Object-Oriented Software Engineering

Instructor : Huang, Chuen-Min

Teamwork1 ver.1

Group 4

ID	Name
B10523038	Edward
B10523024	Steven
B10523032	Xavier
B10523033	Wing
B10523021	Johnny
B10523037	Yee
B10523006	Peggy
B10523007	Bess
B10523005	Aliss
B10523056	Sandy

Date 2019/ 11 / 10

Context

1) To have a user-friendly GUI, please create a Window containing a set of pane objects. It may include TextPane, ListBox, Button, and so on. Display apply Façade, Mediator to design the GUI. In addition to this, please also apply another 3 patterns to your project. 0

2) You need to evaluate the pieces of design quality by using object-oriented quality metrics (WMC, DIT, NOC, CBO, RFC, LCOM). The figure shall be drawn like the provided references below. You shall explain each metric by giving examples of your design. 5

WMC 5

DIT 5

NOC 6

CBO 7

RFC 7

LCOM 8

3) Create Junit test cases and Junit test suite to test one selected class. 8

4) Conduct part of the software testing including white box and black box. 10

White box testing 10

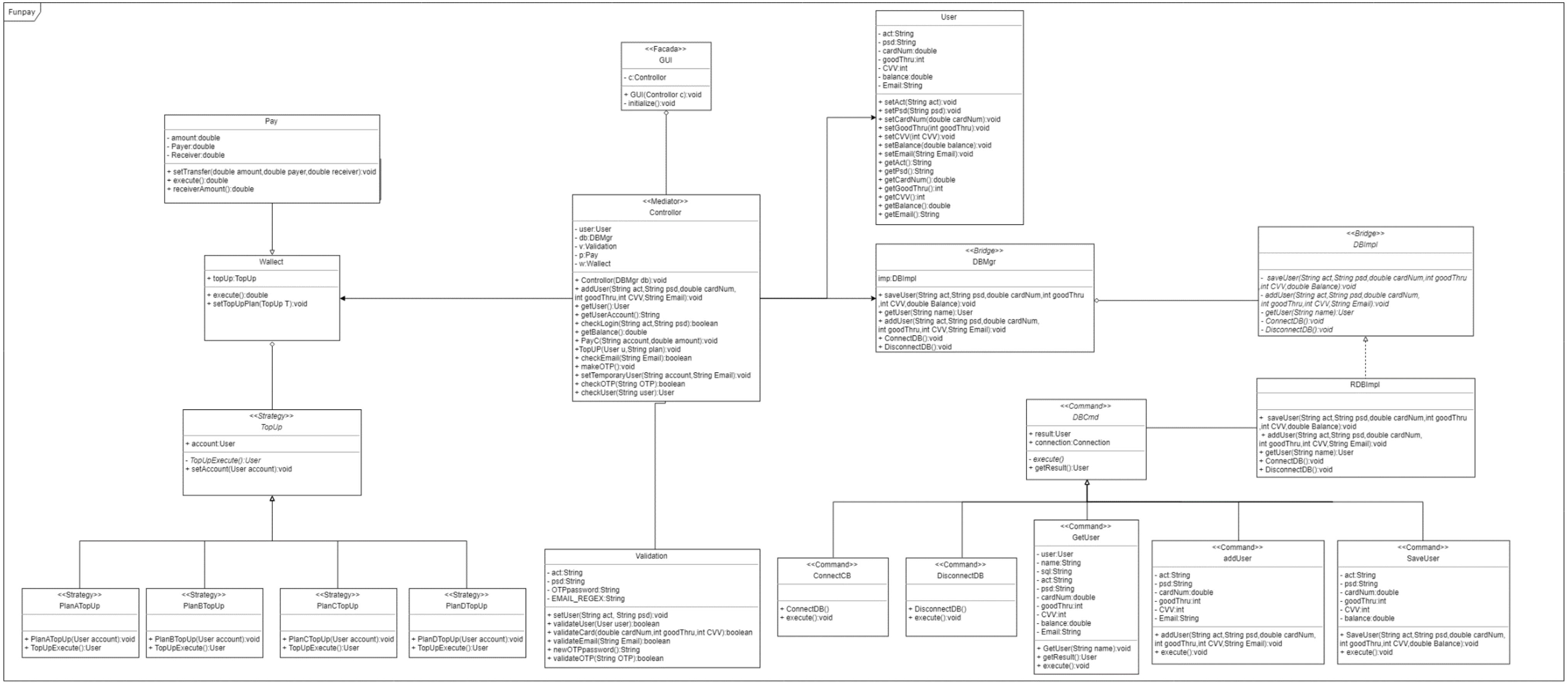
Black box testing 13

5) Please analyze the invocation chains of your design. 15

6) Based on the metrics result, please identify places that need improvement, Selecting an improvement strategy. 40

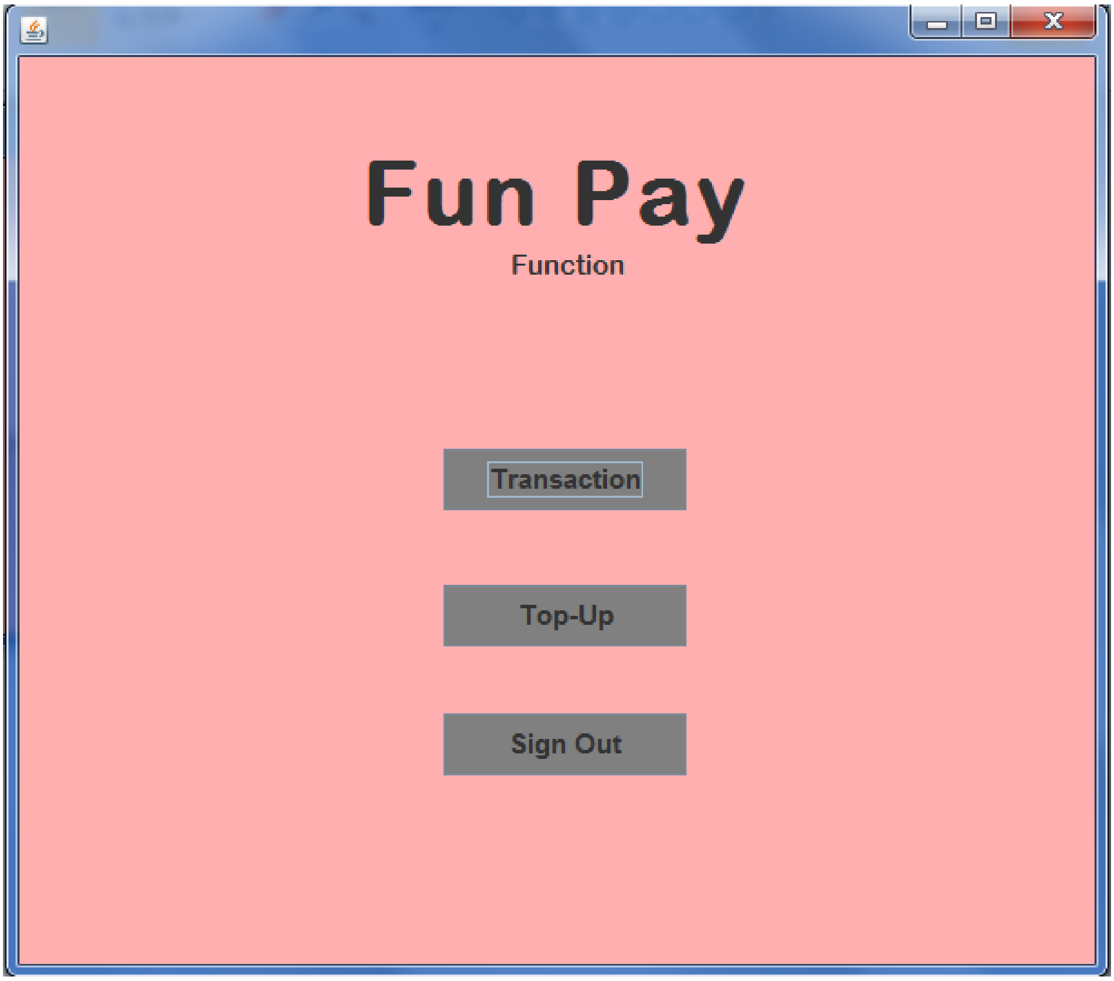
Participation 42

1) To have a user-friendly GUI, please create a Window containing a set of pane objects. It may include TextPane, ListBox, Button, and so on. Display apply Façade, Mediator to design the GUI. In addition to this, please also apply another 3 patterns to your project.

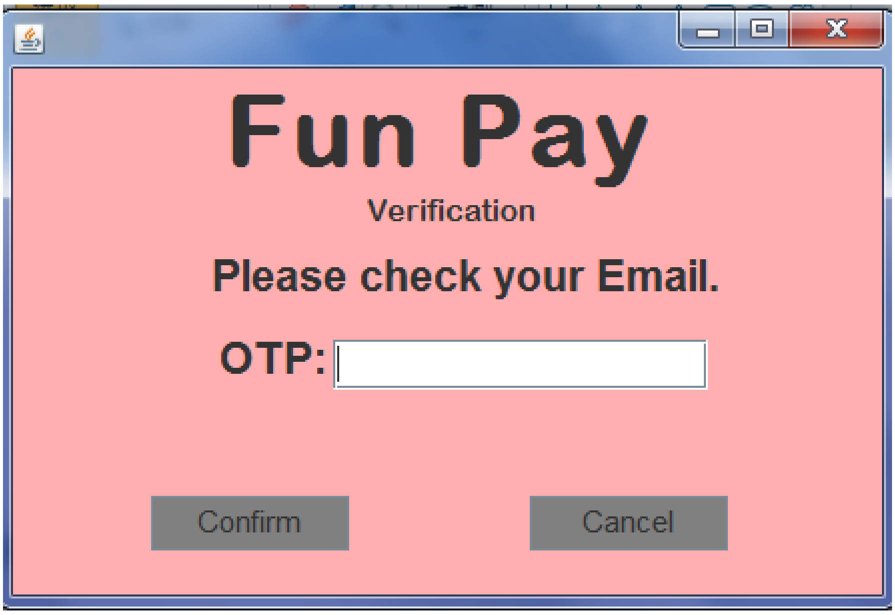


Funpay
<<Facada>>
GUI
- c:Controllor
+ GUI(Controllor c):void
- initialize():void
Pay
- amount:double
- Payer:double
- Receiver:double
+ setTransfer(double amount,double payer,double receiver):void
+ execute():double
+ receiverAmount():double
<<Mediator>>
Controllor
- user:User
- db:DBMgr
- v:Validation
- p:Pay
- w:Wallect
+ Controllor(DBMgr db):void
+ addUser(String act,String psd,double cardNum, int goodThru,int CVV,String Email):void
+ getUser():User
+ getUserAccount():String
+ checkLogin(String act,String psd):boolean
+ getBalance():double
+ PayC(String account,double amount):void
+TopUP(User u,String plan):void
+ checkEmail(String Email):boolean
+ makeOTP():void
+ setTemporaryUser(String account,String Email):void
+ checkOTP(String OTP):boolean
+ checkUser(String user):User
Wallect
+ topUp:TopUp
+ execute():double
+ setTopUpPlan(TopUp T):void
User
- act:String
- psd:String
- cardNum:double
- goodThru:int
- CVV:int
- balance:double
- Email:String
+ setAct(String act):void
+ setPsd(String psd):void
+ setCardNum(double cardNum):void
+ setGoodThru(int goodThru):void
+ setCVV(int CVV):void
+ setBalance(double balance):void
+ setEmail(String Email):void
+ getAct():String
+ getPsd():String
+ getCardNum():double
+ getGoodThru():int
+ getCVV():int
+ getBalance():double
+ getEmail():String
<<Bridge>>
DBMgr
imp:DBImpl
+ saveUser(String act,String psd,double cardNum,int goodThru ,int CVV,double Balance):void
+ getUser(String name):User
+ addUser(String act,String psd,double cardNum, int goodThru,int CVV,String Email):void
+ ConnectDB():void
+ DisconnectDB():void
<<Bridge>>
DBImpl
- saveUser(String act,String psd,double cardNum,int goodThru ,int CVV,double Balance):void
- addUser(String act,String psd,double cardNum, int goodThru,int CVV,String Email):void
- getUser(String name):User
- ConnectDB():void
- DisconnectDB():void
RDBImpl
+ saveUser(String act,String psd,double cardNum,int goodThru ,int CVV,double Balance):void
+ addUser(String act,String psd,double cardNum, int goodThru,int CVV,String Email):void
+ getUser(String name):User
+ ConnectDB():void
+ DisconnectDB():void
<<Strategy>>
TopUp
+ account:User
- TopUpExecute():User
+ setAccount(User account):void
<<Command>>
DBCmd
+ result:User
+ connection:Connection
- execute()
+ getResult():User
<<Strategy>>
PlanATopUp
+ PlanATopUp(User account):void
+ TopUpExecute():User
<<Strategy>>
PlanBTopUp
+ PlanBTopUp(User account):void
+ TopUpExecute():User
<<Strategy>>
PlanCTopUp
+ PlanCTopUp(User account):void
+ TopUpExecute():User
<<Strategy>>
PlanDTopUp
+ PlanDTopUp(User account):void
+ TopUpExecute():User
Validation
- act:String
- psd:String
- OTPpassword:String
- EMAIL_REGEX:String
+ setUser(String act, String psd):void
+ validateUser(User user):boolean
+ validateCard(double cardNum,int goodThru,int CVV):boolean
+ validateEmail(String Email):boolean
+ newOTPpassword():String
+ validateOTP(String OTP):boolean
<<Command>>
ConnectCB
+ ConnectDB()
+ execute():void
<<Command>>
DisconnectDB
+ DisconnectDB()
+ execute():void
<<Command>>
GetUser
- user:User
- name:String
- sql:String
- act:String
- psd:String
- cardNum:double
- goodThru:int
- CVV:int
- balance:double
- Email:String
+ GetUser(String name):void
+ getResult():User
+ execute():void
<<Command>>
addUser
- act:String
- psd:String
- cardNum:double
- goodThru:int
- CVV:int
- Email:String
+ addUser(String act,String psd,double cardNum, int goodThru,int CVV,String Email):void
+ execute():void
<<Command>>
SaveUser
- act:String
- psd:String
- cardNum:double
- goodThru:int
- CVV:int
- balance:double
+ SaveUser(String act,String psd,double cardNum, int goodThru,int CVV,double Balance):void
+ execute():void

