



SEHH2242 OBJECT ORIENTATED PROGRAMMING 101B-G04  
Assignment 3 group project

## Contents

Assumptions that made in the ATM.....	2
GUI Design .....	3
A complete listing of source codes with explanations of the key program statements : .....	23
Test cases.....	77
Appendix.....	116

# Assumptions that made in the ATM

Assuem the JAVA version is 14.

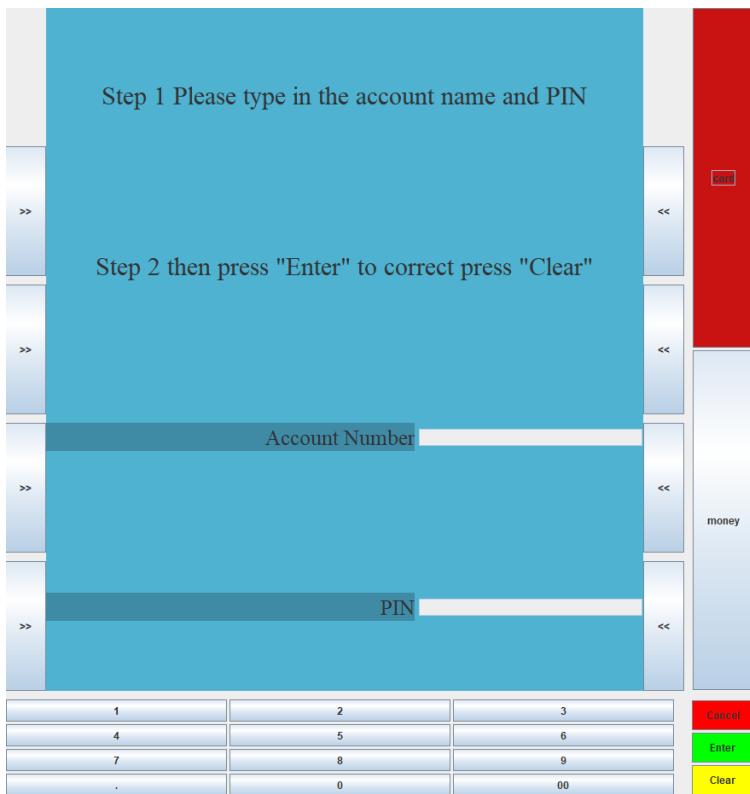


The “Enter” button shows in green color. User may use the “Enter” button to next step. The “Clear” button shows in yellow, which used to clear the user’ inputs. The “Cancel” button shows with red color. Once the user presses the “Cancel” button it will exit the ATM back to welcome. The side buttons “>>” and “<<” in silver color are implementing for choosing options. User choose, user need to click it when there are some options. The number pads in silver color are used for typing the amount. Those buttons on above are implemented to achieve the real ATM system.

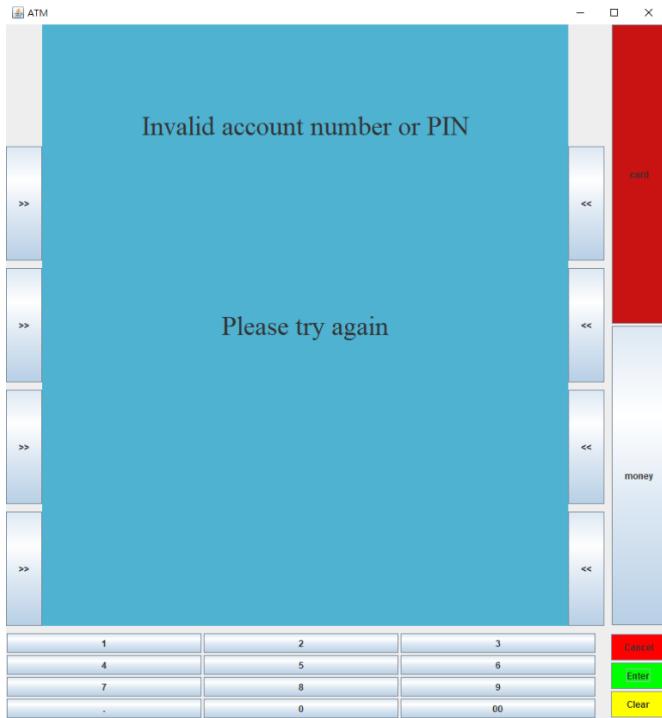
# GUI Design



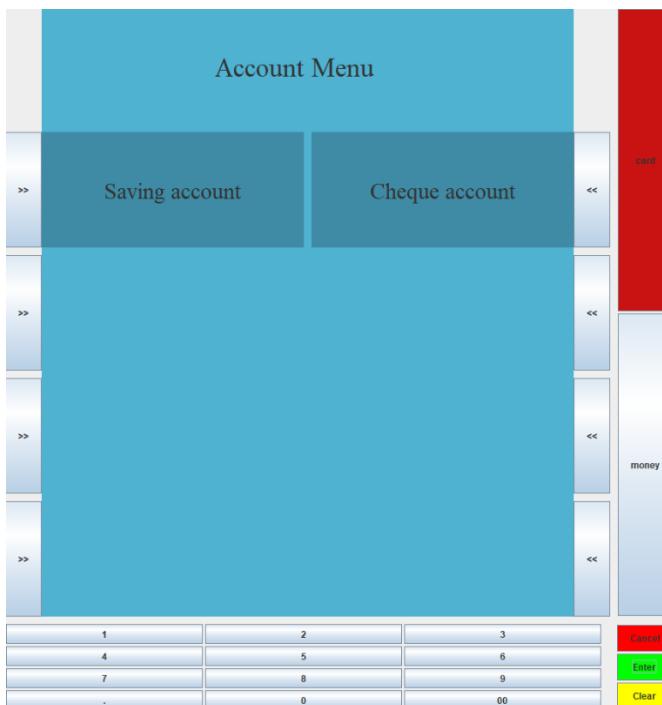
In the ATM greeting screen, user needs to click the “card” button on right ceiling place. This step is instead of putting ATM card. After putting the card into the ATM machine, the “card” button background will turn to red color.



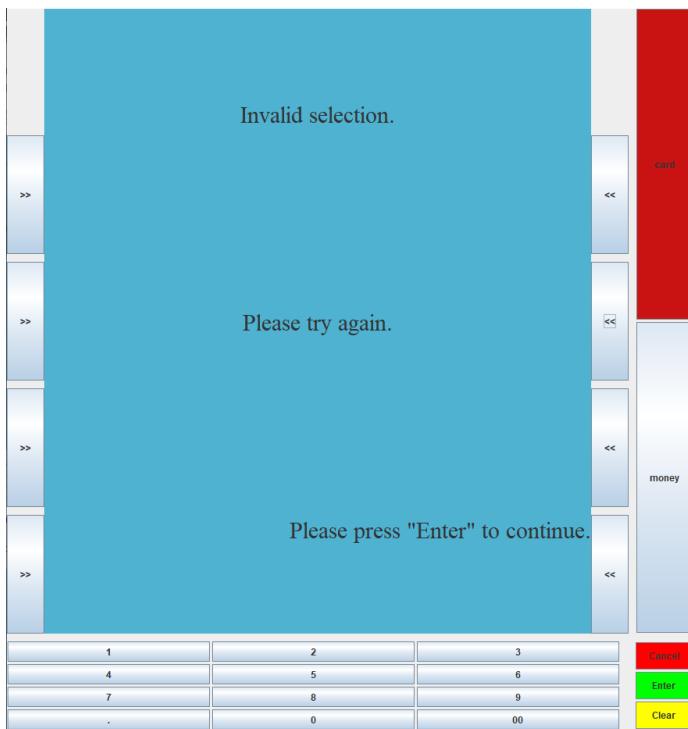
In the login screen, this screen shows two step notices for teaching user how to type in the account name and PIN number. When every time user inputs account number and PIN number by keypad, it also needs to press “Enter” button to go to next page. If there are typing wrong problem occurs, user can click “Clear” button and try again.



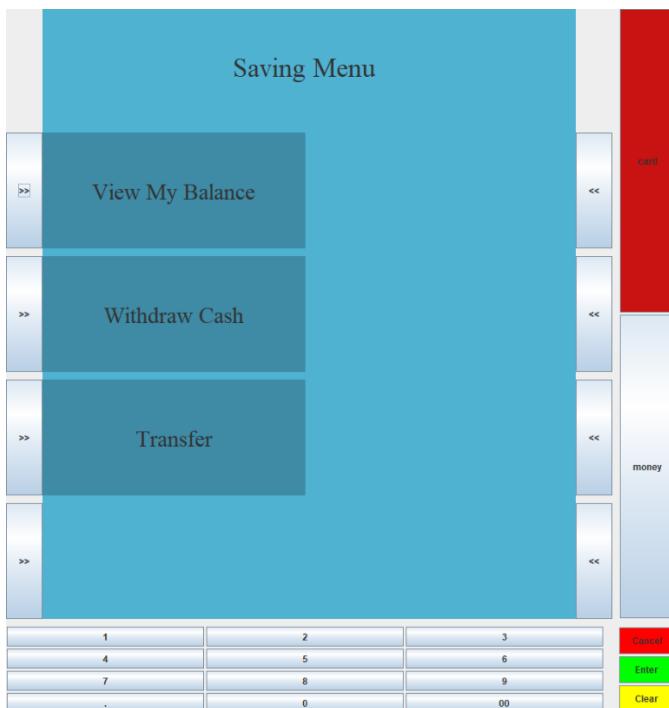
If user type in wrong account number or PIN, it will occur error message.



Before using any functions of the ATM machine, “Account Menu” have two options for user to choose the account of bank, including “Saving account” and “Cheque account”. Users need to click the button appropriately, which is “>>” or “<<” near to the options.

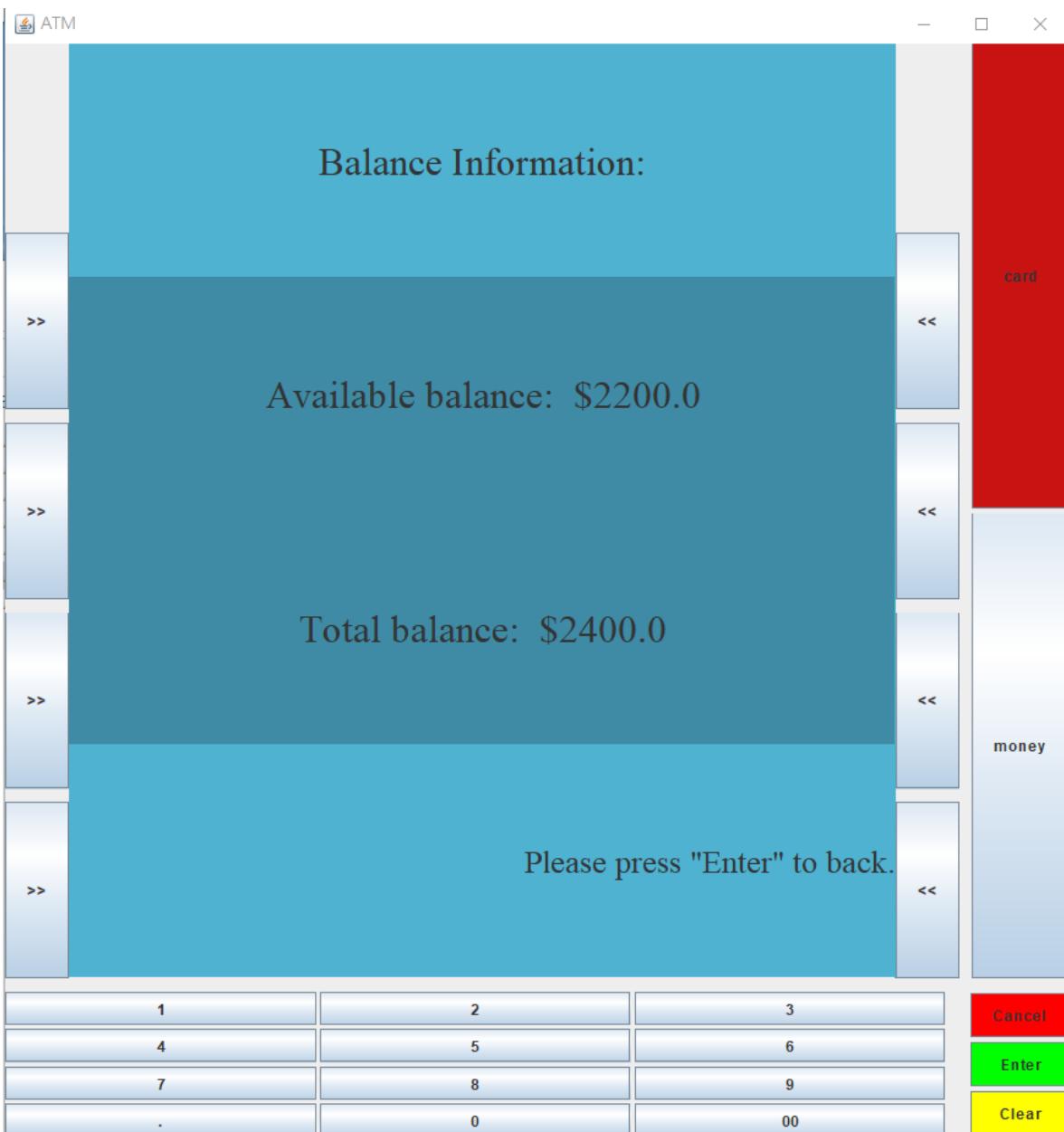


If they press the wrong “>>” or “<<” then will display Invalid selection message.



