0,1
1200         background_color: (53, 14,25,0)
1201     Button:
1202         pos:root.right - self.width, root.top - self.height
1203         on_release: app.root.current = "Customer1"
1204         text: "Back"
1205         size_hint: 0.04,0.04
1206         pos_hint: {"right": 0.97, 'top':1}
1207         font_size: 16
1208         color: 1,0,0,1
1209         background_color: (53, 14,25,0)
1210 #####

```

```

1212 <transactionScreen >:
1213     name: "transaction"
1214     l1 : l1
1215     l2 : l2
1216     l3 : l3
1217     l4 : l4
1218     l5 : l5
1219     l6 : l6
1220     Image:
1221         source: '6.jpg'
1222         width: dp(1900)
1223         height: dp(1500)
1224         allow_stretch: True
1225         keep_ratio: False
1226
1227     Label:
1228         text:" Last five transactions "
1229         pos_hint: {"x":0, "y":0.45}
1230         font_size: 40
1231         color: 0,0,1,1
1232     Button:
1233         on_release: app.root.current = "main"
1234         text: "Home"
1235         pos_hint: {"left": 1, 'top':1}
1236
1237     Button:
1238         on_release: app.root.current = "Customer2"
1239         text: "Back"
1240         pos_hint: {"right": 1, 'top':1}
1241     Label:
1242         id:l1
1243         text:"ID "
1244         pos_hint: {"x":-0.4, "y":0.25}
1245     Label:
1246         id:l2
1247         text:"Account Number "
1248         pos_hint: {"x":-0.3, "y":0.25}
1249     Label:
1250         id:l3
1251         text:"Custumer State "
1252         pos_hint: {"x":-0.2, "y":0.25}
1253     Label:
1254         id:l4
1255         text:"Amount "
1256         pos_hint: {"x":-0.1, "y":0.25}
1257     Label:
1258         id:l5
1259         text:"Bank "
1260         pos_hint: {"x":0, "y":0.25}
1261     Label:
1262         id:l6
1263         text:"Transfer to "
1264         pos_hint: {"x":0.1, "y":0.25}
1265
1266     Button:
1267         on_release:
1268             root.trans()
1269             text: "SHOW"
1270             font_size: 30
1271             color: 1,1,1,1
1272             pos_hint: {"x": 0.4, 'top':0.25}
1273             size_hint: 0.2,0.07
1274             background_color: (0, 0, 1, 1)
1275 #####  

1276 <transferScreen >

```

```

1277     name: "transfer"
1278     acn: acn
1279     Image:
1280         source: 'indian-rupe.jpg'
1281         allow_stretch: True
1282         keep_ratio: True
1283         size_hint_y: None
1284         size_hint_x: None
1285         width: self.parent.width
1286         height: self.parent.width/self.image_ratio
1287
1288     Button:
1289         pos: root.right - self.width, root.top - self.height
1290         on_release: app.root.current = "main"
1291         text: "Home"
1292         size_hint: 0.04,0.04
1293         pos_hint: {"right": 1, 'top':1}
1294         font_size: 16
1295         color: 1,0,0,1
1296         background_color: (53, 14,25,0)
1297     Button:
1298         pos: root.right - self.width, root.top - self.height
1299         on_release: app.root.current = "Customer2"
1300         text: "Back"
1301         size_hint: 0.04,0.04
1302         pos_hint: {"right": 0.97, 'top':1}
1303         font_size: 16
1304         color: 1,0,0,1
1305         background_color: (53, 14,25,0)
1306     Button:
1307         text: "Submit"
1308         font_size: 16
1309         pos_hint: {"x": 0.13, 'top':0.7 }
1310         size_hint: (0.06,0.04)
1311         on_release:
1312             root.ver_acc()
1313     Label:
1314         text: "Beneficiary Account number: "
1315         font_size: 16
1316         pos_hint: {"x": 0.08, 'top':0.8 }
1317         size_hint: None, None
1318         size: self.texture_size
1319
1320     TextInput:
1321         id: acn
1322         multiline:False
1323         pos_hint: {"x": 0.2, 'top':0.8 }
1324         size_hint: (0.1,0.04)
1325 #####
1326 <transferScreen2 >
1327     name: "transfer2"
1328     amo: amo
1329     Image:
1330         source: 'indian-rupe.jpg'
1331         allow_stretch: True
1332         keep_ratio: True
1333         size_hint_y: None
1334         size_hint_x: None
1335         width: self.parent.width
1336         height: self.parent.width/self.image_ratio
1337     Button:
1338         pos: root.right - self.width, root.top - self.height
1339         on_release: app.root.current = "main"
1340         text: "Home"
1341         size_hint: 0.04,0.04

```