home



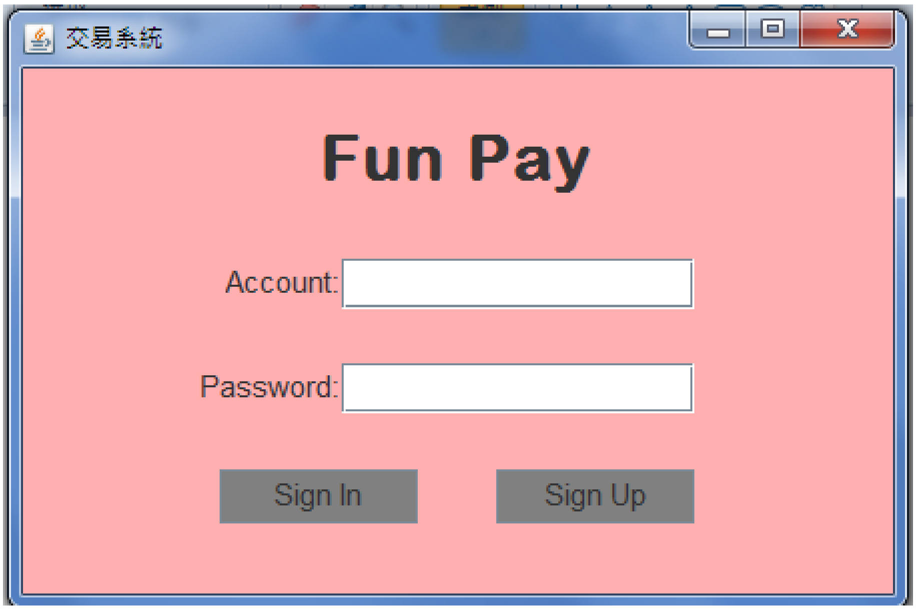
OTPValidation



Please check your Email.

OTP:

signIn



交易系统

Fun Pay

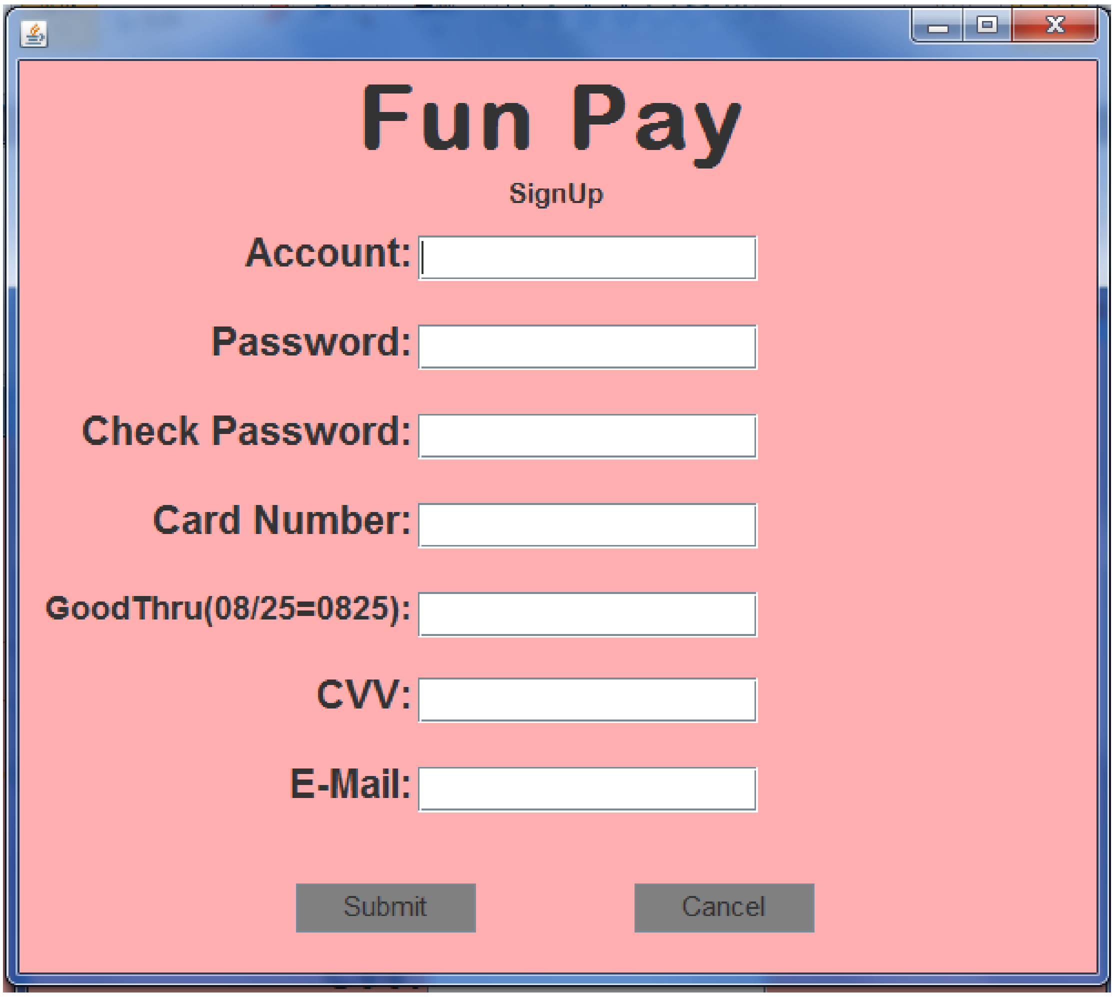
Account:

Password:

Sign In

Sign Up

signUp



Fun Pay

SignUp

Account:

Password:

Check Password:

Card Number:

GoodThru(08/25=0825):

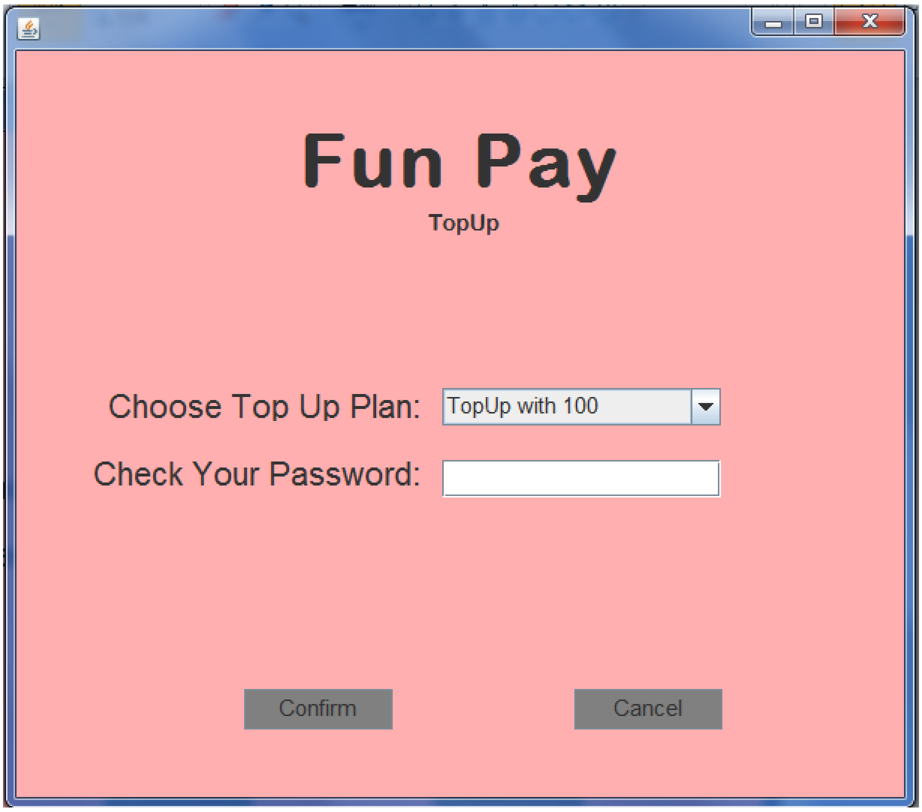
CVV:

E-Mail:

Submit

Cancel

topUp



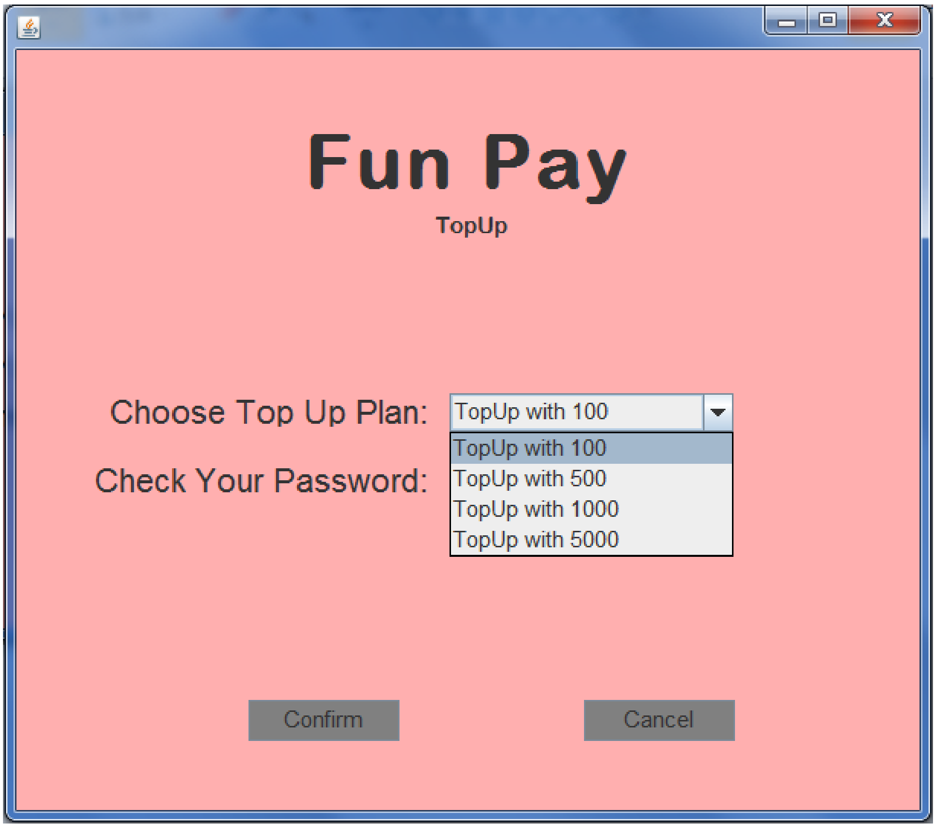
Fun Pay

TopUp

Choose Top Up Plan: TopUp with 100

Check Your Password:

topUp2



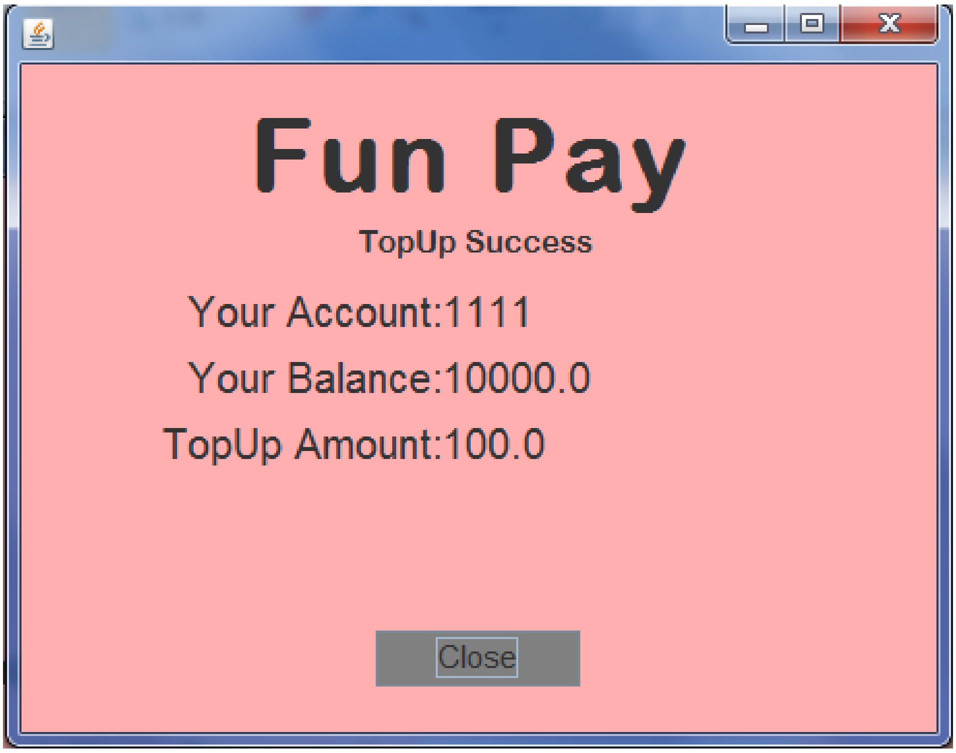
Fun Pay

TopUp

Choose Top Up Plan: TopUp with 100

Check Your Password:

topUpSuccess



Fun Pay

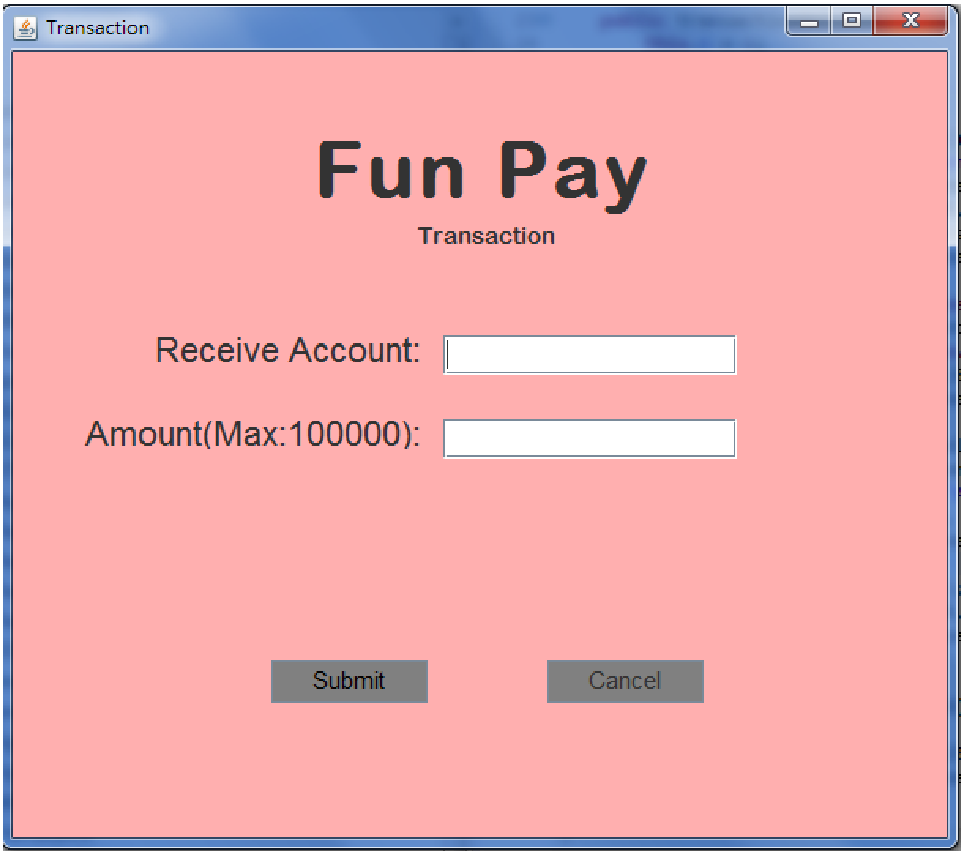
TopUp Success

Your Account:1111

Your Balance:10000.0

TopUp Amount:100.0

transaction



Transaction

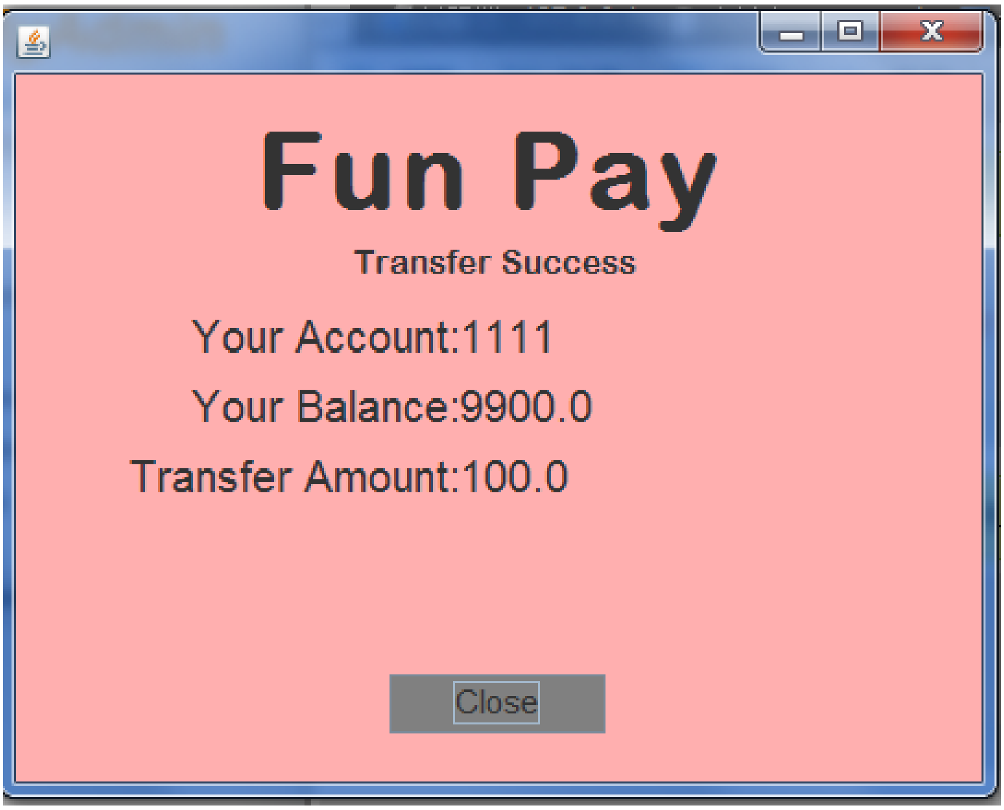
Fun Pay

Transaction

Receive Account:

Amount(Max:100000):

transactionSuccess



Fun Pay

Transfer Success

Your Account:1111

Your Balance:9900.0

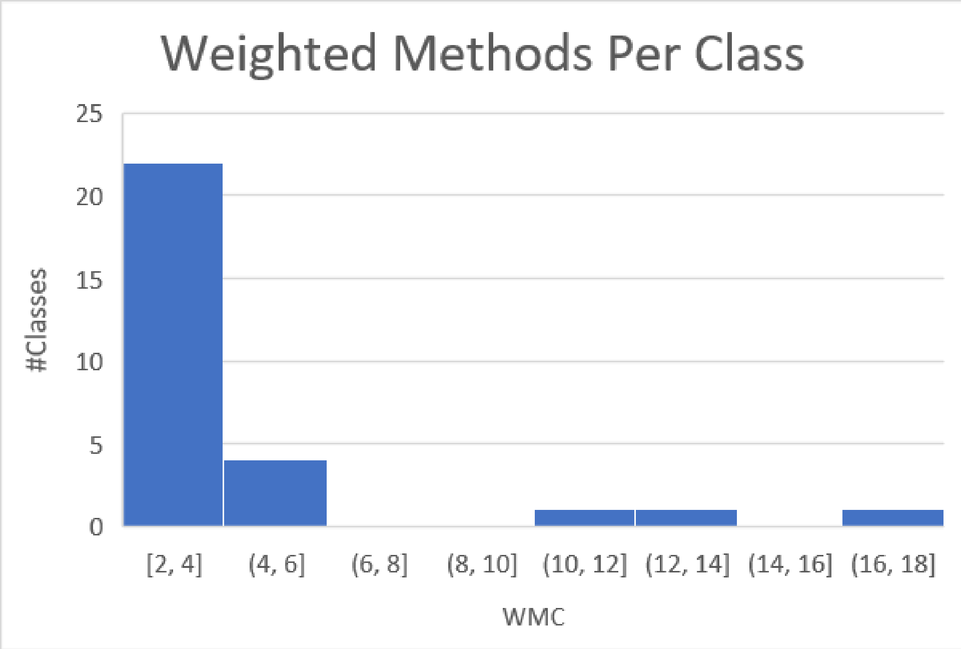
Transfer Amount:100.0

2) **You need to evaluate the pieces of design quality by using object-oriented quality metrics (WMC, DIT, NOC, CBO, RFC, LCOM). The figure shall be drawn like the provided references below. You shall explain each metric by giving examples of your design.**

WMC

WMC relates directly to Bunge's definition of complexity of a thing, since methods are properties of object classes and complexity is determined by the cardinality of its set of properties.

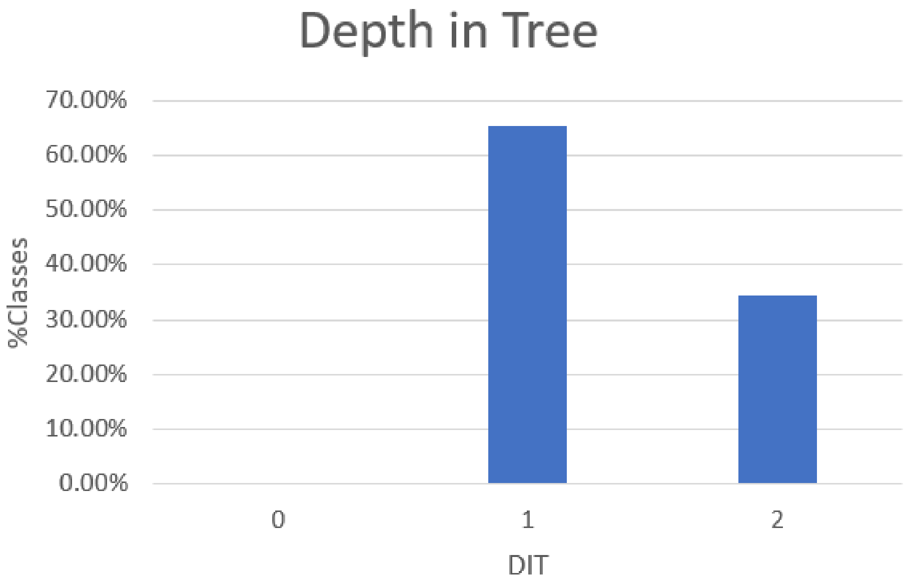
In this metrics, our code is possible to reuse.



DIT

The greater the DIT the more likely the class will inherit and use such method. The behavior of the class is affected and could be more difficult to predict.

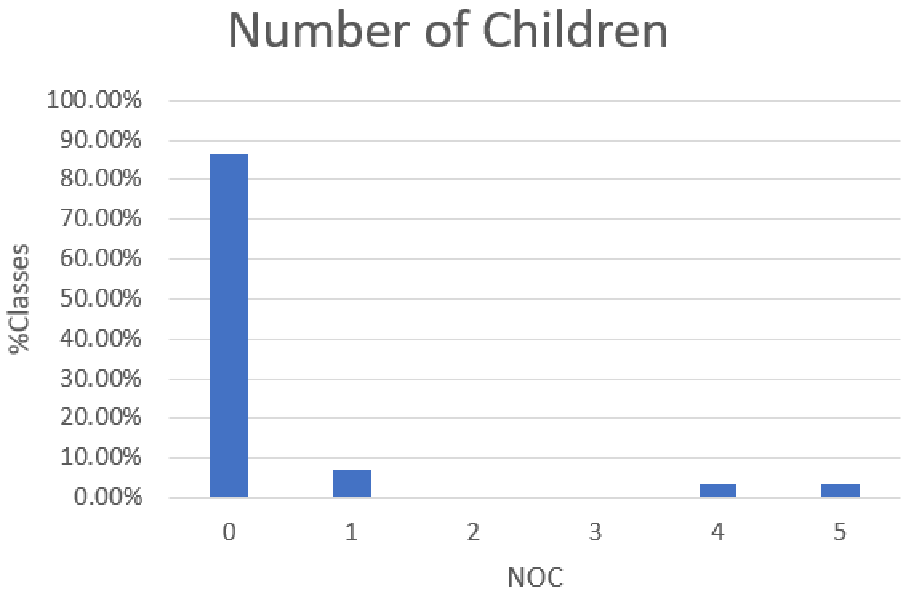
Our DIT almost are 1 and 2, it means our code aren't difficult to predict.



NOC

The greater the NOC the greater likelihood to reusing features.

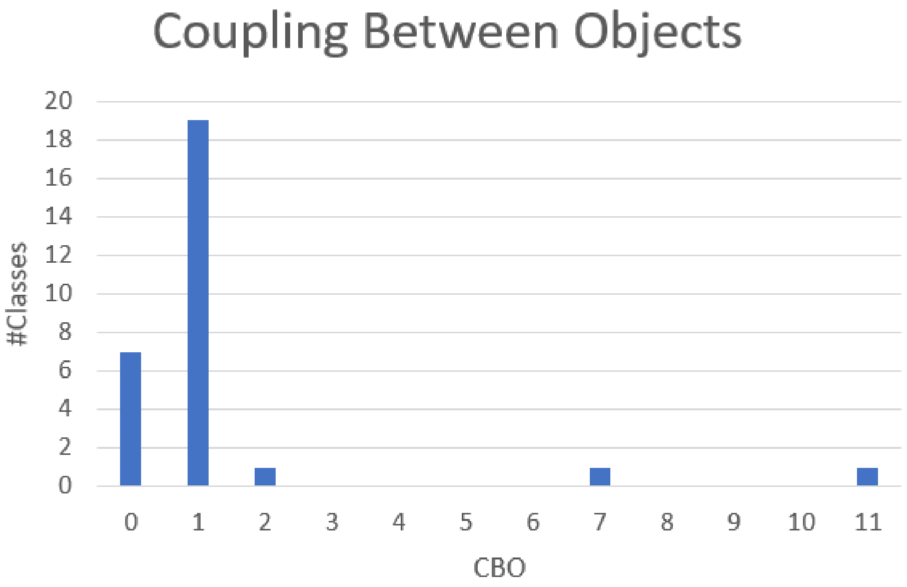
Our NOC within the range of 0 to 2 have 27 , it means our code reusability is not high.



CBO

According to Godin & Miceli article, CBO above 14 is hard to maintain.