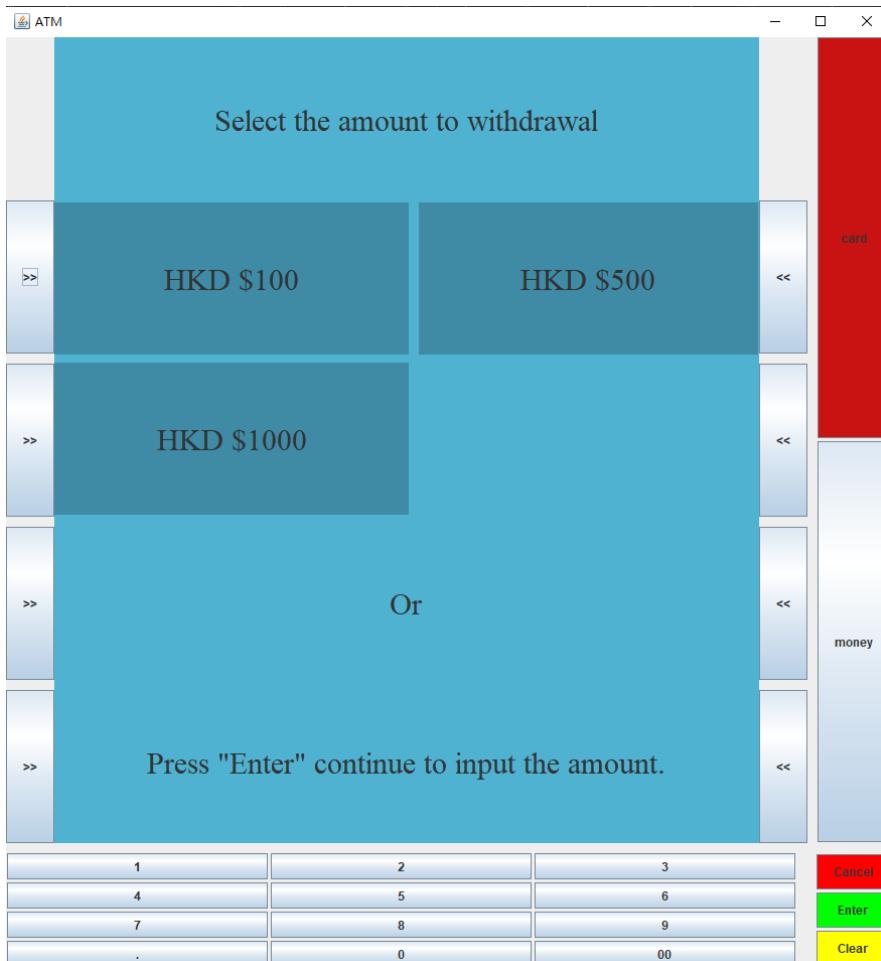
In the Saving Menu page, there are three choices, including “View my Balance”, “Withdrawal Cash” and “Transfer”. User can press the button “>>” near these to fulfill needs.

## View balance

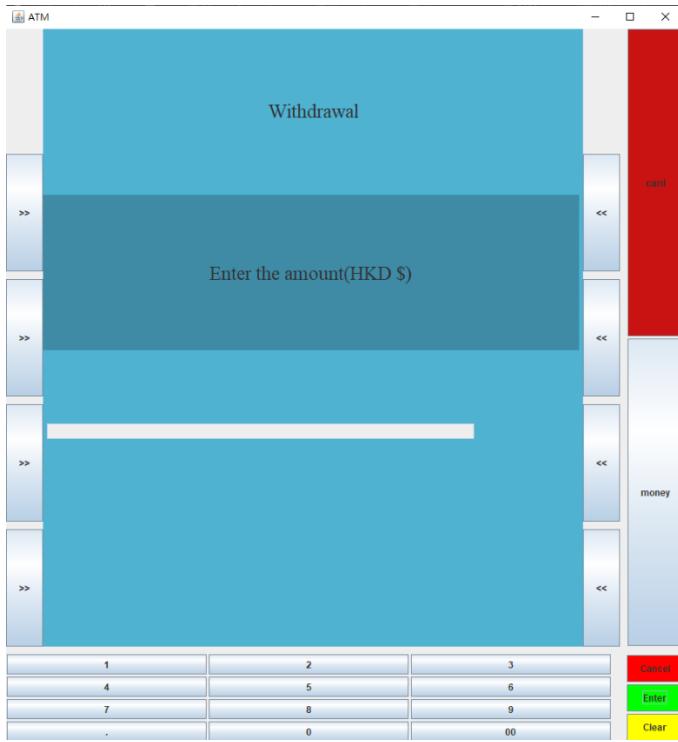


The function of “view my balance” can show the available balance and the total balance with the dollar sign and 1 decimal place. After viewing the balance information. User can press “Enter” to return saving menu.

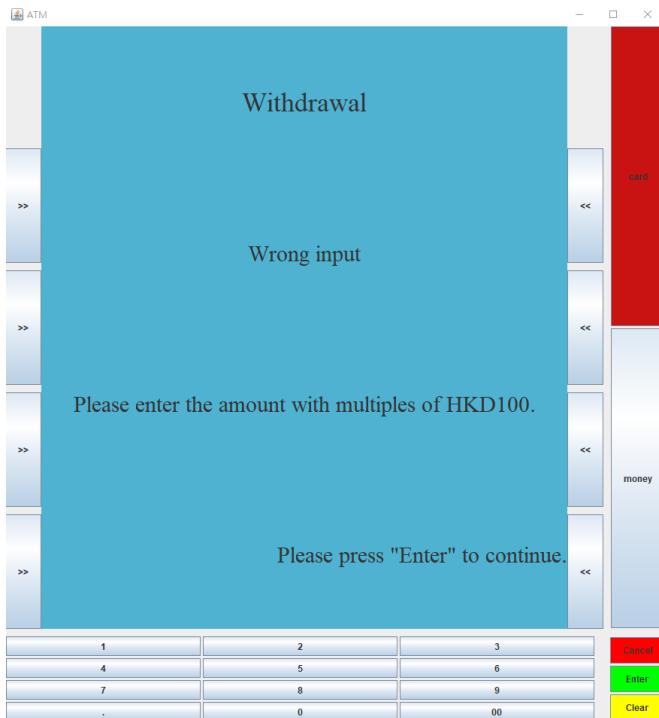
## Withdrawal



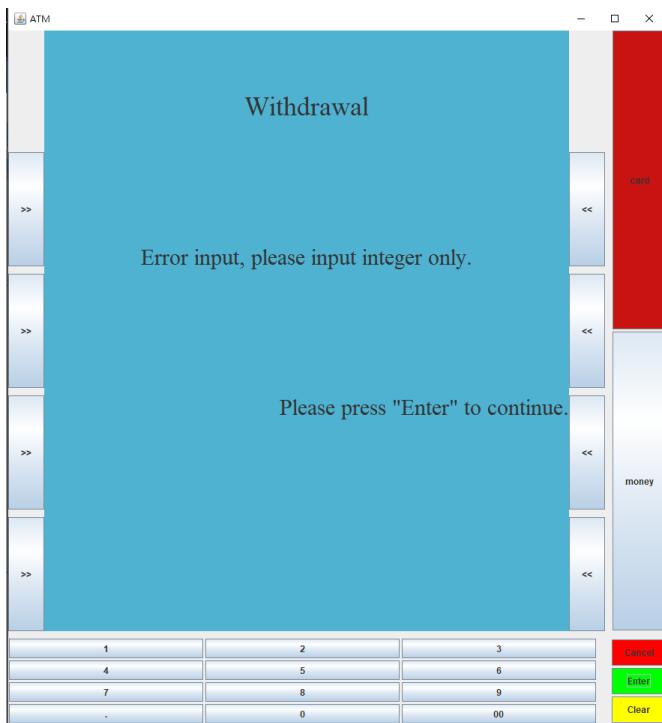
After pressing the button, it turns to the “Select Amount” page. User can choose “HKD \$100”, “HKD \$500” or “HKD \$1000” option by pressing the button. Then, it will go to next page. If user wants to input the specify amount, press “Enter” to go to type in page.



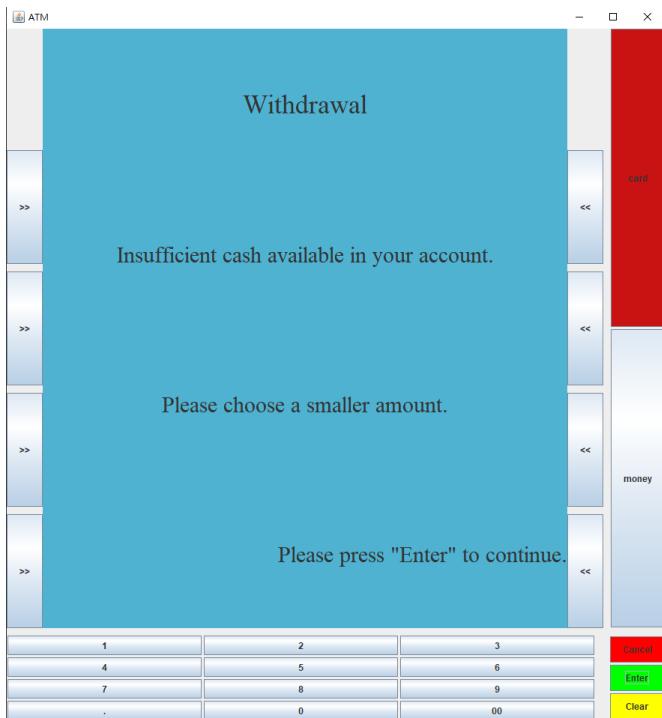
In this page, user needs to type in the amount in keypad and press “Enter”.



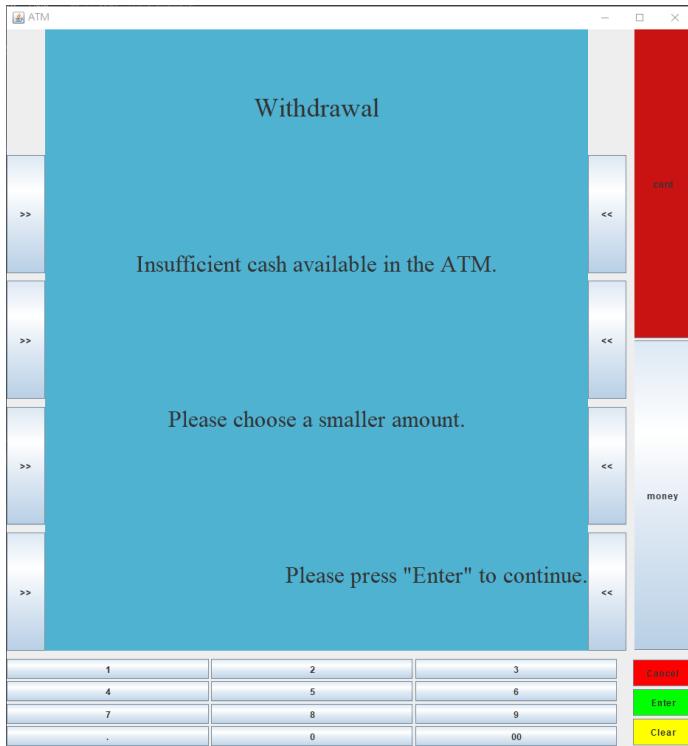
If user type in the amount not multiples of 100 like 50, it will occur these error messages.