```

1342     pos_hint: {"right": 1, 'top':1}
1343     font_size: 16
1344     color: 1,0,0,1
1345     background_color: (53, 14,25,0)
1346     Button:
1347         pos:root.right - self.width, root.top - self.height
1348         on_release: app.root.current = "transfer"
1349             text: "Back"
1350             size_hint: 0.04,0.04
1351             pos_hint: {"right": 0.97, 'top':1}
1352             font_size: 16
1353             color: 1,0,0,1
1354             background_color: (53, 14,25,0)
1355     Label:
1356         text: "Enter Amount: "
1357         font_size: 16
1358         pos_hint: {"x": 0.08, 'top':0.50 }
1359         size_hint: (0.04,0.04)
1360
1361     TextInput:
1362         id: amo
1363         multiline:False
1364         pos_hint: {"x": 0.13 , 'top ':0.50 }
1365         size_hint: (0.1,0.04)
1366     Button:
1367         text:"Transfer"
1368         font_size: 16
1369         pos_hint: {"x": 0.13 , 'top ':0.43}
1370         size_hint: (0.06 ,0.04)
1371         on_release:
1372             root.sendmoney()
1373 #####3
1374 <deleteScreen >:
1375     name: "delete"
1376     sbi: sbi
1377     icici: icici
1378     hdfc: hdfc
1379     Image:
1380         source: 'indian-rupe.jpg'
1381         allow_stretch: True
1382         keep_ratio: True
1383         size_hint_y: None
1384         size_hint_x: None
1385         width: self.parent.width
1386         height: self.parent.width/self.image_ratio
1387     Button:
1388         id: sbi
1389         on_release:
1390             root.sbibank()
1391         text: "SBI"
1392         font_size: 30
1393         color: 0,0,0,1
1394         pos_hint: {"x": 0.08, "top":0.75 }
1395         #size_hint: 0.04,0.04
1396         background_color: (0, 0, 0, 0)
1397     Button:
1398         id: icici
1399         on_release:
1400             root.icicibank()
1401         text: "ICICI"
1402         font_size: 30
1403         color: 0,0,0,1
1404         pos_hint: {"x": 0.08, "top":0.65 }
1405         #size_hint: 0.04,0.04
1406         background_color: (0, 0, 0, 0)

```