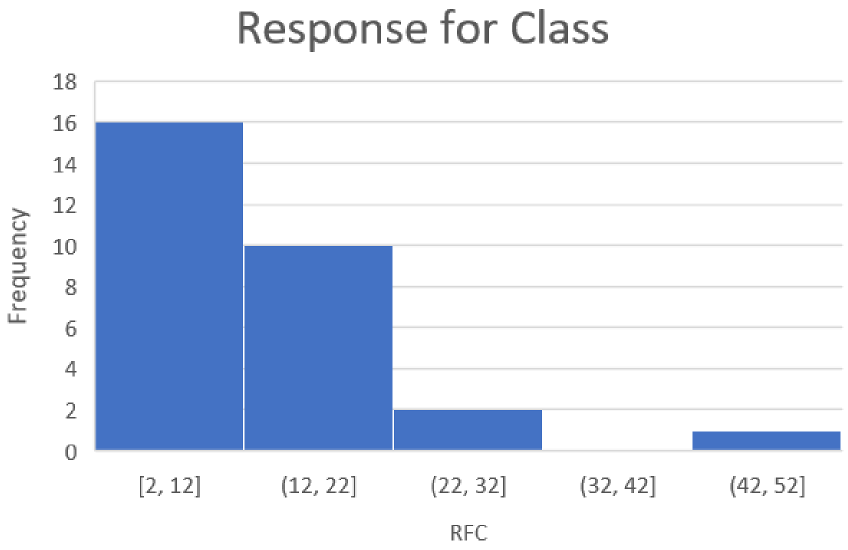
Our highest CBO is 11 and then is 7, most are 1, so our project doesn’t hard to test and reuse.



RFC

The larger the RFC the more difficult to test and debug due to more complex interaction relationships and more effort required to understand the methods and prepare test cases and test stubs.

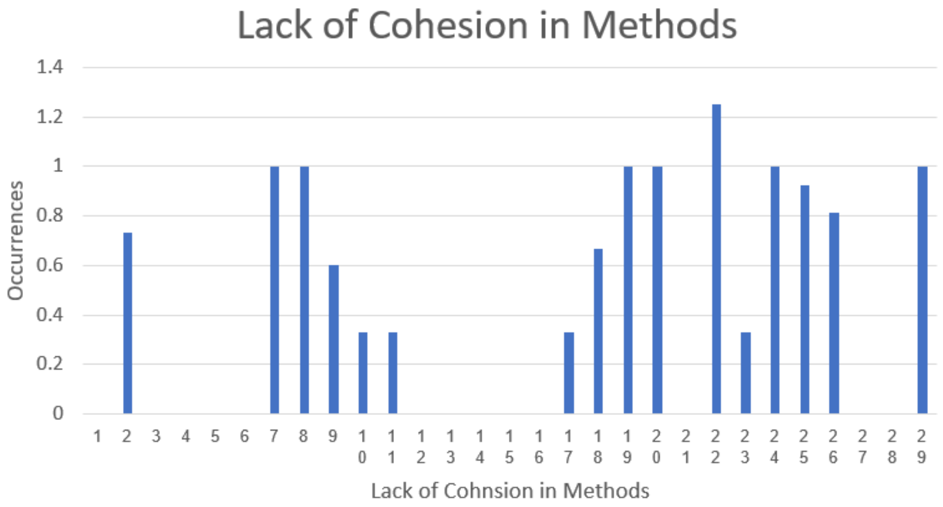
Our RFC highest is Controller, it is 48. It means Controller is hard to test and debug.



LCOM

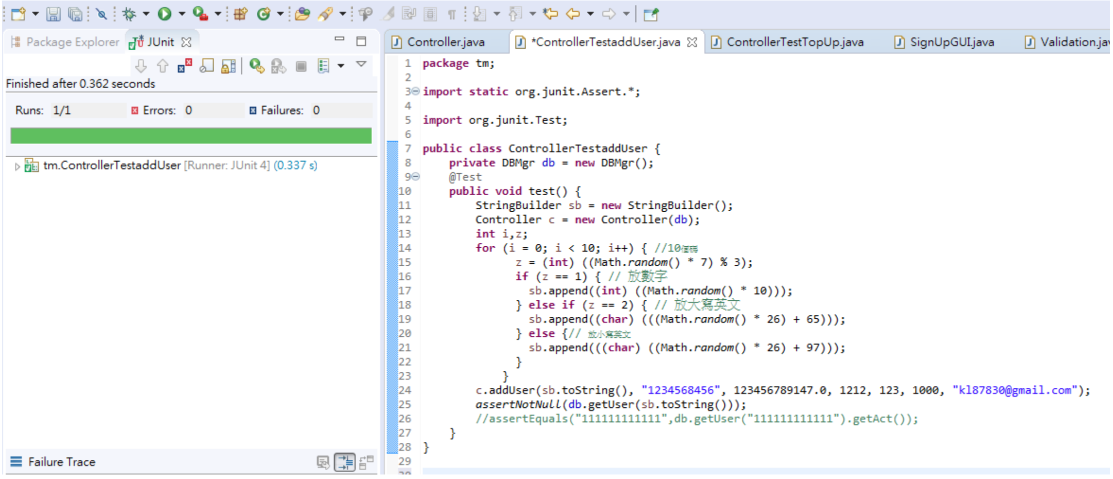
This metric measures the correlation between the methods and the local instance variables of a class. High cohesion indicates good class subdivision. Lack of cohesion or low cohesion increases complexity.

Our LCOM most values are near or under 1, it means that cohesion is normal.



3) Create Junit test cases and Junit test suite to test one selected class.

JUnit Test: Controller – Is add User working or not



Finished after 0.362 seconds

Runs: 1/1 Errors: 0 Failures: 0

tm.ControllerTestaddUser [Runner: JUnit 4] (0.337 s)

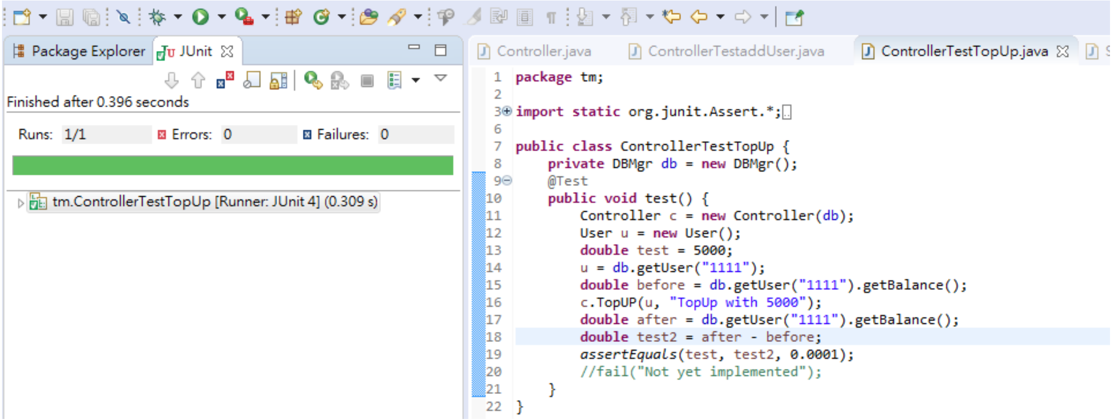
```java
package tm;

import static org.junit.Assert.*;

import org.junit.Test;

public class ControllerTestaddUser {
    private DBMgr db = new DBMgr();
    @Test
    public void test() {
        StringBuilder sb = new StringBuilder();
        Controller c = new Controller(db);
        int i,z;
        for (i = 0; i < 10; i++) { //10個碼
            z = (int) ((Math.random() * 7) % 3);
            if (z == 1) { // 放數字
              sb.append((int) ((Math.random() * 10)));
            } else if (z == 2) { // 放大寫英文
              sb.append((char) (((Math.random() * 26) + 65)));
            } else {// 放小寫英文
              sb.append(((char) ((Math.random() * 26) + 97)));
            }
          }
        c.addUser(sb.toString(), "1234568456", 123456789147.0, 1212, 123, 1000, "kl87830@gmail.com");
        assertNotNull(db.getUser(sb.toString()));
        //assertEquals("11111111111",db.getUser("11111111111").getAct());
    }
}
```

JUnit Test: Controller – Is Top Up working or not



Finished after 0.396 seconds

Runs: 1/1 Errors: 0 Failures: 0

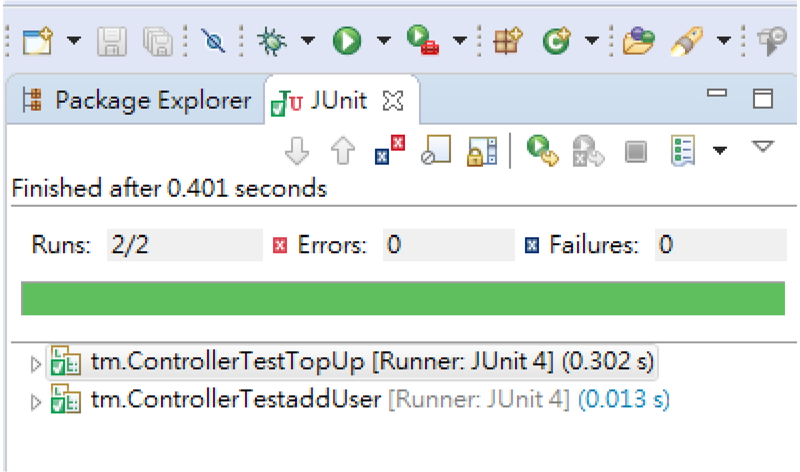
tm.ControllerTestTopUp [Runner: JUnit 4] (0.309 s)

```java
package tm;

import static org.junit.Assert.*;

public class ControllerTestTopUp {
    private DBMgr db = new DBMgr();
    @Test
    public void test() {
        Controller c = new Controller(db);
        User u = new User();
        double test = 5000;
        u = db.getUser("1111");
        double before = db.getUser("1111").getBalance();
        c.TopUP(u, "TopUp with 5000");
        double after = db.getUser("1111").getBalance();
        double test2 = after - before;
        assertEquals(test, test2, 0.0001);
        //fail("Not yet implemented");
    }
}
```

Junit Test Suit – Combine add User and Top Up, is working or not



Finished after 0.401 seconds

Runs: 2/2 Errors: 0 Failures: 0

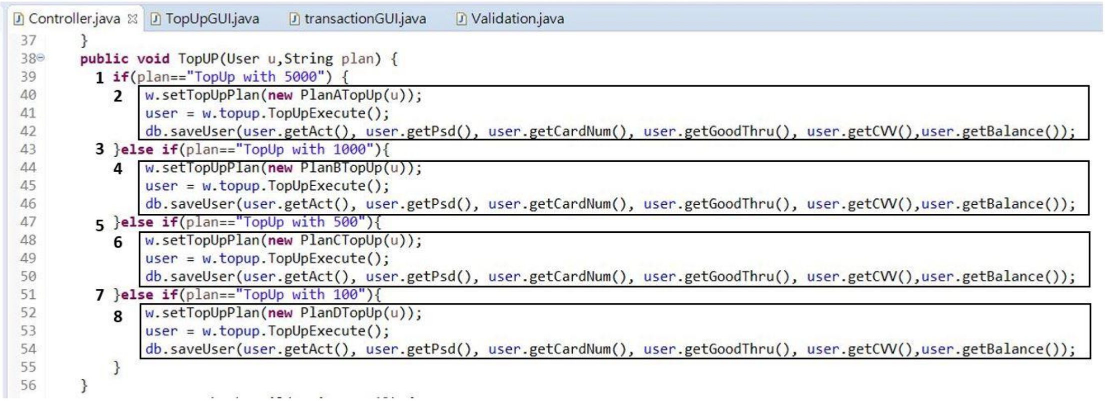
tm.ControllerTestTopUp [Runner: JUnit 4] (0.302 s)
tm.ControllerTestaddUser [Runner: JUnit 4] (0.013 s)

4) Conduct part of the software testing including white box and black box.

White box testing

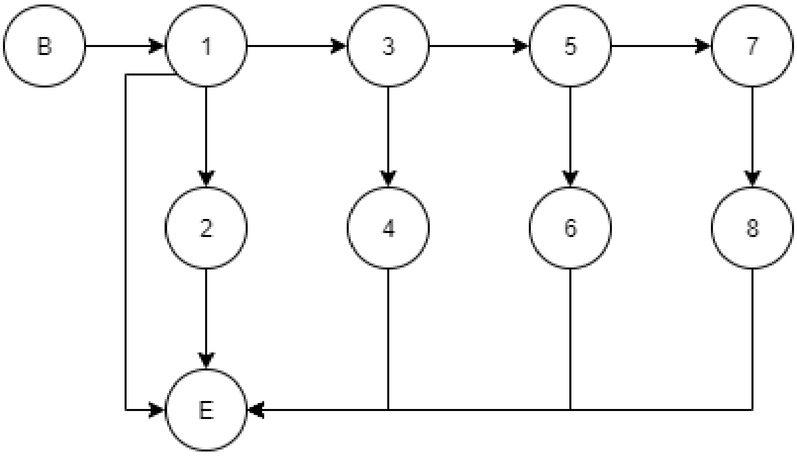
Basis Path Testing

Basis path testing is a technique for designing test cases based on paths. It is based on the program control flow graph. By analyzing the Cyclomatic Complexity of the control structure, the basic executable path set is derived. The test case is designed to ensure that at least each executable statement of the program is executed at least once.



```
    }
    public void TopUP(User u,String plan) {
      1 if(plan=="TopUp with 5000") {
        2  w.setTopUpPlan(new PlanATopUp(u));
           user = w.topup.TopUpExecute();
           db.saveUser(user.getAct(), user.getPsd(), user.getCardNum(), user.getGoodThru(), user.getCVV(),user.getBalance());
      3 }else if(plan=="TopUp with 1000"){
        4  w.setTopUpPlan(new PlanBTopUp(u));
           user = w.topup.TopUpExecute();
           db.saveUser(user.getAct(), user.getPsd(), user.getCardNum(), user.getGoodThru(), user.getCVV(),user.getBalance());
      5 }else if(plan=="TopUp with 500"){
        6  w.setTopUpPlan(new PlanCTopUp(u));
           user = w.topup.TopUpExecute();
           db.saveUser(user.getAct(), user.getPsd(), user.getCardNum(), user.getGoodThru(), user.getCVV(),user.getBalance());
      7 }else if(plan=="TopUp with 100"){
        8  w.setTopUpPlan(new PlanDTopUp(u));
           user = w.topup.TopUpExecute();
           db.saveUser(user.getAct(), user.getPsd(), user.getCardNum(), user.getGoodThru(), user.getCVV(),user.getBalance());
        }
    }
```