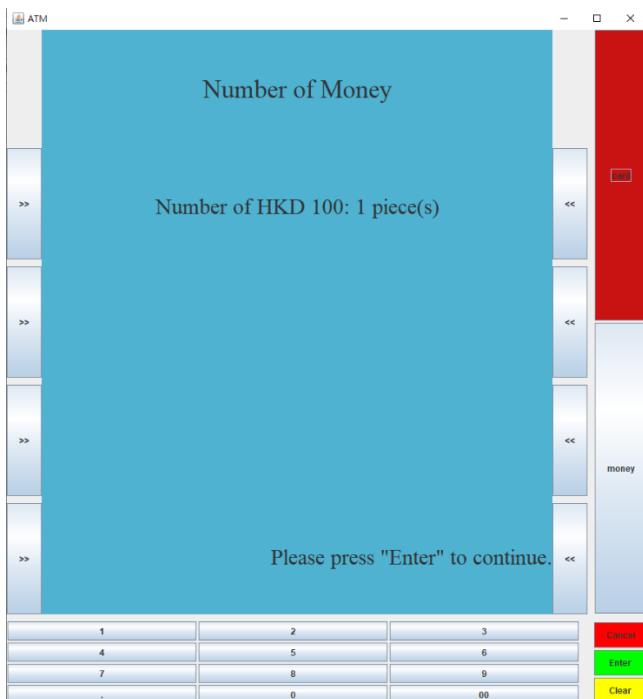
If user type in the amount not integer like 10.1, it will occur these error messages.



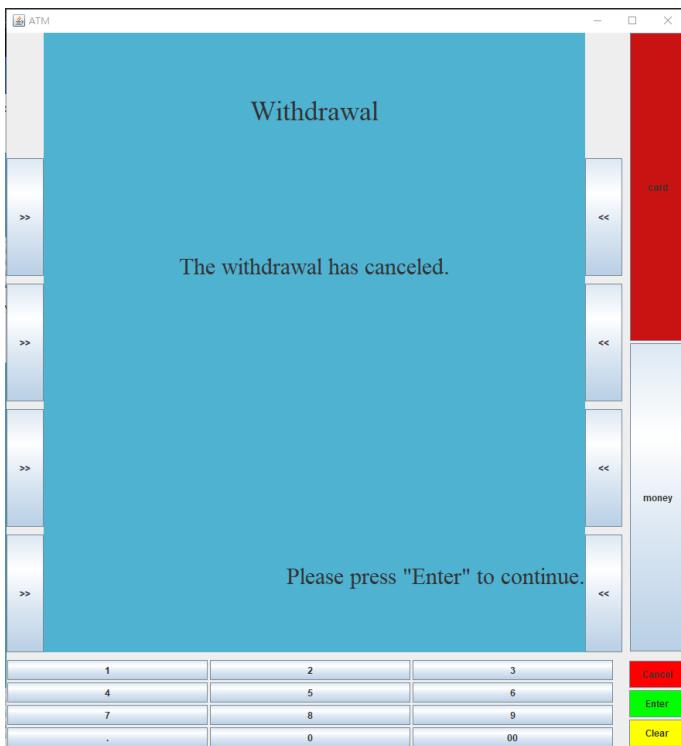
If user type in the amount is larger than he/she haves, it will occur these error messages.



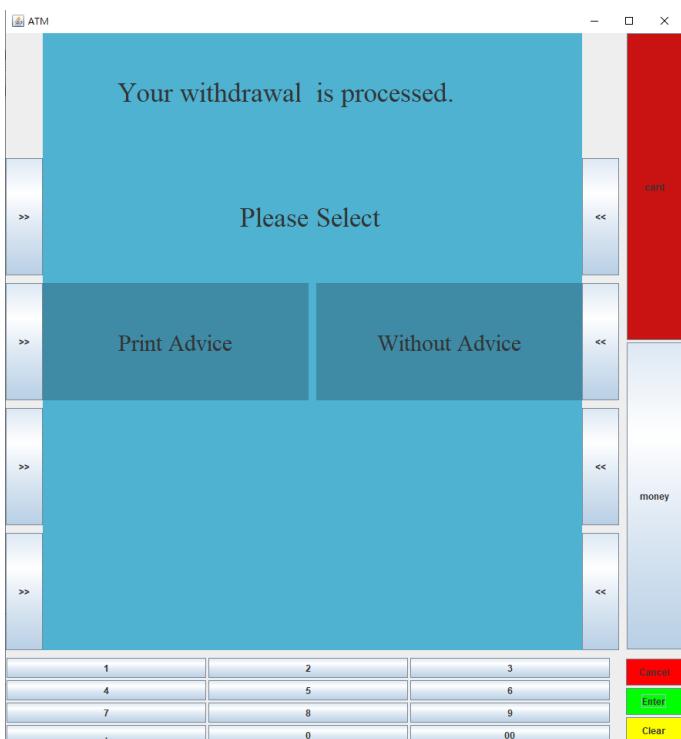
If user type in the amount is larger than the ATM haves, it will occur these error messages.



In this page, it shows the message to user for confirming the amount of money and number of currencies it will display the number of HKD when it has. Then, user need to press "Enter" to continue.



In this page, it shows the cancel message press enter to continue.



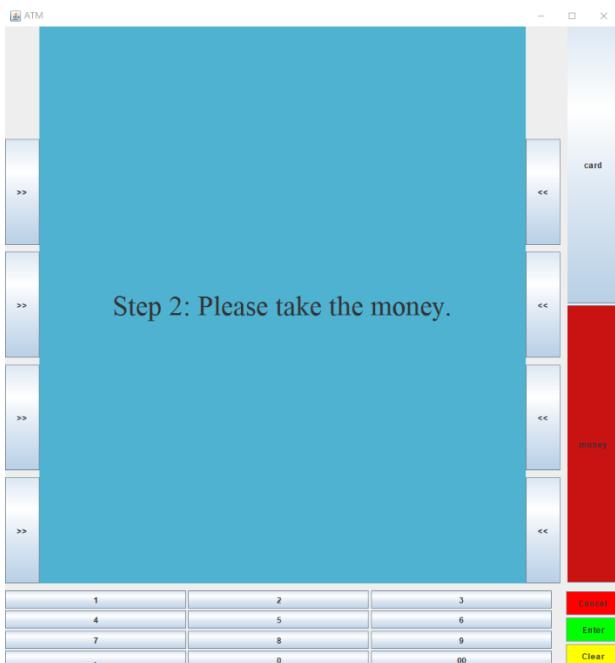
This page shows the withdrawal is processed, and let user choose print advice or not.

Finish Step1:



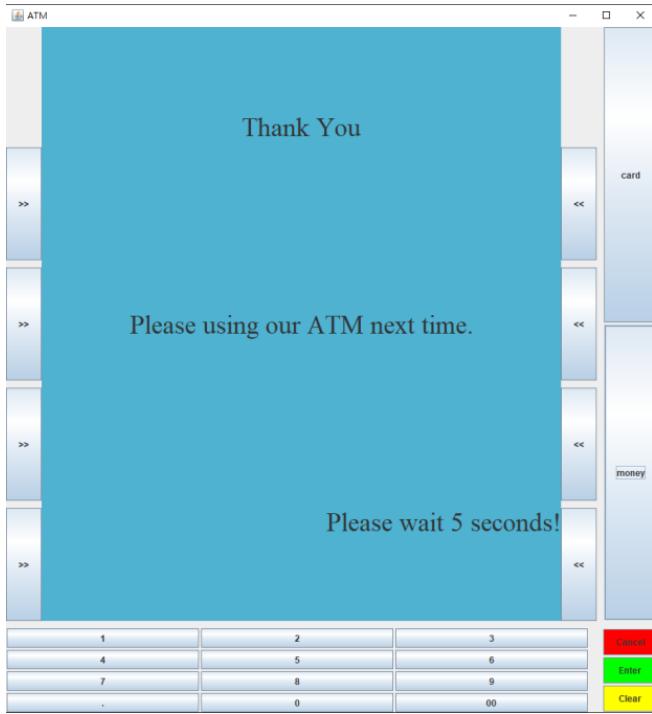
We will show a message of take your card first, user can take ATM card by pressing the money button with red color, to simulate the real atm system. After press the card button, the color will turn white, which simulate the real take card situation.

Step2:

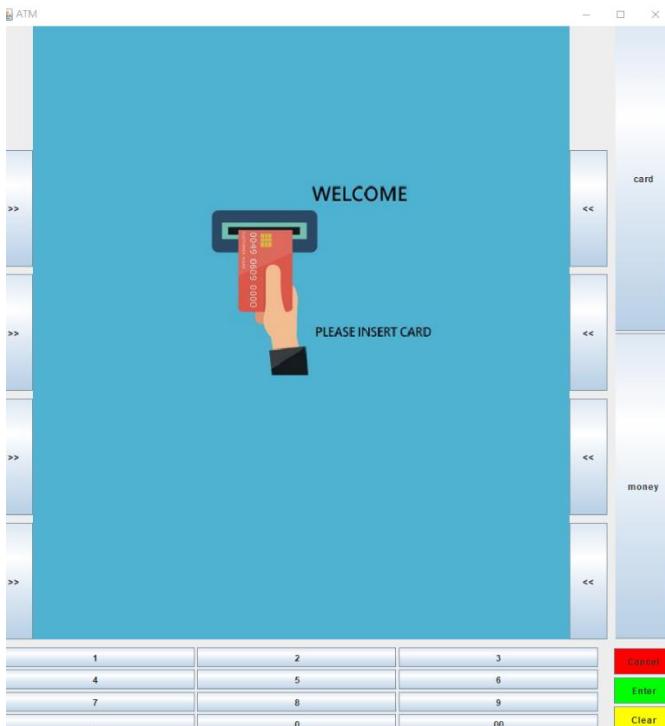


After that, user can take money by pressing the money button with color, to simulate the real atm system.