```

1407
1408     Button:
1409         on_release:
1410             root.hdfcbank()
1411         id: hdfc
1412         text: "HDFC"
1413         font_size: 30
1414         color: 0,0,0,1
1415         pos_hint: {"x": 0.08, "top":0.55 }
1416         #size_hint: 0.04,0.04
1417         background_color: (0, 0, 0, 0)
1418     Button:
1419         pos:root.right - self.width, root.top - self.height
1420         on_release: app.root.current = "main"
1421         text: "Home"
1422         size_hint: 0.04,0.04
1423         pos_hint: {"right": 1, 'top':1}
1424         font_size: 16
1425         color: 1,0,0,1
1426         background_color: (53, 14,25,0)
1427     Button:
1428         pos:root.right - self.width, root.top - self.height
1429         on_release: app.root.current = "Customer2"
1430         text: "Back"
1431         size_hint: 0.04,0.04
1432         pos_hint: {"right": 0.97, 'top':1}
1433         font_size: 16
1434         color: 1,0,0,1
1435         background_color: (53, 14,25,0)
1436
1437
1438 #####CreateScreen#####
1439
1440 <CreateScreen >
1441     name: "Create"
1442     nam: nam
1443     dd: dd
1444     m: m
1445     yy: yy
1446     add: add
1447     country: country
1448     bank: bank
1449     mob: mob
1450     pas: pas
1451     Image:
1452         source: 'createaccount.jpeg',
1453         allow_stretch: True
1454         keep_ratio: True
1455         size_hint_y: None
1456         size_hint_x: None
1457         width: self.parent.width
1458         height: self.parent.width/self.image_ratio
1459     Button:
1460         pos:root.right - self.width, root.top - self.height
1461         on_release: app.root.current = "main"
1462         text: "Home"
1463         size_hint: 0.04,0.04
1464         pos_hint: {"right": 1, 'top':1}
1465         font_size: 16
1466         color: 1,0,0,1
1467         background_color: (53, 14,25,0)
1468     Button:
1469         pos:root.right - self.width, root.top - self.height
1470         on_release: app.root.current = "customer"
1471         text: "Back"

```

```

1472     size_hint: 0.04,0.04
1473     pos_hint: {"right": 0.97, 'top':1}
1474     font_size: 16
1475     color: 1,0,0,1
1476     background_color: (53, 14,25,0)
1477
1478 Label:
1479     text: "Name: "
1480     font_size: 16
1481     pos_hint: {"x": 0.08, 'top ':0.8 }
1482     size_hint: None, None
1483         size: self.texture_size
1484
1485 TextInput:
1486     id: nam
1487     multiline:False
1488     pos_hint: {"x": 0.13, 'top ':0.8 }
1489     size_hint: (0.1,0.04)
1490 Label:
1491     text: "DOB(dd-mm-yy): "
1492     font_size: 16
1493     pos_hint: {"x": 0.08, 'top ':0.75 }
1494     size_hint: (0.04,0.04)
1495
1496 TextInput:
1497     id: dd
1498     multiline:False
1499     pos_hint: {"x": 0.14, 'top ':0.75 }
1500     size_hint: (0.02,0.04)
1501 Label:
1502     text: "—"
1503     font_size: 16
1504     pos_hint: {"x": 0.146, 'top ':0.75 }
1505     size_hint: (0.04,0.04)
1506
1507 TextInput:
1508     id: m
1509     multiline:False
1510     pos_hint: {"x": 0.17, 'top ':0.75 }
1511     size_hint: (0.02,0.04)
1512 Label:
1513     text: "—"
1514     font_size: 16
1515     pos_hint: {"x": 0.176, 'top ':0.75 }
1516     size_hint: (0.04,0.04)
1517
1518 TextInput:
1519     id: yy
1520     multiline:False
1521     pos_hint: {"x": 0.20, 'top ':0.75 }
1522     size_hint: (0.03,0.04)
1523 Label:
1524     text: "Address: "
1525     font_size: 16
1526     pos_hint: {"x": 0.08, 'top ':0.7 }
1527     size_hint: (0.04,0.04)
1528
1529 TextInput:
1530     id: add
1531     multiline:False
1532     pos_hint: {"x": 0.13, 'top ':0.7 }
1533     size_hint: (0.1,0.04)
1534 Label:
1535     text: "Country: "
1536     font_size: 16

```