Derive Basis Path:



Path1: B→1→E

Path2: B→1→2→E

Path3: B→1→3→4→E

Path4: B→1→3→5→6→E

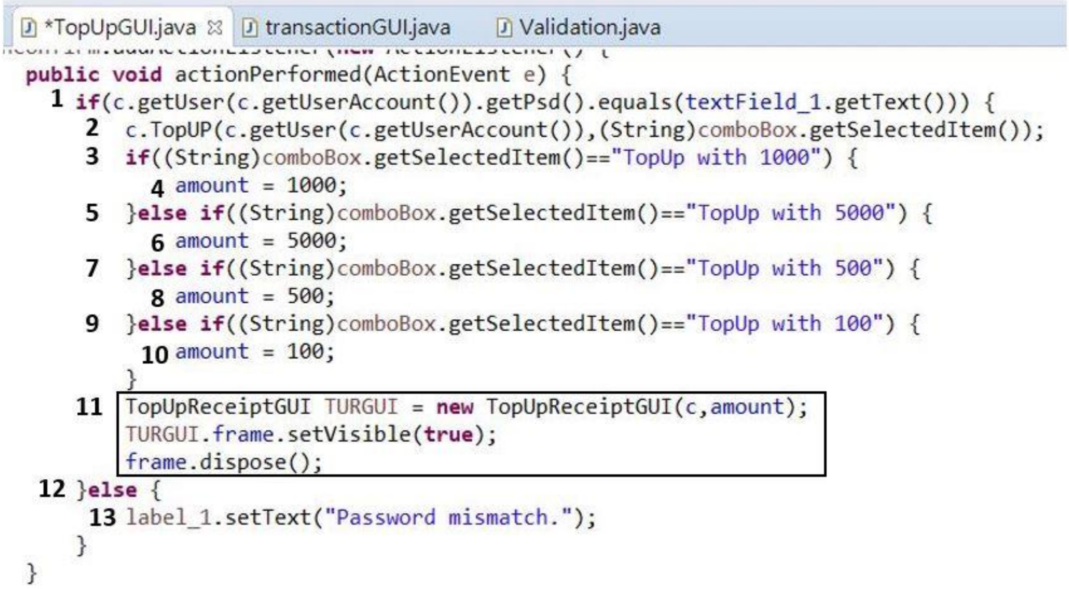
Path5: B→1→3→5→7→8→E

The Cyclomatic Complexity = 5

Number of edges – Number of nodes = 13 – 10 + 2 = 5

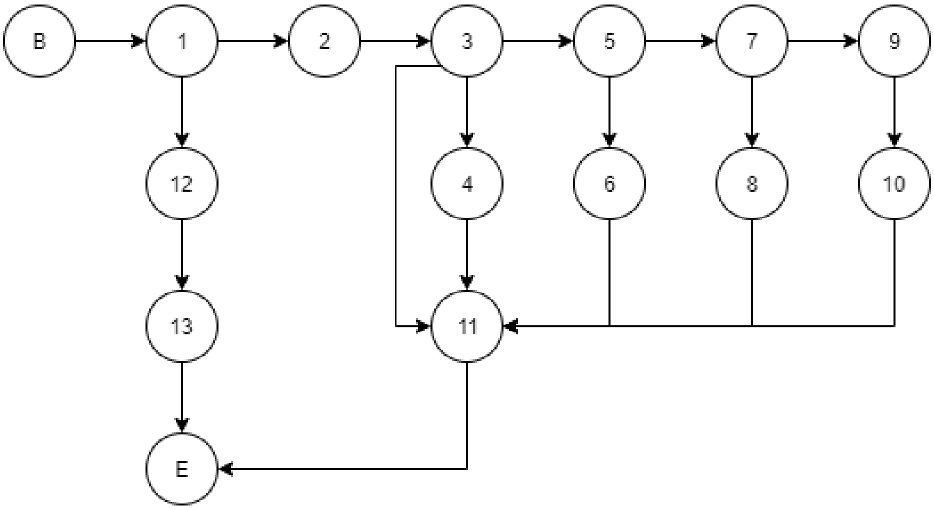
Number of closed region = 4 + 1 = 5

Number of atomic binary predicate = 4 + 1 = 5



```
public void actionPerformed(ActionEvent e) {
  1 if(c.getUser(c.getUserAccount()).getPsd().equals(textField_1.getText())) {
     2  c.TopUP(c.getUser(c.getUserAccount()),(String)comboBox.getSelectedItem());
     3  if((String)comboBox.getSelectedItem()=="TopUp with 1000") {
          4 amount = 1000;
     5  }else if((String)comboBox.getSelectedItem()=="TopUp with 5000") {
          6 amount = 5000;
     7  }else if((String)comboBox.getSelectedItem()=="TopUp with 500") {
          8 amount = 500;
     9  }else if((String)comboBox.getSelectedItem()=="TopUp with 100") {
         10 amount = 100;
        }
    11  TopUpReceiptGUI TURGUI = new TopUpReceiptGUI(c,amount);
        TURGUI.frame.setVisible(true);
        frame.dispose();
 12 }else {
     13 label_1.setText("Password mismatch.");
    }
}
```

Derive Basis Path:



Path1: B→1→12→13→E

Path2: B→1→2→3→11→E

Path3: B→1→2→3→4→11→E

Path4: B→1→2→3→5→6→11→E

Path5: B→1→2→3→5→7→8→11→E

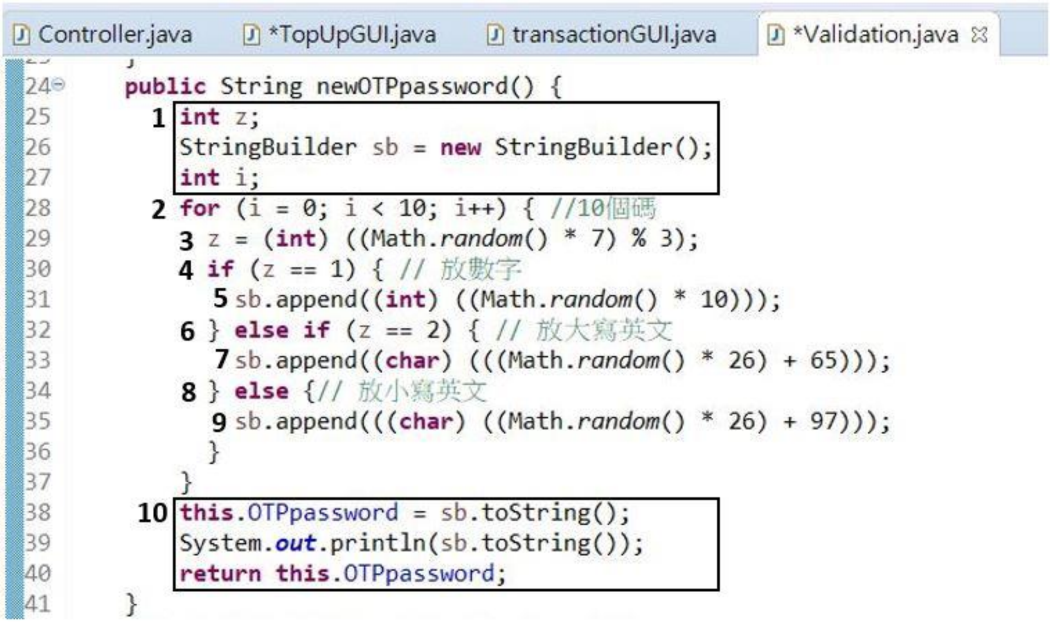
Path6: B→1→2→3→5→7→9→10→11→E

The Cyclomatic Complexity = 6

Number of edges – Number of nodes = 19 - 15 + 2= 6

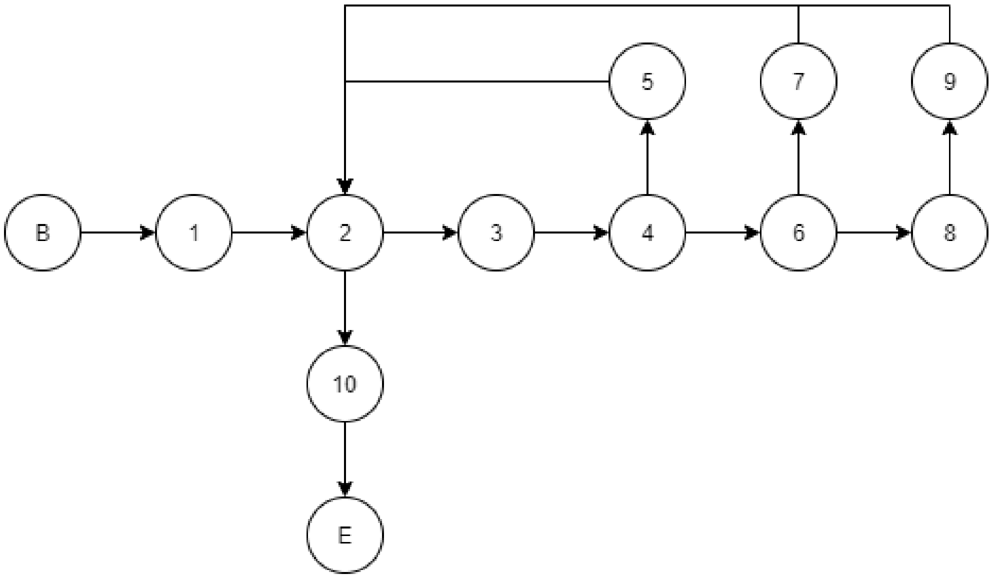
Number of closed region = 5 + 1 = 6

Number of atomic binary predicate = 5 + 1 = 6



```
    public String newOTPpassword() {
      1 int z;
        StringBuilder sb = new StringBuilder();
        int i;
      2 for (i = 0; i < 10; i++) { //10個碼
        3 z = (int) ((Math.random() * 7) % 3);
        4 if (z == 1) { // 放數字
           5 sb.append((int) ((Math.random() * 10)));
        6 } else if (z == 2) { // 放大寫英文
           7 sb.append((char) (((Math.random() * 26) + 65)));
        8 } else {// 放小寫英文
           9 sb.append(((char) ((Math.random() * 26) + 97)));
          }
        }
     10 this.OTPpassword = sb.toString();
        System.out.println(sb.toString());
        return this.OTPpassword;
    }
```

Derive Basis Path:



Path1: B→1→2→10→E

Path2: B→1→2→3→4→5→2→10→E

Path3: B→1→2→3→4→6→7→2→10→E

Path4: B→1→2→3→4→6→8→9→2→10→E

The Cyclomatic Complexity = 4
Number of edges – Number of nodes = 14 – 12 + 2 = 4
Number of closed region = 4 + 1 = 4
Number of atomic binary predicate = 4 + 1 = 4

Black box testing

Equivalence Partitioning: Transaction

1. Partition 1: (Receive account is self and Receive account is not self and Amount >0.0 and Amount <100000.0)
   Result: You can't transact with yourself.
2. Partition 2: (Receive account is self and Receive account is not self and Amount <0.0)
   Result: You can't transact with yourself.
3. Partition 3: (Receive account is self and Receive account is not self and Amount >100000.0)
   Result: You can't transact with yourself.
4. Partition 4: (Receive account is existed and Receive account is not self and Amount >0.0 and Amount <100000.0)
   Result: Starting OTP validation
5. Partition 5: (Receive account is existed and Receive account is not self and Amount <0.0)
   Result: You can't type this number.
6. Partition 6: (Receive account is existed and Receive account is not self and Amount >100000.0)
   Result: Amount most under 100000
7. Partition 7: (Receive account is not existed and Amount >0.0 and Amount <100000.0)
   Result: Account mismatch
8. Partition 8: (Receive account is not existed and Amount <0.0)
   Result: Account mismatch
9. Partition 9: (Receive account is not existed and Amount >100000.0)
   Result: Account mismatch

Boundary Value Analysis: Transaction

1. Partition 1: (Receive account is self and Receive account is not self and Amount >0.0 and Amount <100000.0)

{Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} * {Amount == 0, Amount ==0.0, Amount == -0.1, Amount == 0.1, Amount == 100000.0, Amount == 100000.1, Amount == 99999.9}

2. Partition 2: (Receive account is self and Receive account is not self and Amount <0.0)

   {Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} * {Amount == 0, Amount ==0.0, Amount == -0.1, Amount == 0.1}

3. Partition 3: (Receive account is self and Receive account is not self and Amount >100000.0)

   {Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} * {Amount == 100000.0, Amount == 100000.1, Amount == 99999.9}

4. Partition 4: (Receive account is existed and Receive account is not self and Amount >0.0 and Amount <100000.0)

   {Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} * {Amount == 0, Amount ==0.0, Amount == -0.1, Amount == 0.1, Amount == 100000.0, Amount == 100000.1, Amount == 99999.9}

5. Partition 5: (Receive account is existed and Receive account is not self and Amount <0.0)

   {Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} * {Amount == 0, Amount ==0.0, Amount == -0.1, Amount == 0.1}

6. Partition 6: (Receive account is existed and Receive account is not self and Amount >100000.0)

   {Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} * {Amount == 100000.0, Amount == 100000.1, Amount == 99999.9}

7. Partition 7: (Receive account is not existed and Amount >0.0 and Amount <100000.0)

   {Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} * {Amount == 0, Amount ==0.0, Amount == -0.1, Amount == 0.1, Amount == 100000.0, Amount == 100000.1, Amount == 99999.9}

8. Partition 8: (Receive account is not existed and Amount <0.0)

   {Receive account == correct account, Receive account == Empty string,

Receive account == null, Receive account == very long string} *
{Amount == 0, Amount ==0.0, Amount == -0.1, Amount == 0.1}

9. Partition 9: (Receive account is not existed and Amount >100000.0)
{Receive account == correct account, Receive account == Empty string, Receive account == null, Receive account == very long string} *
{Amount == 100000.0, Amount == 100000.1, Amount == 99999.9}

Cause-effect Analysis: Sing In

	1	2	3	4	5
Account	Null	Empty	Wrong account	Right account	Right account
password	-	-	-	Wrong password	Right password
Rule count	2	2	2	1	1
Result	Account incorrect	Account incorrect	Account incorrect	Password incorrect	Login success

5) Please analyze the invocation chains of your design.