### Step3:



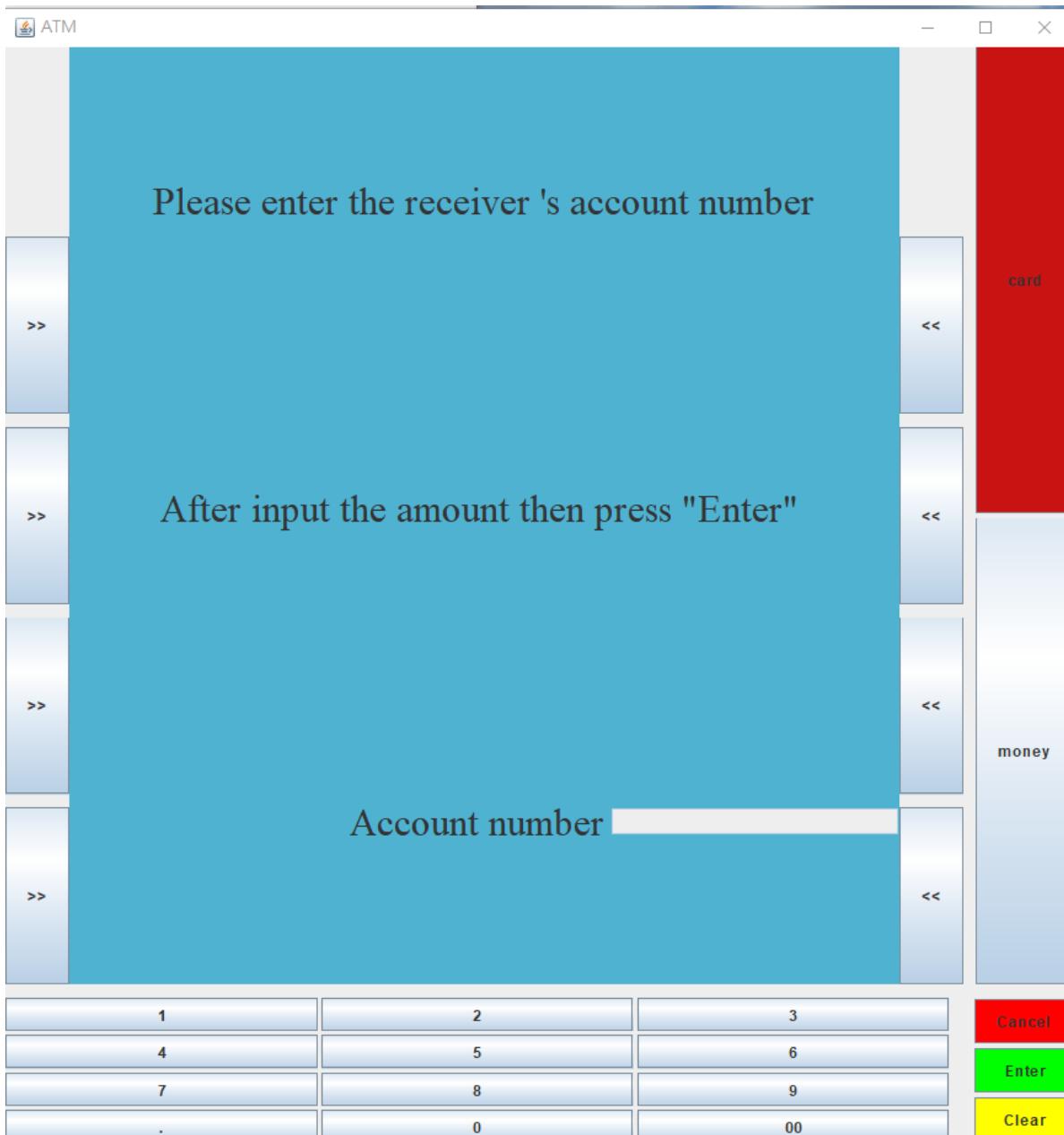
After take card and money, the red color button will turn white. This message will be show to user that the service is finish.



Lastly, the system will back to the very first page of the ATM system like the real ATM system.

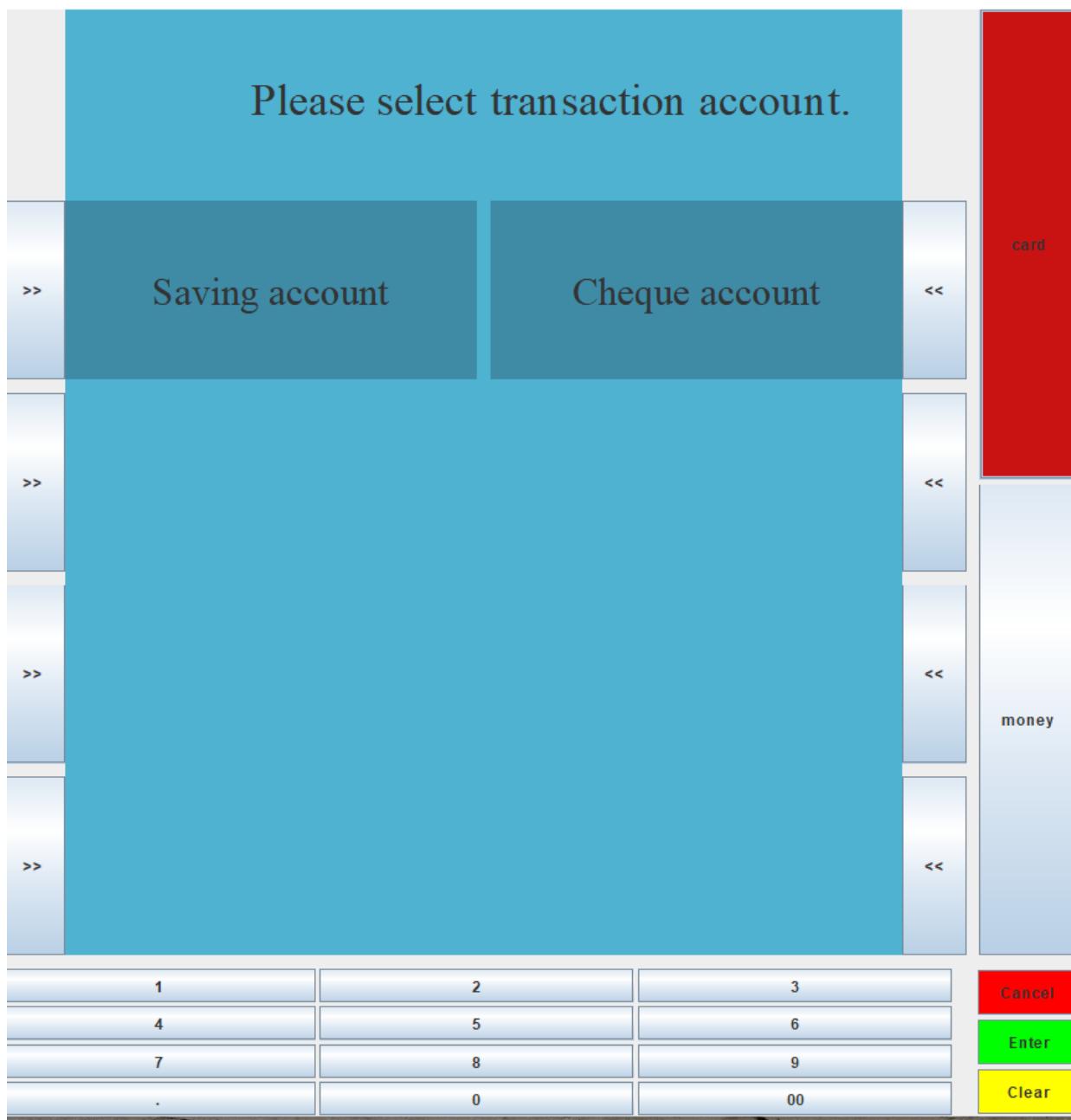
## Transfer

Step1:



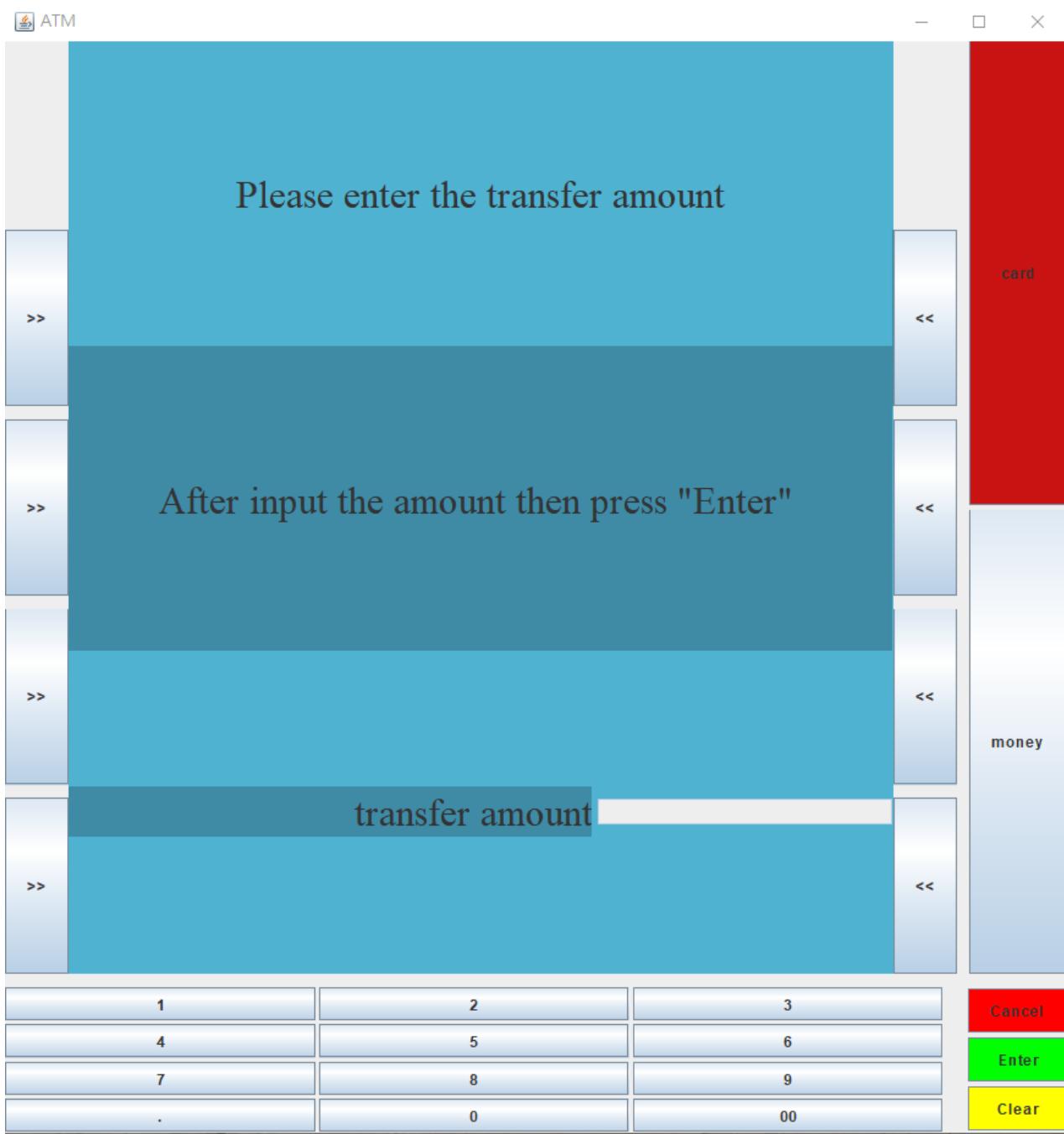
After getting in the transfer function, the screen required to input the Receiver's account number. And press “Enter” to get to next step.

Step2:



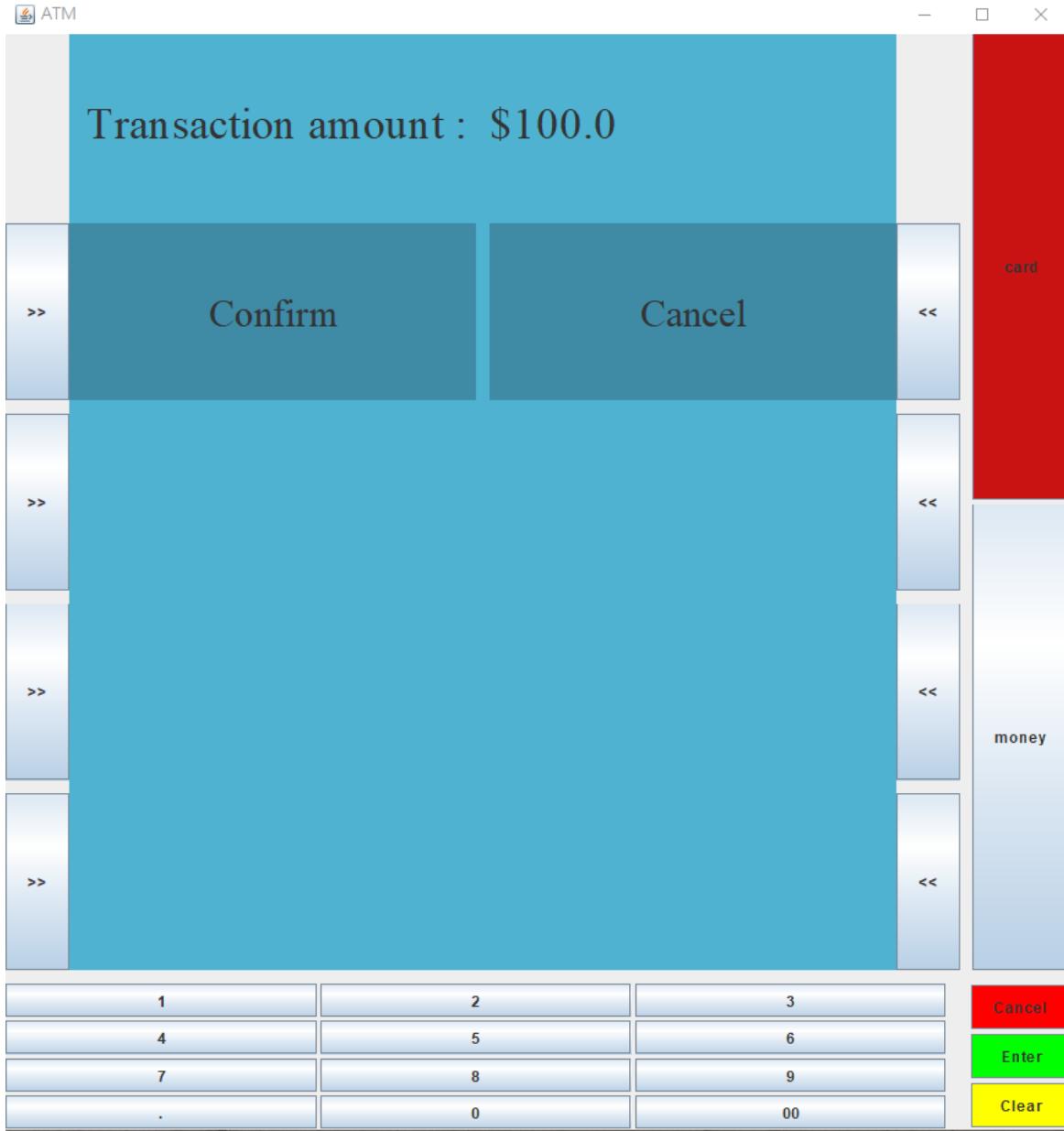
Then, choose the saving account or cheque account by pressing the button on side. Therefore, the system will go to next step.