```

1537     pos_hint: {"x": 0.08, 'top':0.65 }
1538     size_hint: (0.04,0.04)
1539
1540     TextInput:
1541         id: country
1542         multiline:False
1543         pos_hint: {"x": 0.13, 'top':0.65}
1544         size_hint: (0.1,0.04)
1545     Label:
1546         text: "Bank: "
1547         font_size: 16
1548         pos_hint: {"x": 0.08, 'top':0.6 }
1549         size_hint: (0.04,0.04)
1550
1551     TextInput:
1552         id: bank
1553         multiline:False
1554         pos_hint: {"x": 0.13, 'top':0.6 }
1555         size_hint: (0.1,0.04)
1556     Label:
1557         text: "Mobile No: "
1558         font_size: 16
1559         pos_hint: {"x": 0.08, 'top':0.55 }
1560         size_hint: (0.04,0.04)
1561
1562     TextInput:
1563         id: mob
1564         multiline:False
1565         pos_hint: {"x": 0.13, 'top':0.55 }
1566         size_hint: (0.1,0.04)
1567     Label:
1568         text: "Password: "
1569         font_size: 16
1570         password: True
1571         pos_hint: {"x": 0.08, 'top':0.50 }
1572         size_hint: (0.04,0.04)
1573
1574     TextInput:
1575         id: pas
1576         multiline:False
1577         password: True
1578         pos_hint: {"x": 0.13, 'top':0.50 }
1579         size_hint: (0.1,0.04)
1580     Button:
1581         text:"Submit"
1582         font_size: 16
1583         pos_hint: {"x": 0.13, 'top':0.43}
1584         size_hint: (0.06,0.04)
1585         on_release:
1586             root.fetch()
1587 #####
1588
1589 <PassupdateScreen >:
1590     name: "Passupdate"
1591     nam2: nam2
1592     ac: ac
1593     Image:
1594         source: 'createaccount.jpeg'
1595         allow_stretch: True
1596         keep_ratio: True
1597         size_hint_y: None
1598         size_hint_x: None
1599         width: self.parent.width
1600         height: self.parent.width/self.image_ratio
1601

```

```

1602 Label:
1603     text: "Name: "
1604     font_size: 16
1605     pos_hint: {"x": 0.08, 'top':0.8 }
1606     size_hint: None, None
1607     size: self.texture_size
1608
1609 TextInput:
1610     id: nam2
1611     multiline:False
1612     pos_hint: {"x": 0.13, 'top':0.8 }
1613     size_hint: (0.1,0.04)
1614
1615 Label:
1616     text: "Account No: "
1617     font_size: 16
1618     pos_hint: {"x": 0.08, 'top':0.70 }
1619     size_hint: (0.04,0.04)
1620
1621 TextInput:
1622     id: ac
1623     multiline:False
1624     pos_hint: {"x": 0.13, 'top':0.70 }
1625     size_hint: (0.1,0.04)
1626
1627 Button:
1628     text:"Get OTP"
1629     font_size: 16
1630     pos_hint: {"x": 0.13, 'top':0.60}
1631     size_hint: (0.06,0.04)
1632     on_release:
1633         root.verify()
1634 Button:
1635     pos:root.right - self.width, root.top - self.height
1636     on_release: app.root.current = "main"
1637         text: "Home"
1638         size_hint: 0.04,0.04
1639         pos_hint: {"right": 1, 'top':1}
1640         font_size: 16
1641         color: 1,0,0,1
1642         background_color: (53, 14,25,0)
1643 Button:
1644     pos:root.right - self.width, root.top - self.height
1645     on_release: app.root.current = "customer"
1646         text: "Back"
1647         size_hint: 0.04,0.04
1648         pos_hint: {"right": 0.97, 'top':1}
1649         font_size: 16
1650         color: 1,0,0,1
1651         background_color: (53, 14,25,0)
1652 #####<PassupdateScreen2>:
1653 <PassupdateScreen2>
1654     name: "Passupdate2"
1655     nam2: nam2
1656     otp: otp
1657     ac: ac
1658     pas1: pas1
1659 Image:
1660     source: 'createaccount.jpeg',
1661     allow_stretch: True
1662     keep_ratio: True
1663     size_hint_y: None
1664     size_hint_x: None
1665     width: self.parent.width
1666     height: self.parent.width/self.image_ratio

```