Range	**Length**
addUser	
execute()→DriverManager.getConnection("jdbc:mysql://localhost:3306/teamwork?useSSL=false&serverTimezone=UTC","root","")→super.connection.createStatement()→st.executeUpdate(sql)	3
ConnectDB	
execute()→Class.forName("com.mysql.cj.jdbc.Driver")→DriverManager.getConnection("jdbc:mysql://localhost:3306/teamwork?useSSL=false&serverTimezone=UTC","root","")→super.connection.createStatement()→st.execute("SELECT * FROM teamwork")→st.getResultSet()→rs.next()→rs.getString("Account")→rs.getString("Password")	8
Controller	
addUser(String act,String psd,double d,int goodThru,int CVV,int Balance,String Email)→db.addUser(act, psd, d, goodThru, CVV,Balance,Email)→imp.addUser(act, psd, d, goodThru, CVV,Balance,Email)→Adduser.execute()→DriverManager.getConnection("jdbc:mysql://localhost:3306/teamwork?useSSL=false&serverTimezon	6

e=UTC","root","")→super.connection.createStatement()→st.executeUpdate(sql)	
getUser(String user) →db.getUser(user)→imp.getUser(name)→getuser.getResult()	3
getUserAccount()→user.getAct()	1
checkLogin(String act,String psd)→v.setUser(act, psd)→getUser(act)→db.getUser(user)→imp.getUser(name)→getuser.getResult()→db.getUser(act) →imp.getUser(name)→getuser.getResult()→v.validateUser(db.getUser(act))→user.getAct().equals(act) →user.getPsd().equals(psd)	11
getBalance()→user.getBalance()	1
PayC(String account,double amount)→db.getUser(account)→imp.getUser(name)→getuser.getResult()→user.getBalance()→a.getBalance()→user.getAct()→user.getPsd()→user.getCardNum()→user.getGoodThru()→user.getCVV()→p.execute()→db.saveUser()→db.saveUser()→imp.saveUser(act, psd, cardNum, goodThru, CVV,Balance) →Saveuser.execute()→a.getAct()→a.getPsd()→a.getCardNum()→a.getGoodThru()→a.getCVV()→p.receiverAmount()→db.saveUser()→user.getBalance()→user.setBalance()	24
1.TopUP()→w.setTopUpPlan(new PlanATopUp(u))→w.topup.TopUpExecute()→user.getAct()→user.getPsd()→user.getCardNum()→user.getGoodThru()→user.getCVV()→user.getBalance()→db.saveUser()→imp.saveUser(act, psd, cardNum, goodThru, CVV,Balance) →Saveuser.execute()	11
2. TopUP()→w.setTopUpPlan(new PlanATopUp(u))→w.topup.TopUpExecute()→user.getAct()→user.getPsd()→user.getCardNum()→user.getGoodThru()→user.getCVV()→user.getBalance()→db.saveUser()→imp.saveUser(act, psd, cardNum, goodThru, CVV,Balance) →Saveuser.execute()	11
3. TopUP()→w.setTopUpPlan(new PlanATopUp(u))→w.topup.TopUpExecute()→user.getAct()→user.getPsd()→user.getCardNum()→user.getGoodThru()→user.getCVV()→user.getBalance()→db.saveUser()→imp.saveUser(act, psd, cardNum, goodThru, CVV,Balance) →Saveuser.execute()	11
4. TopUP()→w.setTopUpPlan(new PlanATopUp(u))→w.topup.TopUpExecute()→user.getAct()→user.getPsd()→user.getCardNum()→user.getGoodThru()→user.getCVV()→user.getB	11

alance()→db.saveUser()→imp.saveUser(act, psd, cardNum, goodThru, CVV,Balance) →Saveuser.execute()	
checkEmail()→v.validateEmail(Email)→Email.matches(EMAIL_REGEX)	2
makeOTP()→v.newOTPpassword()→sb.toString()→sb.toString()→user.getAct()→user.getEmail()→otp.execute()→props.setProperty("mail.smtp.auth", "true")→props.setProperty("mail.transport.protocol", "smtp")→props.setProperty("mail.smtp.host", "smtp.gmail.com")→props.setProperty("mail.smtp.starttls.enable", "true")→props.setProperty("mail.debug", "true")→Session.getInstance(props,new Authenticator()→getPasswordAuthentication()→PasswordAuthentication("rt47207@gmail.com", "lzqjxilnvkeegmbv")→msg.setFrom(new InternetAddress("rt47207@gmail.com"))→msg.setSubject("Fun Pay Verification Code")→msg.setContent("<span style='font-size:30px;'>驗證碼："+OTP+"</span>", "text/html;charset=utf-8")→msg.setRecipients(RecipientType.TO, InternetAddress.parse(Email))→Transport.send(msg)	19
setTemporaryUser()→user.setAct(account)→user.setEmail(Email)	3
checkOTP()→v.validateOTP(OTP)→this.OTPpassword.matches(OTP)	3
checkUser()→db.getUser(user)→imp.getUser(name)→getuser.getResult()	3
DBcmd	
getResult()	0
DBimpl	
DBMgr	
saveUser()→imp.saveUser(act, psd, cardNum, goodThru, CVV,Balance)→Saveuser.execute()	2
getUser()→imp.getUser(name)→getuser.getResult()	2
addUser()→imp.addUser(act, psd, d, goodThru, CVV,Balance,Email) →Adduser.execute()→DriverManager.getConnection("jdbc:mysql://localhost:3306/teamwork?useSSL=false&serverTimezone=UTC","root","")→super.connection.createStatement()→st.executeUpdate(sql)	5
ConnectDB()→imp.ConnectDB()→CDB.execute()	2
DisconnectDB()→imp.DisconnectDB()→DCDB.execute()	2
DisconnectDB	
execute()→Class.forName("com.mysql.cj.jdbc.Driver")→DriverManager.getConnection("jdbc:mysql://localhost:3306/teamwork?useSSL=false&se	3

rverTimezone=UTC","root","")→super.connection.close()	
GetUser	
getResult()→this.execute()→user.setAct(account)→user.setPsd(psd)→user.setCardNum(cardNum)→user.setGoodThru(goodThru)→user.setCVV(CVV)→user.setBalance(balance)→user.setEmail(Email)	8
execute()→DriverManager.getConnection("jdbc:mysql://localhost:3306/teamwork?useSSL=false&serverTimezone=UTC","root","")→super.connection.createStatement()→st.execute(sql)→st.getResultSet()→rs.next()→rs.getString("Account")→rs.getString("Password")→rs.getDouble("CardNumber")→rs.getInt("GoodThru")→rs.getInt("CVV")→rs.getDouble("Balance")→rs.getString("Email")	12
GUI	
initialize()→window.getContentPane().setBackground(Color.PINK)→window.getContentPane().setForeground(Color.BLACK)→window.setTitle("\u4EA4\u6613\u7CFB\u7D71")→window.setBounds(100, 100, 450, 300)→window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)→window.getContentPane().setLayout(null)→lblAccount.setFont(new Font("Arial", Font.PLAIN, 15))→lblAccount.setHorizontalAlignment(SwingConstants.CENTER)→lblAccount.setBackground(Color.WHITE)→lblAccount.setBounds(101, 95, 57, 25)→window.getContentPane().add(lblAccount)→lblPassword.setFont(new Font("Arial", Font.PLAIN, 15))→lblPassword.setHorizontalAlignment(SwingConstants.CENTER)→lblPassword.setBackground(Color.WHITE)→lblPassword.setBounds(88, 147, 70, 25)→window.getContentPane().add(lblPassword)→lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER)→lblNewLabel.setFont(new Font("Arial Rounded MT Bold", Font.BOLD, 32))→lblNewLabel.setBounds(128, 24, 176, 41)→window.getContentPane().add(lblNewLabel)→textField.setBounds(159, 95, 176, 25)→textField.setBounds(159, 95, 176, 25)→textField.setColumns(10)→btnSignUp.setBackground(Color.GRAY)→btnSignUp.setFont(new Font("Arial", Font.PLAIN, 15))→btnSignUp.setBounds(236, 200, 99, 27)→window.getContentPane().add(btnSignUp)→textField_1.setColumns(10)→textField_1.setBounds(159, 147, 176,	52

25)→window.getContentPane().add(textField_1)→label.setFont(new Font("新細明體", Font.PLAIN, 17))→label.setForeground(Color.RED)→label.setBounds(336, 98, 57, 19)→window.getContentPane().add(label)→btnSignUp.addActionListener(new ActionListener()→actionPerformed(ActionEvent e)→windowSU.frame.setVisible(true)→window.setVisible(false)→btnSubmit.setBackground(Color.GRAY)→btnSubmit.setFont(new Font("Arial", Font.PLAIN, 15))→btnSubmit.setBounds(98, 200, 99, 27)→window.getContentPane().add(btnSubmit)→label_1.setForeground(Color.RED)→label_1.setFont(new Font("新細明體", Font.PLAIN, 17))→label_1.setBounds(336, 150, 57, 19)→window.getContentPane().add(label_1)→btnSubmit.addActionListener(new ActionListener()→actionPerformed(ActionEvent e)→label.setText(" ")→label_1.setText(" ")→c.getUser(textField.getText()).getAct()!=null→c.checkUser(textField.getText()).getAct()==null	
GUI2	
initialize()→frame.getContentPane().setBackground(Color.PINK)→frame.getContentPane().setLayout(null)→lblFunPay.setHorizontalAlignment(SwingConstants.CENTER)→lblFunPay.setFont(new Font("Arial Rounded MT Bold", Font.BOLD, 50))→lblFunPay.setBounds(152, 44, 288, 61)→frame.getContentPane().add(lblFunPay)→lblFunction.setHorizontalAlignment(SwingConstants.CENTER)→lblFunction.setFont(new Font("Arial Rounded MT Bold", Font.PLAIN, 15))→lblFunction.setBounds(234, 102, 136, 25)→frame.getContentPane().add(lblFunction)→btnTransaction.addActionListener(new ActionListener()→actionPerformed(ActionEvent arg0)→transactionGUI windowTr = new transactionGUI(c)→windowTr.frmTransaction.setVisible(true)→frame.dispose()→btnTransaction.setFont(new Font("Arial", Font.BOLD, 15))→btnTransaction.setBackground(Color.GRAY)→btnTransaction.setBounds(234, 216, 134, 34)→frame.getContentPane().add(btnTransaction)→btnTopUp.setFont(new Font("Arial", Font.BOLD, 15))→btnTopUp.setBackground(Color.GRAY)→btnTopUp.setBounds(234, 291, 134, 34)→frame.getContentPane().add(btnTopUp)→btnTopUp.addActionListe	38

ner(new ActionListener()→actionPerformed(ActionEvent arg0)→TopUpGUI gui = new TopUpGUI(c)→gui.frame.setVisible(true)→frame.dispose()→btnSignOut.setFont(new Font("Arial", Font.BOLD, 15))→btnSignOut.setBackground(Color.GRAY)→btnSignOut.setBounds(234, 362, 134, 34)→frame.getContentPane().add(btnSignOut)→frame.setBounds(100, 100, 609, 538)→frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)→btnSignOut.addActionListener(new ActionListener()→actionPerformed(ActionEvent arg0)→gui.window.setVisible(true)→frame.dispose()	
OTP	
execute()→props.setProperty("mail.smtp.auth", "true")→props.setProperty("mail.transport.protocol", "smtp")→props.setProperty("mail.smtp.host", "smtp.gmail.com")→props.setProperty("mail.smtp.starttls.enable", "true")→props.setProperty("mail.debug", "true")→Session.getInstance(props,new Authenticator()→getPasswordAuthentication()→msg.setFrom(new InternetAddress("rt47207@gmail.com"))→msg.setSubject("Fun Pay Verification Code")→msg.setContent("<span style='font-size:30px;'>驗證碼："+OTP+"</span>", "text/html;charset=utf-8")→msg.setRecipients(RecipientType.TO, InternetAddress.parse(Email))→Transport.send(msg)	12
Pay	
setTransfer()	0
execute()	0
receiverAmount()	0
PlanATopUp	
TopUpExecute()→super.setAccount(account)→account.getBalance()→super.account.setBalance(Balance)_	3
PlanBTopUp	
TopUpExecute()→super.setAccount(account)→account.getBalance()→super.account.setBalance(Balance)	3
PlanCTopUp	
TopUpExecute()→super.setAccount(account)→account.getBalance()→super.account.setBalance(Balance)	3