Step3:



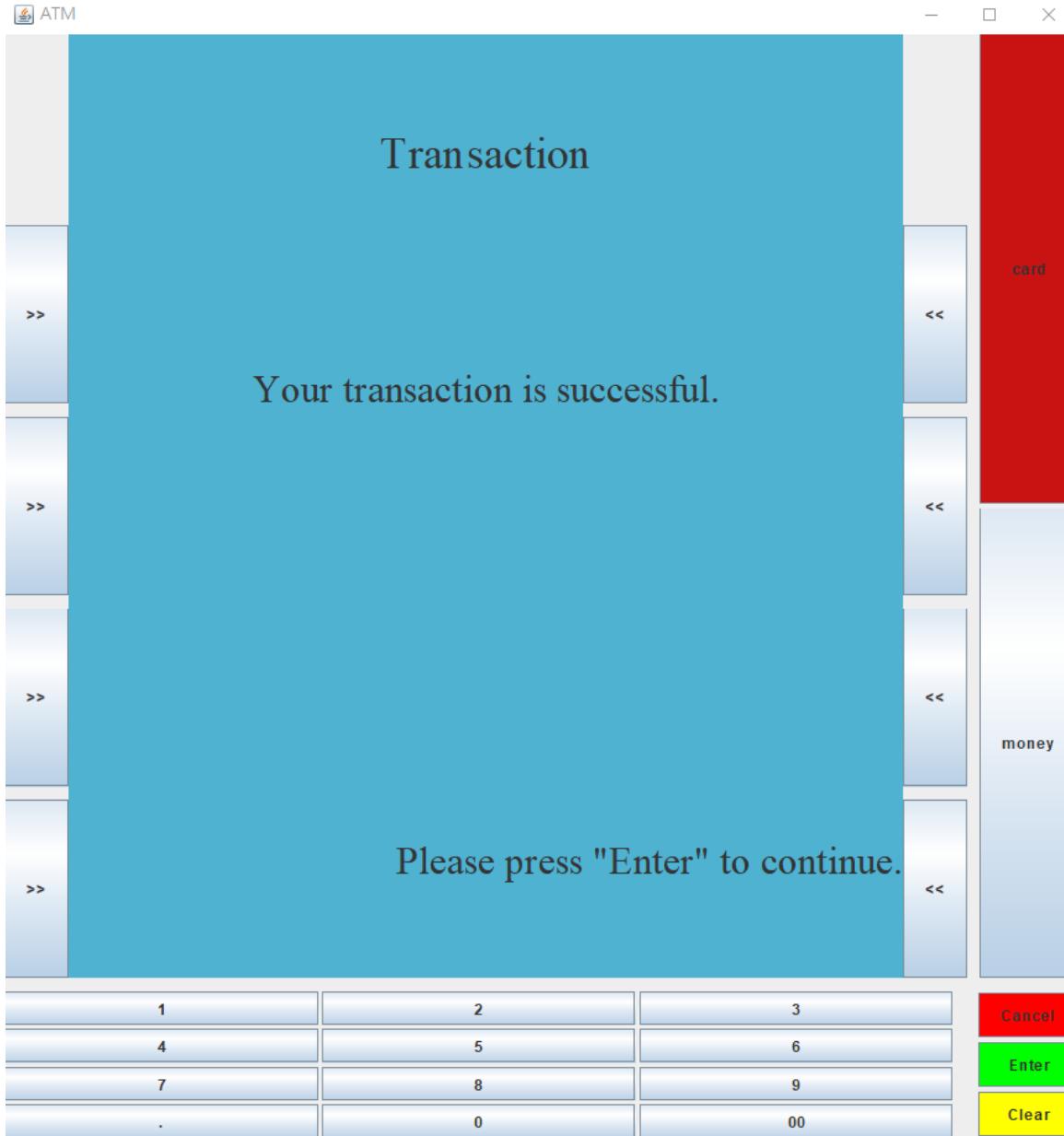
After that, input the amount you want to transfer. And press “Enter” to get to next step.

Step4:

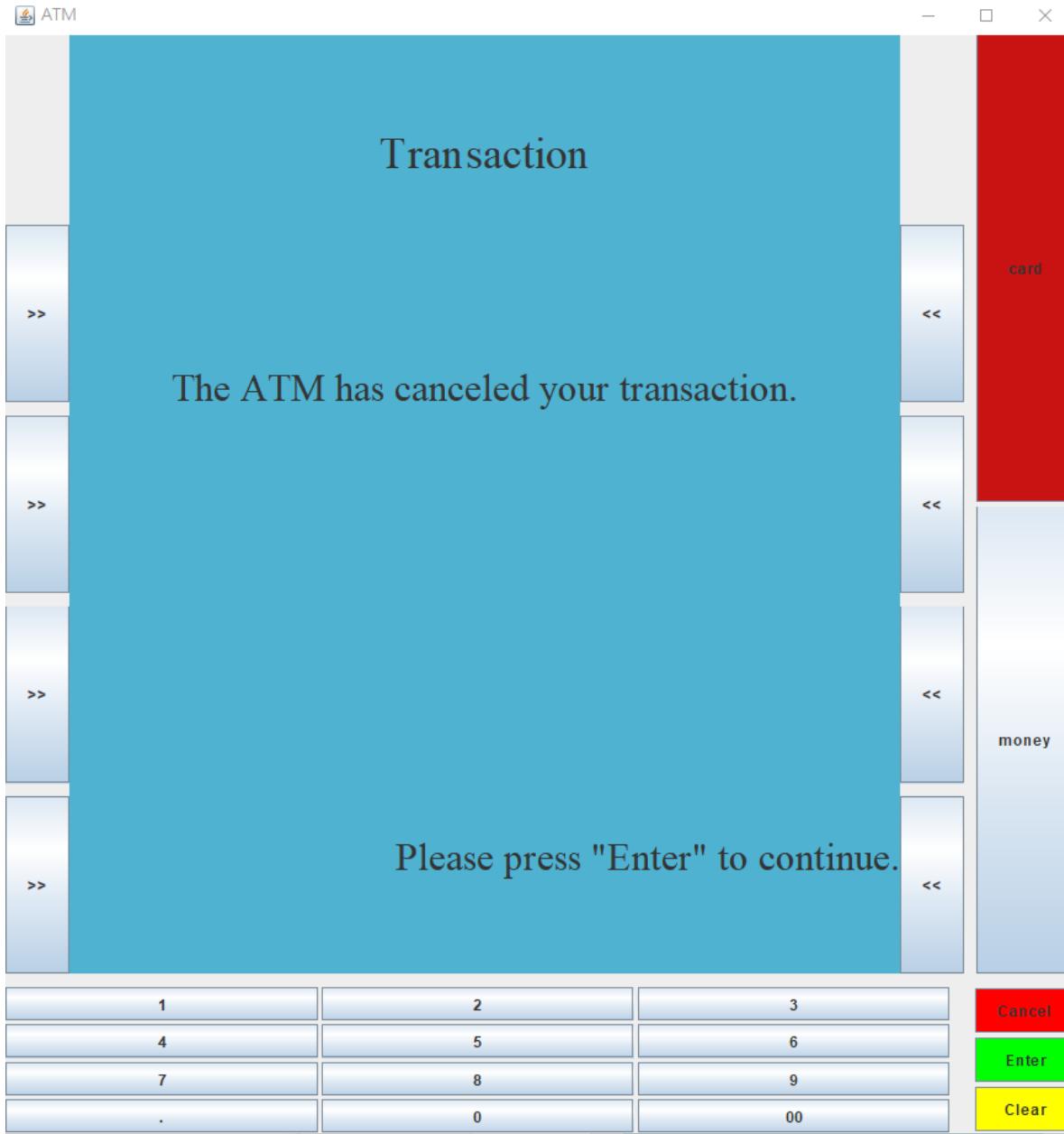


Lastly, the system will show a message of your amount input and asked for the confirm and cancel.

Step5:

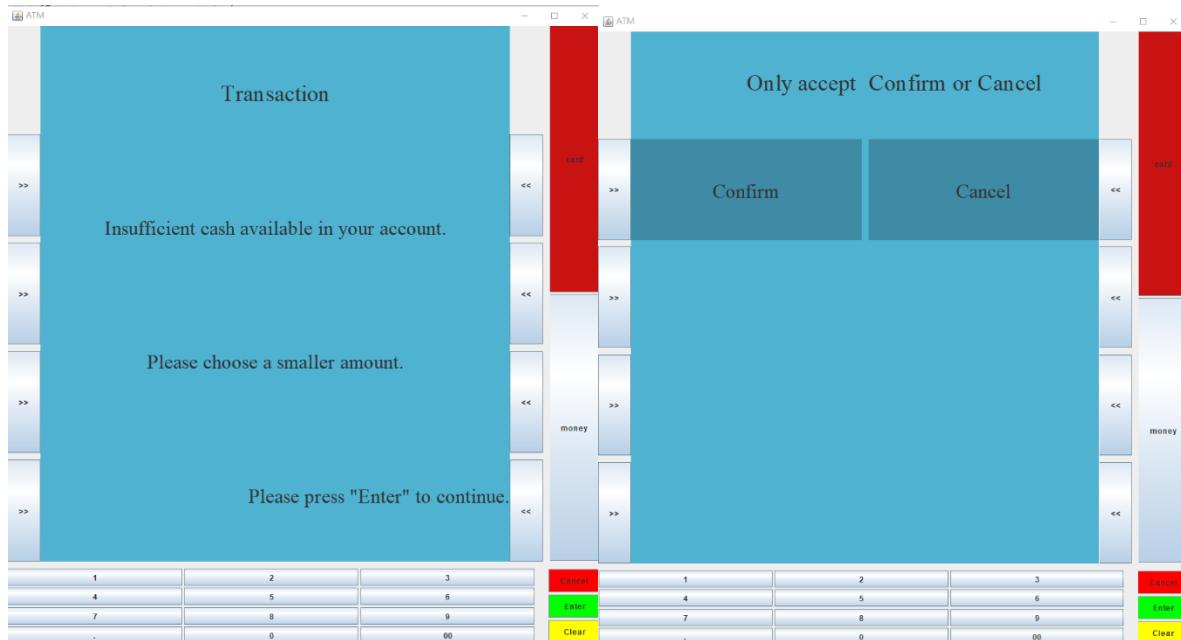


After the confirmation, the message of show your transaction is successful will be shown.



If user choose the cancel function, then the message will be shown after will exit the ATM.





Here are the invalid messages of GUI design.