```

1667 Label:
1668     text: "User Name: "
1669     font_size: 16
1670     pos_hint: {"x": 0.08, 'top':0.8 }
1671     size_hint: None, None
1672         size: self.texture_size
1673
1674 TextInput:
1675     id: nam2
1676     multiline:False
1677     pos_hint: {"x": 0.13, 'top':0.8 }
1678     size_hint: (0.1,0.04)
1679
1680 Label:
1681     text: "Account No: "
1682     font_size: 16
1683     pos_hint: {"x": 0.08, 'top':0.70 }
1684     size_hint: (0.04,0.04)
1685
1686 TextInput:
1687     id: ac
1688     multiline:False
1689     pos_hint: {"x": 0.13, 'top':0.70 }
1690     size_hint: (0.1,0.04)
1691 Label:
1692     text: "Enter OTP: "
1693     font_size: 16
1694     pos_hint: {"x": 0.08, 'top':0.60 }
1695     size_hint: (0.04,0.04)
1696
1697 TextInput:
1698     id: otp
1699     multiline:False
1700     pos_hint: {"x": 0.13, 'top':0.60 }
1701     size_hint: (0.1,0.04)
1702 Label:
1703     text: "New Password: "
1704     font_size: 16
1705     pos_hint: {"x": 0.08, 'top':0.50 }
1706     size_hint: (0.04,0.04)
1707
1708 TextInput:
1709     id: pas1
1710     multiline:False
1711     pos_hint: {"x": 0.13, 'top':0.50 }
1712     size_hint: (0.1,0.04)
1713 Button:
1714     text:"Update Password"
1715     font_size: 16
1716     pos_hint: {"x": 0.13, 'top':0.40}
1717     size_hint: (0.06,0.04)
1718     on_release:
1719         root.fetch()
1720 Button:
1721     pos:root.right - self.width , root.top - self.height
1722     on_release: app.root.current = "main"
1723         text: "Home"
1724         size_hint: 0.04,0.04
1725         pos_hint: {"right": 1, 'top':1}
1726         font_size: 16
1727         color: 1,0,0,1
1728         background_color: (53, 14,25,0)
1729 Button:
1730     pos:root.right - self.width , root.top - self.height
1731

```

```
1732     on_release: app.root.current = "Passupdate"
1733     text: "Back"
1734     size_hint: 0.04,0.04
1735     pos_hint: {"right": 0.97, 'top':1}
1736     font_size: 16
1737     color: 1,0,0,1
1738     background_color: (53, 14,25,0)
1739
```

References

- [1] *Introduction to Kivy ; A Cross-platform Python Framework.* <https://www.geeksforgeeks.org/introduction-to-kivy/>.
- [2] *Kivy Documentation.* <https://buildmedia.readthedocs.org/media/pdf/kivy/latest/kivy.pdf>.
- [3] *Kivy framework.* <https://kivy.org/doc/stable/api-kivy.html>.
- [4] *KIVY-Interactive Applications and Games in Python.* https://drive.google.com/drive/folders/0B010dD5Y_iESR1ozc3FjMzNzbnM.
- [5] *Kivy master.* <https://kivy.readthedocs.io/en/master/api-kivy.core.image.html>.
- [6] *Kivy Tutorials.* <https://techwithtim.net/tutorials/kivy-tutorial/>.
- [7] *MySQL 8.0 Reference Manual.* <https://dev.mysql.com/doc/refman/8.0/en/>.
- [8] *Python Kivy tutorial – Build desktop GUI apps using Python.* <https://likegeeks.com/kivy-tutorial/>.
- [9] *Python kivy.uix.image.Image() Examples.* <https://www.programcreek.com/python/example/98770/kivy.uix.image.Image/>.
- [10] sentdex. *Kivy Application Development.* https://www.youtube.com/playlist?list=PLQVvva0QuDe_16XiJ4OyGTEqIKugAdTy.