PlanDTopUp	
TopUpExecute() -> setAccount(account) -> getBalance(Balance) -> setBalance(Balance)	3
RDBimpl	
SaveUser(String act,String psd,double cardNum,int goodThru,int CVV,double Balance) -> excute()	2
addUser(String act,String psd,double cardNum,int goodThru,int CVV,double Balance,String Email) -> excute()	2
getUser(String name) -> getuser.getResult()	2
ConnectDB() -> CDB.excute()	2
DisconnectDB() -> DCDB.excute()	2
Receipt	
Receipt(Controller c, double amount) -> initialize()	2

initialize() -> getContentPane().setBackground(Color PINK) -> setBounds(100,100,458,361) -> setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) -> getContentPane().setLayOut(null)-> label.setHorizontalAlignment(SwingConstants.CENTER) -> label.setFont(new Font("Arial Rounded MT Bold",BOLD,50)) -> label.setBounds(151,73,136,25) -> frame.getContentPane().add(label) -> lblTransferSuccess.setHorizontalAlignment(SwingConstants.CENTER) -> lblTransferSuccess.setFont(new Font("Arial Rounded MT Bold",Font.PLAIN,15")) -> lblTransferSuccess.setBounds(151,73,136,25) -> frame.getContentPane().add(lblTransferSuccess) ->lblYourAccount.setFont(new Font("Arial", Font.***PLAIN***, 20)) -> lblYourAccount.setHorizontalAlignment(SwingConstants.RIGHT) -> lblYourAccount.setBounds(61,143,143,19) -> frame.getContentPane().add(lblYourAccount) -> account.setFont(new Font("Arial",Font.PLAIN,20)) -> account.setBounds(203,111,158,19)->frame.getContentPane().add(account) -> lblYourBalance.setHorizontalAlignment(SwingConstants.RIGHT),lblYourBalance.setFont(new Font("Arial",Font.PLAIN,20)) -> lblyourBalance.setBounds(61,143,143,19) -> frame.getContentPane().add(lblYourBanlance) -> lblYourBalance.setHorizontalAlignment(SwingConstants.***RIGHT***) -> lblYourBalance.setFont(new Font("Arial", Font.***PLAIN***, 20))-> lblYourBalance.setBounds(61, 143, 143, 19)->frame.getContentPane().add(lblYourBalance) -> balance.setFont(new Font("Arial", Font.***PLAIN***, 20) -> balance.setBounds(203, 143, 158, 19) -> frame.getContentPane().add(balance) ->lblTransferAccount.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblTransferAccount.setFont(new Font("Arial", Font.***PLAIN***, 20))->lblTransferAccount.setBounds(42, 175, 162, 19) -> frame.getContentPane().add(lblTransferAccount)->Tamount.setFont(new Font("Arial", Font.***PLAIN***, 20)) -> Tamount.setBounds(203, 175, 158, 19)-> frame.getContentPane().add(Tamount) -> label_6.setBounds(175, 228, 158, 19)->frame.getContentPane().add(label_6)-> btnClose.setBackground(Color.***GRAY***)-> btnClose.setFont(new Font("Arial", Font.***PLAIN***, 15)) ->btnClose.setBounds(171, 274, 99, 27)-> frame.getContentPane().add(btnClose)->	40
actionPerformed(ActionEvent e) -> gui2.fream.setVisible(true) -> frame.dispose()	3

SaveUser	
excute() **-**> st.excuteUpdate(sql)	2
SignUPGUI	

initialize() -> frame.getContentPane().setBackground(Color.***PINK***) -> frame.getContentPane().setLayout(null) -> label.setHorizontalAlignment(SwingConstants.***CENTER***) ->label.setFont(new Font("Arial Rounded MT Bold", Font.***BOLD***, 50)) -> label.setBounds(149, 0, 288, 61) -> frame.getContentPane().add(label) -> lblSignup.setHorizontalAlignment(SwingConstants.***CENTER***) -> lblSignup.setFont(new Font("Arial Rounded MT Bold", Font.***PLAIN***, 15)) -> lblSignup.setBounds(227, 60, 136, 25)-> frame.getContentPane().add(lblSignup)->lblAccount.setFont(new Font("Arial", Font.***BOLD***, 22))->lblAccount.setHorizontalAlignment(SwingConstants.***RIGHT***) -> lblAccount.setBounds(124, 87, 92, 36) ->frame.getContentPane().add(lblAccount) -> textField.setBounds(219, 96, 187, 25)-> frame.getContentPane().add(textField)->textField.setColumns(10)-> lblPassword.setHorizontalAlignment(SwingConstants.***RIGHT***)-> blPassword.setFont(new Font("Arial", Font.***BOLD***, 22))-> lblPassword.setBounds(97, 136, 119, 36)-> frame.getContentPane().add(lblPassword)-> textField_1.setColumns(10) -> textField_1.setBounds(219, 145, 187, 25)-> frame.getContentPane().add(textField_1)-> lblCardNumber.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblCardNumber.setFont(new Font("Arial", Font.***BOLD***, 22))-> lblCardNumber.setBounds(63, 234, 153, 36)-> frame.getContentPane().add(lblCardNumber)->textField_2.setColumns(10)-> textField_2.setBounds(219, 243, 187, 25)-> frame.getContentPane().add(textField_2)-> lblEmail.setHorizontalAlignment(SwingConstants.***RIGHT***) -> lblEmail.setFont(new Font("Arial", Font.***BOLD***, 22))-> lblEmail.setBounds(73, 379, 143, 36)-> frame.getContentPane().add(lblEmail)-> textField_3.setColumns(10) -> textField_3.setBounds(219, 388, 187, 25)-> frame.getContentPane().add(textField_3)-> lblCheckPassword.setHorizontalAlignment(SwingConstants.***RIGHT***)-> lblCheckPassword.setFont(new Font("Arial", Font.***BOLD***, 22))-> lblCheckPassword.setBounds(14, 185, 202, 36)-> frame.getContentPane().add(lblCheckPassword)-> textField_4.setColumns(10)-> textField_4.setBounds(219, 194, 187, 25)->	123

frame.getContentPane().add(textField_4)->btnCancel.setFont(new Font("Arial", Font.***PLAIN***, 15))-> btnCancel.setBackground(Color.***GRAY***)-> btnCancel.setBounds(338, 452, 99, 27)-> frame.getContentPane().add(btnCancel) -> lblGoodThru.setHorizontalAlignment(SwingConstants.***RIGHT***)-> lblGoodThru.setFont(new Font("Arial", Font.***BOLD***, 18))-> lblGoodThru.setBounds(14, 283, 202, 36)->…	

frame.getContentPane().add(lblGoodThru)->textField_5.setColumns(10)-> textField_5.setBounds(219, 292, 187, 25)-> frame.getContentPane().add(textField_5)-> lblCvv.setHorizontalAlignment(SwingConstants.***RIGHT***)-> lblCvv.setFont(new Font("Arial", Font.***BOLD***, 22))-> lblCvv.setBounds(73, 330, 143, 36)-> frame.getContentPane().add(lblCvv)->textField_6.setColumns(10)-> textField_6.setBounds(219, 339, 187, 25)-> frame.getContentPane().add(textField_6)-> AccountEL.setForeground(Color.***RED***)->AccountEL.setFont(new Font("Arial", Font.***PLAIN***, 15))-> AccountEL.setBounds(420, 99, 157, 19)-> frame.getContentPane().add(AccountEL)-> PasswordEL.setForeground(Color.***RED***)-> PasswordEL.setFont(new Font("Arial", Font.***PLAIN***, 15))-> PasswordEL.setBounds(420, 148, 157, 19)-> frame.getContentPane().add(PasswordEL)-> CheckPasswordEL.setForeground(Color.***RED***)-> CheckPasswordEL.setFont(new Font("Arial", Font.***PLAIN***, 15))-> CheckPasswordEL.setBounds(420, 197, 157, 19)-> frame.getContentPane().add(CheckPasswordEL)-> CardNumberEL.setForeground(Color.***RED***)-> CardNumberEL.setFont(new Font("Arial", Font.***PLAIN***, 15))-> CardNumberEL.setBounds(420, 246, 157, 19)-> frame.getContentPane().add(CardNumberEL)-> CVVEL.setForeground(Color.***RED***)-> CVVEL.setFont(new Font("Arial", Font.***PLAIN***, 15))-> CVVEL.setBounds(420, 342, 157, 19)-> frame.getContentPane().add(CVVEL)-> EmailEL.setForeground(Color.***RED***)-> EmailEL.setFont(new Font("Arial", Font.***PLAIN***, 15))-> EmailEL.setBounds(420, 391, 157, 19)-> frame.getContentPane().add(EmailEL)-> frame.setBounds(100, 100, 608, 539)-> frame.setDefaultCloseOperation(JFrame.***EXIT_ON_CLOSE***)-> actionPerformed(ActionEvent e)-> windowGUI.window.setVisible(true)-> frame.dispose()->btnSubmit.setFont(new Font("Arial", Font.***PLAIN***, 15))-> btnSubmit.setBackground(Color.***GRAY***)-> btnSubmit.setBounds(152, 452, 99, 27)-> frame.getContentPane().add(btnSubmit)-> label_1.setForeground(Color.***RED***)-> label_1.setFont(new Font("Arial", Font.***PLAIN***, 15))-> label_1.setBounds(420, 300, 143, 19)->	

frame.getContentPane().add(label_1)->actionPerformed(ActionEvent e)-> EmailEL.setText(“”)->AccountEL.setText(“”)->CheckPasswordEL.setText(“”)>CVVEL.setText(“”)->PasswordEL.setText(“”)->label_1.setText(“”)-> c.checkEmail(textField_3.getText())-> EmailEL.setText("Wrong Email!”)-> .getUser(textField.getText()).getAct()-> AccountEL.setText("Have same account!”)->…	

textField.getText().trim().isEmpty()-> AccountEL.setText("Don't be Null!”)-> textField_1.getText().trim().isEmpty()-> PasswordEL.setText("Don't be Null!”)-> !textField_1.getText().equals(textField_4.getText())-> CheckPasswordEL.setText("Doesn't match with password!”)-> textField_2.getText().length()!=12-> CardNumberEL.setText("Incorrect Card Number!”-> textField_6.getText().length()!=3-> CVVEL.setText("Incorrect CVV!”)-> textField_5.getText().length()!=4-> label_1.setText("Incorrect GoodThru!”)-> c.checkEmail(textField_3.getText())&&c.getUser(textField.getText()).getAct()==null&&textField_1.getText().equals(textField_4.getText())&&textField_2.getText().length()==12&&textField_6.getText().length()==3-> e1.printStackTrace()-> SUOC.frame.setVisible(true)->frame.dispose()	
SignUpOTPCheck	
c.setTemporaryUser(act,Email)-> c.makeOTP()-> initialize()	3

initialize()-> frame.getContentPane().setBackground(Color.***PINK***)-> frame.getContentPane().setLayout(null)-> label.setHorizontalAlignment(SwingConstants.***CENTER***)-> label.setFont(new Font("Arial Rounded MT Bold", Font.***BOLD***, 50))-> label.setBounds(69, 0, 288, 61)-> frame.getContentPane().add(label)-> lblVerification.setHorizontalAlignment(SwingConstants.***CENTER***)-> lblVerification.setFont(new Font("Arial Rounded MT Bold", Font.***PLAIN***, 15))-> lblVerification.setBounds(151, 58, 136, 25)-> frame.getContentPane().add(lblVerification)-> lblOtp.setHorizontalAlignment(SwingConstants.***RIGHT***)-> lblOtp.setFont(new Font("Arial", Font.***BOLD***, 22))-> lblOtp.setBounds(14, 126, 143, 36)-> frame.getContentPane().add(lblOtp)-> textField.setColumns(10)-> textField.setBounds(160, 135, 187, 25)-> frame.getContentPane().add(textField)-> lblPleaseCheckYour.setHorizontalAlignment(SwingConstants.***CENTER***)-> lblPleaseCheckYour.setFont(new Font("Arial", Font.***BOLD***, 22))-> lblPleaseCheckYour.setBounds(80, 85, 293, 36)-> frame.getContentPane().add(lblPleaseCheckYour)-> label_1.setForeground(Color.***RED***)-> label_1.setHorizontalAlignment(SwingConstants.***CENTER***)-> label_1.setFont(new Font("Arial", Font.***BOLD***, 22))-> label_1.setBounds(80, 164, 293, 36)-> frame.getContentPane().add(label_1)-> frame.setBounds(100, 100, 450, 300)-> frame.setDefaultCloseOperation(JFrame.***EXIT_ON_CLOSE***)-> btnCancel.setFont(new Font("Arial", Font.***PLAIN***, 15))-> btnCancel.setBackground(Color.***GRAY***)-> btnCancel.setBounds(258, 213, 99, 27)-> frame.getContentPane().add(btnCancel)-> actionPerformed(ActionEvent e)-> windowSUG.frame.setVisible(true)-> frame.dispose()-> button.setFont(new Font("Arial", Font.***PLAIN***, 15))-> button.setBackground(Color.***GRAY***)-> button.setBounds(69, 213, 99, 27)-> frame.getContentPane().add(button)-> button.addActionListener(new ActionListener()-> actionPerformed(ActionEvent e)-> !c.checkOTP(textField.getText())->label_1.setText("OTP mismatch.”-> c.addUser(act, psd,cardNum,goodThru,CVV,0,Email)-> windowGUI.window.setVisible(true)	46
testMain	
db.ConnectDB()-> EventQueue.*invokeLater*(new Runnable()	2
run()-> window.window.setVisible(true)-> e.printStackTrace()	3