A complete listing of source codes with explanations of the key program statements :

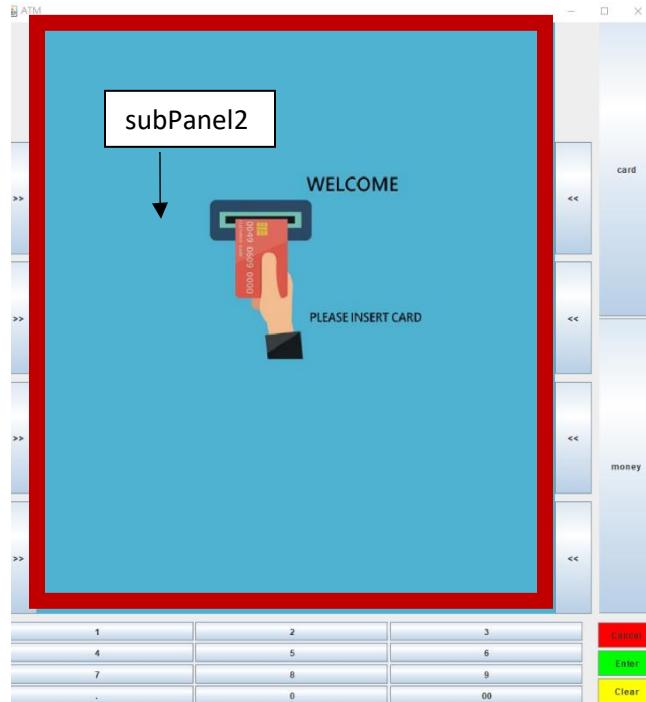
Overall common code

```
Color atmBlue = new Color( r: 79, g: 178, b: 209); //set the background color  
Color choiceBlue = new Color( r: 63, g: 139, b: 166); //set the choice colour
```

Here is the colour we set one is the background and one is the choice colour.

```
subPanel2.revalidate();  
subPanel2.repaint();
```

Here is to refresh it.



```
BorderLayout layout = (BorderLayout)subPanel2.getLayout();  
subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
```

It is to remove the previous layout in subPanel2 so that can add the new one.

From the ATM,

```
private Color atmBlue = new Color( r: 79, g: 178, b: 209);
private Color choiceBlue = new Color( r: 63, g: 139, b: 166);
private Color choiceRed = new Color( r: 201, g: 18, b: 18);

private JButton[] buttons; // array of buttons
private static final String[] names =
{ "1", "2", "3", "4", "5", "6", "7", "8", "9", ".", "0", "00" };

private JButton[] leftButtons; // array of buttons
private JPanel leftButtonJPanel; // panel to hold buttons

private JButton[] rightButtons; // array of buttons
private JPanel rightButtonJPanel; // panel to hold buttons

private JButton[] extraButtons; // array of buttons
private JPanel extraButtonJPanel; // panel to hold buttons

private JButton[] checkButtons; // array of buttons
private JPanel checkButtonJPanel; // panel to hold buttons

private JPanel buttonJPanel; // panel to hold buttons

private String line1 = "";
private String linePass = "";

private JPanel accountPanel;
private JPanel accountPanelBlank1 = new JPanel();
private JPanel accountPanelBlank2 = new JPanel();
private JPanel accountPanel1 = new JPanel();
private JPanel accountPanel2 = new JPanel();
private JPanel accountPanel3 = new JPanel();
private JPanel accountPanel4 = new JPanel();
private JPanel accountPanel5 = new JPanel();
private JPanel accountPanel6 = new JPanel();
private JPanel accountPanel7 = new JPanel();
private JPanel accountPanel8 = new JPanel();

private JPanel savingPanel;
private JPanel chequePanel;

private int stepCount = 0;
private int index = -1;
private int accountIndex = -1;

private int accountNumberInput = -1; // for enter the amount
private String pinInput; // for enter the amount
private ButtonHandler handler = new ButtonHandler();

private JLabel label = new JLabel();
private boolean chooseAccount = false;
private int withdrawalConfirm = 0;
private boolean takeCard = false;
private boolean takeMoney = false;
private boolean cancelPressed = false;

private JPanel subPanel = new JPanel();
private JPanel subPanel2 = new JPanel();
private JPanel subPanel3 = new JPanel();

// text box for the account number
private JTextField passTextField = new JTextField( columns: 50 );
// text box for the PIN
private JPasswordField passwordText = new JPasswordField();

private boolean checkCardStatus = false; // insert card or not
private boolean enterPassReturn = false; // waiting user enter account number
private boolean enterPINReturn = false; // waiting user enter PIN
private boolean invalidConfirm = false; // waiting user to confirm the wrong message
private boolean enterReturn = false; // waiting user to press enter
private boolean chooseSaving = false; // waiting user choose the options in saving account menu
private boolean chooseCheque = false; // waiting user choose the options in cheque account menu

private boolean userAuthenticated; // whether user is authenticated
private int currentAccountNumber; // current user's account number
private Screen screen; // ATM's screen
private Keypad keypad; // ATM's keypad
private CashDispenser cashDispenser; // ATM's cash dispenser
private BankDatabase bankDatabase; // account information database
private int x; // store is it saving or cheque account

// constants corresponding to main menu options
private static final int BALANCE_INQUIRY = 1;
private static final int WITHDRAWAL = 2;
private static final int TRANSFER = 3;
private static final int EXIT = 4;
```

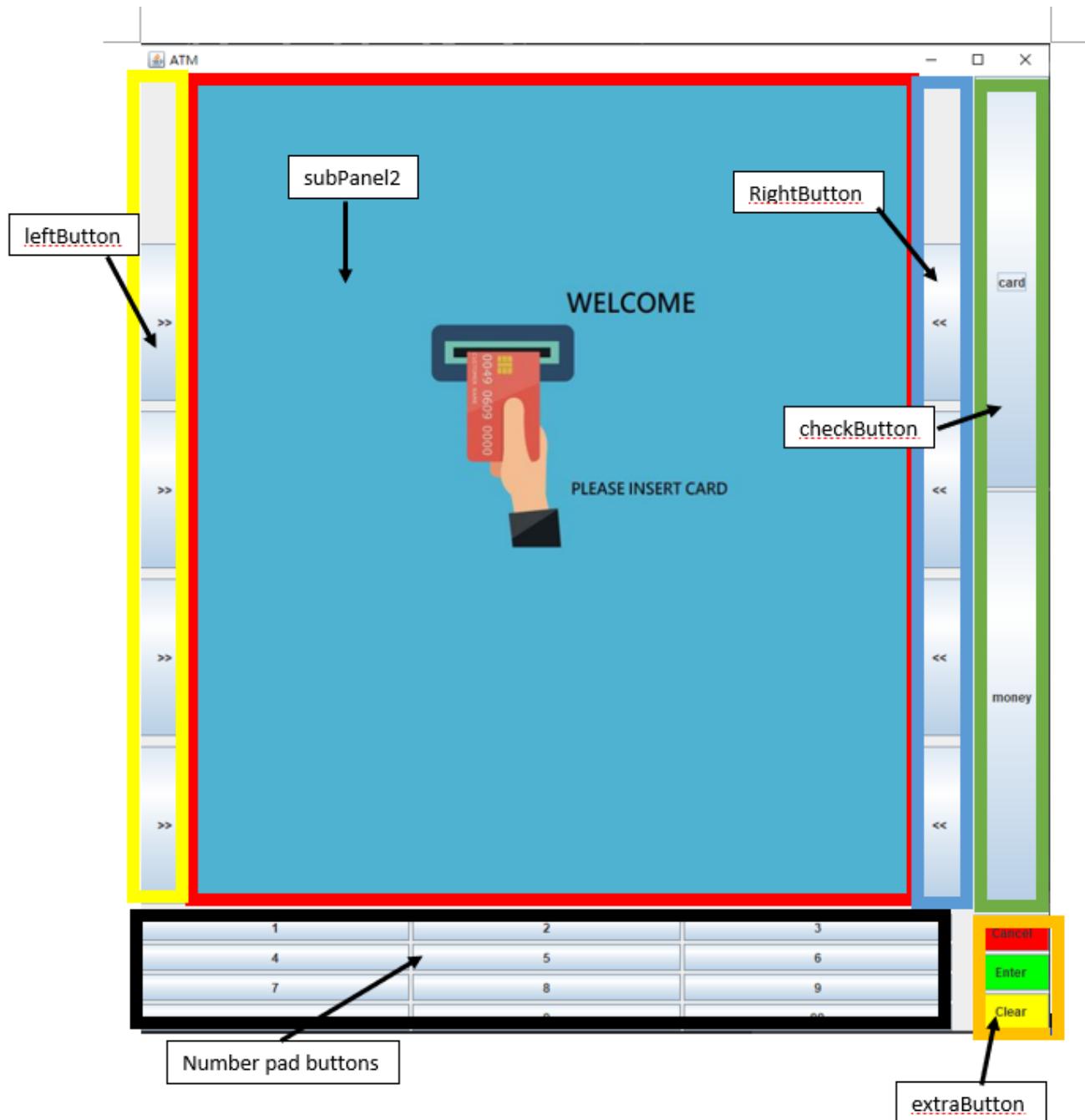
Initializing the variable.

```
JFrame frame = new JFrame();
frame.setTitle("ATM");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setBounds( x: 200, y: 150, width: 1000, height: 1000); // set size
frame.setLayout(new BorderLayout( hgap: 10, vgap: 10));
frame.setVisible(true);
```

Here is to set the frame title, size ,layout and set it visible.

```
accountPanelBlank1.setBackground(atmBlue);
accountPanelBlank2.setBackground(atmBlue);
accountPanel1.setBackground(atmBlue);
accountPanel2.setBackground(atmBlue);
accountPanel3.setBackground(atmBlue);
accountPanel4.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);
accountPanel6.setBackground(atmBlue);
accountPanel7.setBackground(atmBlue);
accountPanel8.setBackground(atmBlue);
```

Here is to set the panel colour to the colour we set.



```

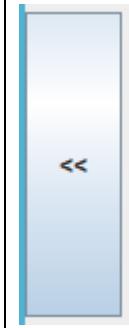
leftButtons = new JButton[4]; // create array of JButtons
leftButtonJPanel = new JPanel(); // set up panel
leftButtonJPanel.setLayout(new GridLayout( rows: 5, cols: 1, hgap: 0, vgap: 10)); //set GridLayout
JLabel blankLabel = new JLabel(); //set panel
leftButtonJPanel.add(blankLabel);
for (int count = 0; count < 4; count++) {
    leftButtons[count] = new JButton();
    leftButtonJPanel.add(leftButtons[count]); // add button to JFrame
} // end for
//set left buttons
leftButtons[0].setActionCommand("B0");
leftButtons[0].setText(">>");
leftButtons[1].setActionCommand("B1");
leftButtons[1].setText(">>");
leftButtons[2].setActionCommand("B2");
leftButtons[2].setText(">>");
leftButtons[3].setActionCommand("B3");
leftButtons[3].setText(">>");
rightButtons = new JButton[4]; // create array of JButtons
rightButtonJPanel = new JPanel(); // set up panel
rightButtonJPanel.setLayout(new GridLayout( rows: 5, cols: 1, hgap: 0, vgap: 10));
JLabel blankLabel1 = new JLabel();
rightButtonJPanel.add(blankLabel1);
for (int count = 0; count < 4; count++) {
    rightButtons[count] = new JButton();
    rightButtonJPanel.add(rightButtons[count]); // add button to JFrame
} // end for
//set right buttons
rightButtons[0].setActionCommand("B4");
rightButtons[0].setText("<<");
rightButtons[1].setActionCommand("B5");
rightButtons[1].setText("<<");
rightButtons[2].setActionCommand("B6");
rightButtons[2].setText("<<");
rightButtons[3].setActionCommand("B7");
rightButtons[3].setText("<<");
extraButtons = new JButton[3]; // create array of JButtons
extraButtonJPanel = new JPanel(); // set up panel
extraButtonJPanel.setLayout(new GridLayout( rows: 3, cols: 1, hgap: 0, vgap: 3));
for (int count = 0; count < 3; count++) {
    extraButtons[count] = new JButton();
    extraButtonJPanel.add(extraButtons[count]); // add button to JFrame
} // end for
//set three buttons
extraButtons[0].setActionCommand("Cancel");
extraButtons[0].setText("Cancel");
extraButtons[0].setBackground(Color.red);
extraButtons[1].setActionCommand("\n");
extraButtons[1].setText("Enter");
extraButtons[1].setBackground(Color.green);
extraButtons[2].setText("Clear");
extraButtons[2].setText("Clear");
extraButtons[2].setBackground(Color.yellow);
buttons = new JButton[names.length]; // create array of JButtons
buttonJPanel = new JPanel(); // set up panel
buttonJPanel.setLayout(new GridLayout( rows: 4, buttons.length, hgap: 3, vgap: 3));
for (int count = 0; count < names.length; count++) {
    buttons[count] = new JButton(names[count]);
    buttonJPanel.add(buttons[count]); // add button to JFrame
} // end for

```

These are the left buttons.



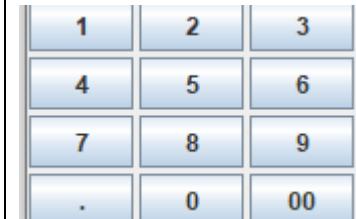
These are the right buttons.



These are the extra buttons.



These are the number pad buttons



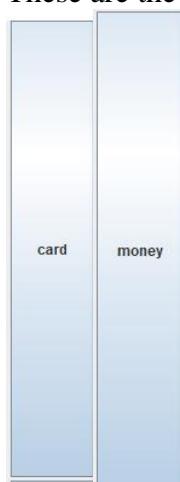
```

checkButtons = new JButton[2]; // create array of JButtons
checkButtonJPanel = new JPanel(); // set up panel
checkButtonJPanel.setLayout(new GridLayout( rows: 2, cols: 1, hgap: 0, vgap: 3));

for (int count = 0; count < 2; count++) {
    checkButtons[count] = new JButton();
    checkButtonJPanel.add(checkButtons[count]); // add button to JFrame
} // end for
checkButtons[0].setActionCommand("Card");
checkButtons[0].setText("card");
checkButtons[1].setActionCommand("Money");
checkButtons[1].setText("money");

```

These are the check buttons



```

frame.add(subPanel, BorderLayout.SOUTH);
frame.add(subPanel2, BorderLayout.CENTER);
frame.add(checkButtonJPanel, BorderLayout.EAST);
subPanel2.setLayout(new BorderLayout());
welcome();
subPanel2.add(leftButtonJPanel, BorderLayout.WEST);
subPanel2.add(rightButtonJPanel, BorderLayout.EAST);
subPanel.setLayout(new BorderLayout( hgap: 20, vgap: 10));
subPanel.add(buttonJPanel, BorderLayout.CENTER);
subPanel.add(extraButtonJPanel, BorderLayout.EAST);

// register event handlers
for (int i = 0; i <= 11; i++) {
    buttons[i].addActionListener(handler);
}
for (int i = 0; i <= 3; i++) {
    leftButtons[i].addActionListener(handler);
}
for (int i = 0; i <= 3; i++) {
    rightButtons[i].addActionListener(handler);
}
for (int i = 0; i <= 2; i++) {
    extraButtons[i].addActionListener(handler);
}
for (int i = 0; i <= 1; i++) {
    checkButtons[i].addActionListener(handler);
}
frame.validate();
frame.pack();

```

Calling the welcome gui in atm

Adding the handler for the buttons

Including

number pad,

left buttons,

right buttons,

extra buttons,

check buttons.

And add it to the panel then add to frame to display it.

frame.validate();  
frame.pack();

It is refresh the frame.

```

public void welcome(){ //welcome page
    subPanel3.setLayout(new BorderLayout());
    label.setIcon(new ImageIcon(new javax.swing.ImageIcon(getClass().getResource( name: "1.JPG")));
        | getImage().getScaledInstance( width: 700, height: 800, Image.SCALE_SMOOTH));
    label.setOpaque(true);
    subPanel3.add(label, BorderLayout.CENTER);
    subPanel2.add(subPanel3, BorderLayout.CENTER);
    subPanel3.setVisible(true);
    subPanel2.revalidate();
}

```

run:

```

public void run()
{
    // welcome and authenticate user; perform transactions
    while ( true )
    {
        // loop while user is not yet authenticated
        while ( !userAuthenticated )
        {
            welcome();
            cancelPressed = false; //initialize cancelPressed
            index = -1; //initialize index
            withdrawalconfirm = 0;//initialize withdrawal count
            screen.displayMessageLine( "\nWelcome!" );
            while(checkCardStatus == false){ //stop when the card is not inserted
                try {
                    Thread.sleep( millis: 100 );
                } catch (InterruptedException ex) {
                }
            }
            authenticateUser(); // authenticate user
        } // end while

        performAccount(); // user is now authenticated

        userAuthenticated = false; // reset before next ATM session
        currentAccountNumber = 0; // reset before next ATM session
        goodbye(); //set time
        screen.displayMessageLine( "\nThank you! Goodbye!" );
    } // end while
} // end method run

```

```

public void resetButtons(){
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.WEST));
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.EAST));
    subPanel2.setLayout(new GridLayout( rows: 5, cols: 1, hgap: 0, vgap: 10));
    subPanel2.setLayout(layout);
    subPanel1.setLayout(new GridLayout( rows: 1, cols: 1, hgap: 0, vgap: 10));
    subPanel1.remove(layout.getComponent(BorderLayout.EAST));
    subPanel1.setLayout(getLayoutComponent(BorderLayout.CENTER));
}

JButtons = new JButton[4]; // create array of JButtons
JButtonsPanel = new JPanel(); // set up panel
JButtonsPanel.setLayout(new GridLayout( rows: 5, cols: 1, hgap: 0, vgap: 10));
JLabel blankLabel = new JLabel();
JButtonsPanel.add(blankLabel);
for (int count = 0; count < 4; count++) {
    JButtons[count] = new JButton();
    JButtonsPanel.add(JButtons[count]); // add button to JPanle
} // end for
JButtons[0].setActionCommand("0");
JButtons[0].setText(">>>");
JButtons[1].setActionCommand("1");
JButtons[1].setText(">>>");
JButtons[2].setActionCommand("2");
JButtons[2].setText(">>>");
JButtons[3].setActionCommand("3");
JButtons[3].setText(">>>");

```

```

signButtons = new JButton[4]; // create array of Buttons
signButtonsPanel = new JPanel(); // set up panel
signButtonsPanel.setLayout(new GridLayout( rows: 5, cols: 1, hgap: 0, vgap: 10));
JLabel blankLabel = new JLabel();
signButtonsPanel.add(blankLabel);
for (int count = 0; count < 4; count++) {
    signButtons[count] = new JButton();
    signButtonsPanel.add(signButtons[count]); // add button to JPanle
} // end for

JButtons[0].setActionCommand("0");
JButtons[0].setText("<<<");
JButtons[1].setActionCommand("1");
JButtons[1].setText("<<<");
JButtons[2].setActionCommand("2");
JButtons[2].setText("<<<");
JButtons[3].setActionCommand("3");
JButtons[3].setText("<<<");

extraButtons = new JButton[3]; // create array of JButtons
extraButtonsPanel = new JPanel(); // set up panel
extraButtonsPanel.setLayout(new GridLayout( rows: 3, cols: 1, hgap: 0, vgap: 10));
for (int count = 0; count < 3; count++) {
    extraButtons[count] = new JButton();
    extraButtonsPanel.add(extraButtons[count]); // add button to JPanle
} // end for
extraButtons[0].setActionCommand("Cancel");
extraButtons[0].setText("Cancel");
extraButtons[0].setBackground(Color.red);
extraButtons[0].setForeground(Color.white);
extraButtons[1].setActionCommand("Ok");
extraButtons[1].setText("Ok");
extraButtons[1].setBackground(Color.green);
extraButtons[1].setForeground(Color.white);
extraButtons[2].setActionCommand("Clear");
extraButtons[2].setText("Clear");
extraButtons[2].setBackground(Color.yellow);

```

Here is the welcome gui which insert a jpg to be the icon and showing in the middle of the screen.

While the account number and PIN are not correct, it calls welcome() again.

Initializing cancelPressed, index and withdrawalconfirm.

It will stay in the welcome gui until the card is inserted.

After the card is inserted, it runs the steps of authenticating the user.

Then run the steps of withdrawing, viewing balance and transferring.

Finally, showing the goodbye page when then user leave the atm.

```

// initialize as new object of chosen type
currentTransaction =
    createTransaction( mainMenuSelection );
currentTransaction.execute(); // execute transaction

// determine the user press cancel or not during withdrawing or transferring mon
if(mainMenuSelection == WITHDRAWAL || mainMenuSelection == TRANSFER){
    currentTransaction.takeValue();
    System.out.println(currentTransaction.takeValue());
}
resetButtons();

```

Here is to re-add the buttons as we re-create the buttons when the program call transaction class.

```

buttons = new JButton[button.length]; // create array of buttons
buttonPanel = new JPanel(); // set up panel
buttonPanel.setLayout(new GridLayout(button.length, 1, 0, 0));
for (int count = 0; count < button.length; count++) {
    buttons[count] = new JButton(buttons[count]);
    buttonPanel.add(buttons[count]); // add button to frame
} // end for

```

```

leftPanel2.add(leftButtonPanel, BorderLayout.WEST);
leftPanel2.add(rightButtonPanel, BorderLayout.EAST);
leftPanel.add(extraButtonPanel, BorderLayout.EAST);
bottomPanel.add(buttonPanel, BorderLayout.CENTER);

```

```

handler = new ButtonHandler();
for (int i = 0; i <= 11; i++) {
    buttons[i].addActionListener(handler);
}
for (int i = 0; i <= 3; i++) {
    leftButtons[i].addActionListener(handler);
    rightButtons[i].addActionListener(handler);
}
for (int i = 0; i <= 2; i++) {
    extraButtons[i].addActionListener(handler);
}

subPanel2.revalidate();
subPanel1.revalidate();

```

```

//remove the panel which are showing
private void reset(){

```

```

    accountPanelBlank1.removeAll();
    accountPanelBlank2.removeAll();
    accountPanel1.removeAll();
    accountPanel2.removeAll();
    accountPanel3.removeAll();
    accountPanel4.removeAll();
    accountPanel5.removeAll();
    accountPanel6.removeAll();
    accountPanel7.removeAll();
    accountPanel8.removeAll();
    accountPanelBlank1.setBackground(atmBlue);
    accountPanelBlank2.setBackground(atmBlue);
    accountPanel1.setBackground(atmBlue);
    accountPanel2.setBackground(atmBlue);
    accountPanel3.setBackground(atmBlue);
    accountPanel4.setBackground(atmBlue);
    accountPanel5.setBackground(atmBlue);
    accountPanel6.setBackground(atmBlue);
    accountPanel7.setBackground(atmBlue);
    accountPanel8.setBackground(atmBlue);
}
```

```

private void authenticateUser()
{

```

```

    pass(); //Showing the page which entering PIN
    while(enterPassReturn == true) { //Stop until the user enters the account and presses "Enter"
        try {
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {
        }
    }
    screen.displayMessage( "\nPlease enter your account number: " );
    int accountNumber = accountNumberInput; //get the account number from gui
    System.out.println(accountNumber); //display in terminal

    //If user press cancel, leave the page of entering account number
    if(cancelPressed == true){
        enterPINReturn = false;
        accountNumber = 0;
        userAuthenticated = true;
    }
    else
        enterPINReturn = true;
}

```

Here is to reset the panel we use so that the next time we use do not have the previous message we set before.

For the `authenticateUser()`, It calls the `pass()` which is a gui for user to enter account number and PIN.

`EnterPassReturn` will be true until the user entering the account number in `pass()`

Then, the input will be saved into `accountNumber`.

```

screen.displayMessage( "\nEnter your PIN: " ); // prompt for PIN

while(enterPINReturn == true) { //Stop until the user enters the PIN and presses "Enter"
    try {
        Thread.sleep( millis: 100 );
    } catch (InterruptedException ex) {
    }
}

system.out.println(pinInput); //display in terminal

int pin;
//If user press cancel, leave the page of entering PIN
if(cancelPressed == true && enterPINReturn == false){
    pin = 0;
    userAuthenticated = true;
}
else{ //
    pin = Integer.parseInt(pinInput); //get the PIN from gui
    System.out.println(pinInput);
    // set userAuthenticated to boolean value returned by database
    userAuthenticated =
        bankDatabase.authenticateUser( userAccountNumber: accountNumber*1000+1, pin );
}

if ( userAuthenticated )
{
    currentAccountNumber = accountNumber; // save user's account #
} // end if
else {
    screen.displayMessageLine(
        "Invalid account number or PIN. Please try again.");
    invalidconfirm = true;
    invalidpin(); //Show the wrong message
    do {
        try {
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {
        }
    }while(invalidconfirm);
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    checkCardStatus = false; //return the card
    checkButtons[0].setBackground(null); //off the red light of "card" button
}

```

After input the account number.