TopUp	
setAccount()	1
TopUpGUI	

e.printStackTrace()-> frame.getContentPane().setBackground(Color.***PINK***)-> frame.setBounds(100, 100, 608, 536)-> frame.setDefaultCloseOperation(JFrame.***EXIT_ON_CLOSE***)-> frame.getContentPane().setLayout(null)-> label.setHorizontalAlignment(SwingConstants.***CENTER***)->label.setFont(new Font("Arial Rounded MT Bold", Font.***BOLD***, 50))->label.setBounds(152, 42, 288, 61)->frame.getContentPane().add(label)->lblTopup.setHorizontalAlignment(SwingConstants.***CENTER***)->lblTopup.setFont(new Font("Arial Rounded MT Bold", Font.***PLAIN***, 15))->lblTopup.setBounds(230, 102, 136, 25)->frame.getContentPane().add(lblTopup)-> lblCheckYourPassword.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblCheckYourPassword.setFont(new Font("Arial", Font.***PLAIN***, 22))->lblCheckYourPassword.setBounds(48, 273, 222, 19)->frame.getContentPane().add(lblCheckYourPassword)->textField_1.setBounds(284, 273, 186, 25)->frame.getContentPane().add(textField_1)->btnCancel.setBackground(Color.***GRAY***)-> btnCancel.setFont(new Font("Arial", Font.***PLAIN***, 15))-> btnCancel.setBounds(372, 426, 99, 27)-> frame.getContentPane().add(btnCancel)-> actionPerformed(ActionEvent e)->windowG.frame.setVisible(true)->frame.dispose()->comboBox.setFont(new Font("Arial", Font.***PLAIN***, 15))-> comboBox.setModel(new DefaultComboBoxModel(new String[] {"TopUp with 100","TopUp with 500","TopUp with 1000", "TopUp with 5000”}))-> comboBox.setBounds(284, 225, 186, 25)->frame.getContentPane().add(comboBox)->lblChooseTopUp.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblChooseTopUp.setFont(new Font("Arial", Font.***PLAIN***, 22))->lblChooseTopUp.setBounds(61, 228, 209, 19)->frame.getContentPane().add(lblChooseTopUp)->label_1.setForeground(Color.***RED***)-> label_1.setHorizontalAlignment(SwingConstants.***RIGHT***)->label_1.setFont(new Font("Arial", Font.***PLAIN***, 22))->label_1.setBounds(185, 329, 222, 19)->frame.getContentPane().add(label_1)->btnConfirm.setBackground(Color.***GRAY***)->btnConfirm.setFont(new Font("Arial", Font.***PLAIN***, 15))->btnConfirm.setBounds(152, 426, 99,	51

27)->frame.getContentPane().add(btnConfirm)->actionPerformed(ActionEvent e)->c.getUser(c.getUserAccount()).getPsd().equals(textField_1.getText())->c.TopUP(c.getUser(c.getUserAccount()),(String)comboBox.getSelectedItem())->(String)comboBox.getSelectedItem()->…	

(String)comboBox.getSelectedItem()->(String)comboBox.getSelectedItem()-> TURGUI.frame.setVisible(true)->frame.dispose()->label_1.setText("Password mismatch.”)	
TopUpReceiptGUI	
initialize()-> frame.getContentPane().setBackground(Color.***PINK***)->frame.setBounds(100, 100, 458, 361)->frame.setDefaultCloseOperation(JFrame.***EXIT_ON_CLOSE***)->frame.getContentPane().setLayout(null)->label.setHorizontalAlignment(SwingConstants.***CENTER***)->label.setFont(new Font("Arial Rounded MT Bold", Font.***BOLD***, 50))->label.setBounds(73, 13, 288, 61)-> frame.getContentPane().add(label)-> lblTransferSuccess.setHorizontalAlignment(SwingConstants.***CENTER***)->lblTransferSuccess.setFont(new Font("Arial Rounded MT Bold", Font.***PLAIN***, 15))->lblTransferSuccess.setBounds(151, 73, 136, 25)->frame.getContentPane().add(lblTransferSuccess)->lblYourAccount.setFont(new Font("Arial", Font.***PLAIN***, 20))->lblYourAccount.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblYourAccount.setBounds(61, 111, 143, 19)->frame.getContentPane().add(lblYourAccount)->account.setFont(new Font("Arial", Font.***PLAIN***, 20))->account.setBounds(203, 111, 158, 19)->frame.getContentPane().add(account)->lblYourBalance.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblYourBalance.setFont(new Font("Arial", Font.***PLAIN***, 20))->lblYourBalance.setBounds(61, 143, 143, 19)->frame.getContentPane().add(lblYourBalance)->balance.setFont(new Font("Arial", Font.***PLAIN***, 20))->balance.setBounds(203, 143, 158, 19)->frame.getContentPane().add(balance)-> lblTransferAccount.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblTransferAccount.setFont(new Font("Arial", Font.***PLAIN***, 20))->lblTransferAccount.setBounds(42, 175, 162, 19)->frame.getContentPane().add(lblTransferAccount)->Tamount.setFont(new Font("Arial", Font.***PLAIN***, 20))->Tamount.setBounds(203, 175, 158, 19)->frame.getContentPane().add(Tamount)->label_6.setBounds(175, 228, 158, 19)->frame.getContentPane().add(label_6)->btnClose.setBackground(Color.***GRAY***)->btnClose.setFont(new	43

Font("Arial", Font.***PLAIN***, 15))->btnClose.setBounds(171, 274, 99, 27)->frame.getContentPane().add(btnClose)->actionPerformed(ActionEvent e)->gui2.frame.setVisible(true)->frame.dispose()	
TransactionGUI	

initialize()-> frmTransaction.setTitle(“Transaction”)-> frmTransaction.getContentPane().setBackground(Color.***PINK***)->frmTransaction.setBounds(100, 100, 608, 536)->frmTransaction.setDefaultCloseOperation(JFrame.***EXIT_ON_CLOSE***)->frmTransaction.getContentPane().setLayout(null)->lblReveiveAccount.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblReveiveAccount.setFont(new Font("Arial", Font.***PLAIN***, 22))->lblReveiveAccount.setBounds(83, 180, 176, 19)->frmTransaction.getContentPane().add(lblReveiveAccount)->lblAmount.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblAmount.setFont(new Font("Arial", Font.***PLAIN***, 22))->lblAmount.setBounds(14, 233, 245, 19)->frmTransaction.getContentPane().add(lblAmount)->textField.setBounds(273, 180, 186, 25)->frmTransaction.getContentPane().add(textField)->textField.setColumns(10)->textField_1.setColumns(10)->textField_1.setBounds(273, 233, 186, 25)->frmTransaction.getContentPane().add(textField_1)->btnExit.setBackground(Color.***GRAY***)->btnExit.setFont(new Font("Arial", Font.***PLAIN***, 15))->btnExit.setBounds(339, 386, 99, 27)->frmTransaction.getContentPane().add(btnExit)->actionPerformed(ActionEvent e)->windowGUI.frame.setVisible(true)->frmTransaction.dispose()->lblFunPay.setHorizontalAlignment(SwingConstants.***CENTER***)->lblFunPay.setFont(new Font("Arial Rounded MT Bold", Font.***BOLD***, 50))->lblFunPay.setBounds(154, 44, 288, 61)->frmTransaction.getContentPane().add(lblFunPay)->lblTransaction.setFont(new Font("Arial Rounded MT Bold", Font.***PLAIN***, 15))->lblTransaction.setHorizontalAlignment(SwingConstants.***CENTER***)->lblNewLabel.setFont(new Font("Arial", Font.***PLAIN***, 15))->lblNewLabel.setBounds(67, 339, 453, 19)frmTransaction.getContentPane().add(lblNewLabel)->btnSubmit.setBackground(Color.***GRAY***)->btnSubmit.setForeground(Color.***BLACK***)->btnSubmit.setFont(new Font("Arial", Font.***PLAIN***, 15))->btnSubmit.setBounds(164, 386, 99, 27)->frmTransaction.getContentPane().add(btnSubmit)->actionPerformed(ActionEvent e)-	53

>c.checkUser(textField.getText()).getAct()-> lblNewLabel.setText("Account mismatch”)->c.getBalance()-Double.*parseDouble*(textField_1.getText())->setText("You don't have enough money.Need to TopUp first.”)->c.getUserAccount().equals(textField.getText())->lblNewLabel.setText("You can't transact with yourself.”)->Double.*valueOf*(textField_1.getText())->lblNewLabel.setText("You can't type this number.”)->Double.*valueOf*(textField_1.getText())->….	

lblNewLabel.setText("Amount most under 100000”)-> e1.printStackTrace()->SUOC.frame.setVisible(true)->frmTransaction.dispose()	
TransactionOTPCheck	
c.makeOTP()->otp.excute()->props.setProperty(“mail.smtp.auth", "true”)->props.setProperty("mail.transport.protocol", “smtp”)-> props.setProperty("mail.smtp.host", "smtp.gmail.com")->props.setProperty("mail.smtp.starttls.enable", “true")->props.setProperty("mail.debug", “true")->PasswordAuthentication getPasswordAuthentication()->new PasswordAuthentication("rt47207@gmail.com", "lzqjxilnvkeegmbv"); ->msg.setFrom(new InternetAddress("rt47207@gmail.com"))->msg.setSubject("Fun Pay Verification Code”)->msg.setContent("<span style='font-size:30px;'>???eX?G"+OTP+"</span>", “text/html;charset=utf-8")->msg.setRecipients(RecipientType.***TO***, InternetAddress.*parse*(Email))->Transport.*send*(msg)->initialize()	15

initializa()-> frame.getContentPane().setBackground(Color.***PINK***)->frame.getContentPane().setLayout(null)-> label.setHorizontalAlignment(SwingConstants.***CENTER***)-> label.setFont(new Font("Arial Rounded MT Bold", Font.***BOLD***, 50))->label.setBounds(69, 0, 288, 61)->frame.getContentPane().add(label)->lblVerification.setHorizontalAlignment(SwingConstants.***CENTER***)->lblVerification.setFont(new Font("Arial Rounded MT Bold", Font.***PLAIN***, 15))->lblVerification.setBounds(151, 58, 136, 25)->frame.getContentPane().add(lblVerification)->lblOtp.setHorizontalAlignment(SwingConstants.***RIGHT***)->lblOtp.setFont(new Font("Arial", Font.***BOLD***, 22))->lblOtp.setBounds(14, 126, 143, 36)->frame.getContentPane().add(lblOtp)->textField.setColumns(10)->textField.setBounds(160, 135, 187, 25)->frame.getContentPane().add(textField)->lblPleaseCheckYour.setHorizontalAlignment(SwingConstants.***CENTER***)->lblPleaseCheckYour.setFont(new Font("Arial", Font.***BOLD***, 22))->lblPleaseCheckYour.setBounds(80, 85, 293, 36)->frame.getContentPane().add(lblPleaseCheckYour)->label_1.setForeground(Color.***RED***)->label_1.setHorizontalAlignment(SwingConstants.***CENTER***)->label_1.setFont(new Font("Arial", Font.***BOLD***, 22))->label_1.setBounds(80, 164, 293, 36)->frame.getContentPane().add(label_1)->frame.setBounds(100, 100, 450, 300)->frame.setDefaultCloseOperation(JFrame.***EXIT_ON_CLOSE***)->btnCancel.setFont(new Font("Arial", Font.***PLAIN***, 15))->btnCancel.setBackground(Color.***GRAY***)->btnCancel.setBounds(258, 213, 99, 27)->frame.getContentPane().add(btnCancel)->actionPerformed(ActionEvent e)->windowTG.frmTransaction.setVisible(true)->frame.dispose()->button.setFont(new Font("Arial", Font.***PLAIN***, 15))->button.setBackground(Color.***GRAY***)->button.setBounds(69, 213, 99, 27)->frame.getContentPane().add(button)->actionPerformed(ActionEvent e)->!c.checkOTP(textField.getText())->label_1.setText("OTP mismatch.")->c.PayC(act, amount)->RGUI.frame.setVisible(true)->frame.dispose()	43
User	
setAct(String)	1

setPsd(String psd)	1
setBalance(double cardNum)	1
setCardNum(double cardNum)	1
setGoodThru(int CVV)	1
setEmail(String Email)	1
getAct()	1
getPsd()	1
getCardNum()	1
getGoodThru()	1
getCVV()	1
getBalance()	1
getEmail()	1
Validation	
validateUser(User user)-> user.getAct().equals(act)&&user.getPsd().equals(psd))-	2
vaildateCard(double cardNum,int godThrum,int CVV)	1
VaildateEmail(String Email)-> Email.matches(EMAIL_REGEX)	2
newOTPpassword()-> sb.append((int) ((Math.*random*() * 10)))-> sb.append((char) (((Math.*random*() * 26) + 65)))->sb.append(((char) ((Math.*random*() * 26) + 97)))-> this.OTPpassword=sb.toString()	5
ValidateOTP(String OTP)-> this.OTPpassWord.matches(OTP)	2
Wallet	
excute()	1
setTopUpPlan(TopUp T)	1

Invocation Chain Length	Number of Chains
1	19
2	16
3	13
5	2
6	1
8	2
11	5
12	2
15	1
19	1
24	1
38	1
40	1
43	2
46	1
51	1
52	1
123	1

6) Based on the metrics result, please identify places that need improvement, Selecting an improvement strategy.

In WMC metrics indicate that classes with large numbers of methods are likely to be more application specific, limiting the possibility of reuse. Our WMC most value within the range of 2 to 4. In this metrics, ours code is possible to reuse, but also few classes are too high like controller class, user class, validation class. So in tw2, we will try to reduce these classes specific method and increase reusing feature.

In Our NOC [0 ot 2] have 27 and [4 to 5] have 2, but the greater the NOC the greater likelihood to reusing features. High reuse of base class. Inheritance is a form of reuse.

So we need to add more children in that "0" classes, but not misuse.

According to Godin & Miceli article, CBO above 14 is hard to maintain.
The highest value is 11, which is the controller class, but not more than 14.

The larger the RFC the more difficult to test and debug due to more complex interaction relationships and more effort required to understand the methods and prepare test cases and test stubs.

Our RFC highest is Controller, it is 48. It means controller class is hard to test and debug.

The value of LCOM is high, it means that the relationships between the methods are paired by the same access. High LCOM value implies that the class should be split into one or more classes. So, if want to maintain a high cohesion, make the value of LCOM smaller.

LCOM<=1 indicates a cohesive class, which is the "good" class.
LCOM>=2 indicates a problem. The class should be split into so many smaller classes.
Our LCOM values are near 1, it means that cohesion is normal and tend to "good" class. But some classes tend to “bad” class, so we will think and improve in TW2.

We will focus on special values and classes that like we indicating, and make our code better to achieve six metrics.

Participation

Number	Name	Grade
B10523038	Edward	H
B10523024	Steven	H
B10523032	Xavier	H
B10523033	Wing	H
B10523021	Johnny	H
B10523037	Yee	H
B10523006	Peggy	H
B10523007	Bess	H
B10523005	Aliss	H
B10523056	Sandy	H