EnterPassReturn will be false.

Then, enterPINReturn will true until the user entering the PIN in pass().

The input will be saved into pin.

When EnterPassReturn and enterPINReturn are false, that means the user entered the account number and PIN.

UserAuthenticated() will check the account number and PIN are correct or not.

If not, it will call invalidpin() to show a wrong message and let user press “enter” to try again.

```

public void pass(){
    enterPassReturn = true;
    reset();
    JPanel userPanel = new JPanel();
    userPanel.setLayout(new GridBagLayout());
    userPanel.setBackground(atmBlue);

    accountPanel1.setBackground(choiceBlue);
    accountPanel2.setBackground(choiceBlue);

    JLabel messageLabel1 = new JLabel();
    messageLabel1.setText("Step 1 Please type in the account number");
    messageLabel1.setFont(new Font("name", "Serif", Font.PLAIN));
    messageLabel1.setHorizontalTextPosition(JLabel.CENTER);
    messageLabel1.setVerticalTextPosition(JLabel.CENTER);

    JLabel messageLabel2 = new JLabel();
    messageLabel2.setText("Step 2 then press \"Enter\" to continue");
    messageLabel2.setFont(new Font("name", "Serif", Font.PLAIN));
    messageLabel2.setHorizontalTextPosition(JLabel.CENTER);
    messageLabel2.setVerticalTextPosition(JLabel.CENTER);

    JLabel userLabel = new JLabel();
    userLabel.setText("Account Number");
    userLabel.setFont(new Font("name", "Serif", Font.PLAIN));
    userLabel.setHorizontalTextPosition(JLabel.RIGHT);
    userLabel.setVerticalTextPosition(JLabel.CENTER);

    JLabel passwordLabel = new JLabel();
    passwordLabel.setText("PIN");
    passwordLabel.setFont(new Font("name", "Serif", Font.PLAIN));
    passwordLabel.setHorizontalTextPosition(JLabel.RIGHT);
    passwordLabel.setVerticalTextPosition(JLabel.CENTER);

    accountPanel1Blank1.setLayout(new BorderLayout());
    accountPanel1Blank1.add(messageLabel1);
    accountPanel1Blank2.setLayout(new BorderLayout());
    accountPanel1Blank2.add(messageLabel2);
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(userLabel);
    accountPanel2.setLayout(new BorderLayout());
    accountPanel2.add(passwordLabel);

    GridBagConstraints c0 = new GridBagConstraints();
    c0.insets = new Insets( top: 0, left: 0, bottom: 0, right: 5 );
    c0.gridx = 0;
    c0.gridy = 2;
    c0.gridwidth = 1;
    c0.gridheight = 1;
    c0.weightx = 0.58;
    c0.weighty = 0.58;
    c0.fill = GridBagConstraints.HORIZONTAL;
    c0.anchor = GridBagConstraints.WEST;
    userPanel.add(accountPanel1, c0);

    // declaration of textField for displaying output
    passTextField.setEditable(false);
    passTextField.setText(linePass); // display line1 in textField
    linePass = "";
    passTextField.setText("");

    GridBagConstraints c1 = new GridBagConstraints();
    c1.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0 );
    c1.gridx = 0;
    c1.gridy = 1;
    c1.gridwidth = 4;
    c1.gridheight = 1;
    c1.weightx = 0.47;
    c1.weighty = 0.47;
    c1.fill = GridBagConstraints.BOTH;
    c1.anchor = GridBagConstraints.REST;
    userPanel.add(accountPanel1Blank1, c1);

    GridBagConstraints c2 = new GridBagConstraints();
    c2.insets = new Insets( top: 0, left: 0, bottom: 0, right: 5 );
    c2.gridx = 0;
    c2.gridy = 0;
    c2.gridwidth = 2;
    c2.gridheight = 1;
    c2.weightx = 0.58;
    c2.weighty = 0.58;
    c2.fill = GridBagConstraints.HORIZONTAL;
    c2.anchor = GridBagConstraints.CENTER;
    userPanel.add(accountPanel1Blank2, c2);
}

```

```

GridBagConstraints c3 = new GridBagConstraints();
c3.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0 );
c3.gridx = 0;
c3.gridy = 1;
c3.gridwidth = 3;
c3.gridheight = 1;
c3.weightx = 0.58;
c3.weighty = 0.58;
c3.fill = GridBagConstraints.HORIZONTAL;
c3.anchor = GridBagConstraints.CENTER;
userPanel.add(passTextField, c3);

GridBagConstraints c4 = new GridBagConstraints();
c4.insets = new Insets( top: 0, left: 0, bottom: 0, right: 5 );
c4.gridx = 0;
c4.gridy = 3;
c4.gridwidth = 1;
c4.gridheight = 1;
c4.weightx = 0.47;
c4.weighty = 0.47;
c4.fill = GridBagConstraints.HORIZONTAL;
c4.anchor = GridBagConstraints.WEST;
userPanel.add(accountPanel12, c4);

```

Here is pass() which is displaying the usernames and password gui. which we use four JLabel to show the message also set the font and add JTextField and JPasswordField.

```

passwordText.setEditable(false);
passwordText.setText(linePass); // display line1 in textField
linePass = "";
passwordText.setText("");

GridBagConstraints c5 = new GridBagConstraints();
c5.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0 );
c5.gridx = 1;
c5.gridy = 3;
c5.gridwidth = 3;
c5.gridheight = 1;
c5.weightx = 0.47;
c5.weighty = 0.47;
c5.fill = GridBagConstraints.HORIZONTAL;
c5.anchor = GridBagConstraints.WEST;
userPanel.add(passwordText, c5);

subPanel2.add(userPanel, BorderLayout.CENTER);
userPanel.setVisible(true);
subPanel2.revalidate();

```

```

public void accountMenu(){
    chooseAccount = true;
    reset();
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));

    accountPanel = new JPanel();
    accountPanel.setLayout(new GridLayout( 5, 2, 10, 10));
    accountPanel.setBackground(atmBlue);
    accountPanel1.setBackground(choiceBlue);
    accountPanel2.setBackground(choiceBlue);

    JLabel accountLabel = new JLabel();
    accountLabel.setText("Account");
    accountLabel.setFont(new Font("Serif", Font.PLAIN, 35));
    accountLabel.setHorizontalTextPosition(JLabel.RIGHT);
    accountLabel.setVerticalTextPosition(JLabel.CENTER);

    JLabel accountLabel1 = new JLabel();
    accountLabel1.setText("Menu");
    accountLabel1.setFont(new Font("Serif", Font.PLAIN, 35));
    accountLabel1.setHorizontalTextPosition(JLabel.LEFT);
    accountLabel1.setVerticalTextPosition(JLabel.CENTER);

    JLabel savingLabel = new JLabel();
    savingLabel.setText("Saving account");
    savingLabel.setFont(new Font("Serif", Font.PLAIN, 30));
    savingLabel.setHorizontalTextPosition(JLabel.CENTER);
    savingLabel.setVerticalTextPosition(JLabel.CENTER);

    JLabel chequeLabel = new JLabel();
    chequeLabel.setText("Cheque account");
    chequeLabel.setFont(new Font("Serif", Font.PLAIN, 30));
    chequeLabel.setHorizontalTextPosition(JLabel.CENTER);
    chequeLabel.setVerticalTextPosition(JLabel.CENTER);

    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.add(accountLabel);
    accountPanelBlank2.setLayout(new BorderLayout());
    accountPanelBlank2.add(accountLabel1);

    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(savingLabel);
    accountPanel2.setLayout(new BorderLayout());
    accountPanel2.add(chequeLabel);

    accountPanel.add(accountPanelBlank1);
    accountPanel.add(accountPanelBlank2);
    accountPanel.add(accountPanel1);
    accountPanel.add(accountPanel2);
    accountPanel.add(accountPanel3);
    accountPanel.add(accountPanel4);
    accountPanel.add(accountPanel5);
    accountPanel.add(accountPanel6);
    accountPanel.add(accountPanel7);
    accountPanel.add(accountPanel8);

    subPanel2.add(accountPanel, BorderLayout.CENTER);
    accountPanel.setVisible(true);
    subPanel2.revalidate();
}

```

Here is the GUI displaying to let the user choose use which account. We use 4 JLabel to display the message.

```

//GUI of saving account's menu
public void savingMenu(){
    chooseSaving = true;
    reset();
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));

    savingPanel = new JPanel();
    savingPanel.setLayout(new GridLayout( 5, 2, 10, 10));
    savingPanel.setBackground(atmBlue);

    accountPanel1.setBackground(choiceBlue);
    accountPanel3.setBackground(choiceBlue);
    accountPanel5.setBackground(choiceBlue);

    JLabel accountLabel = new JLabel();
    accountLabel.setText("Saving");
    accountLabel.setFont(new Font("Serif", Font.PLAIN, 35));
    accountLabel.setHorizontalTextPosition(JLabel.RIGHT);
    accountLabel.setVerticalTextPosition(JLabel.CENTER);

    JLabel accountLabel1 = new JLabel();
    accountLabel1.setText("Menu");
    accountLabel1.setFont(new Font("Serif", Font.PLAIN, 35));
    accountLabel1.setHorizontalTextPosition(JLabel.LEFT);
    accountLabel1.setVerticalTextPosition(JLabel.CENTER);

    JLabel balanceLabel = new JLabel();
    balanceLabel.setText("View My Balance");
    balanceLabel.setFont(new Font("Serif", Font.PLAIN, 30));
    balanceLabel.setHorizontalTextPosition(JLabel.CENTER);
    balanceLabel.setVerticalTextPosition(JLabel.CENTER);

    JLabel withdrawLabel = new JLabel();
    withdrawLabel.setText("Withdraw Cash");
    withdrawLabel.setFont(new Font("Serif", Font.PLAIN, 30));
    withdrawLabel.setHorizontalTextPosition(JLabel.CENTER);
    withdrawLabel.setVerticalTextPosition(JLabel.CENTER);

    JLabel transferLabel = new JLabel();
    transferLabel.setText("Transfer");
    transferLabel.setFont(new Font("Serif", Font.PLAIN, 30));
    transferLabel.setHorizontalTextPosition(JLabel.CENTER);
    transferLabel.setVerticalTextPosition(JLabel.CENTER);

    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.add(accountLabel);
    accountPanelBlank2.setLayout(new BorderLayout());
    accountPanelBlank2.add(accountLabel1);

    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(balanceLabel);
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(withdrawLabel);
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(transferLabel);

    savingPanel.add(accountPanelBlank1);
    savingPanel.add(accountPanelBlank2);
    savingPanel.add(accountPanel1);
    savingPanel.add(accountPanel3);
    savingPanel.add(accountPanel4);
    savingPanel.add(accountPanel5);
    savingPanel.add(accountPanel6);
    savingPanel.add(accountPanel7);
    savingPanel.add(accountPanel8);

    subPanel2.add(savingPanel, BorderLayout.CENTER);
    savingPanel.setVisible(true);
    subPanel2.revalidate();
}

```

Here is the GUI displaying saving account. We use 5 JLabel to display the message.

```

public void chequeMenu(){
    chooseCheque = true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));

    chequePanel = new JPanel();
    chequePanel.setLayout(new GridLayout( 5, 2, 10, 10));
    chequePanel.setBackground(atmBlue);

    accountPanel1.setBackground(choiceBlue);
    accountPanel3.setBackground(choiceBlue);
    accountPanel5.setBackground(choiceBlue);

    JLabel accountLabel = new JLabel();
    accountLabel.setText("Cheque");
    accountLabel.setFont(new Font( "Serif", Font.PLAIN, 35));
    accountLabel.setHorizontalTextAlignment(JLabel.RIGHT);
    accountLabel.setVerticalTextAlignment(JLabel.CENTER);

    JLabel accountLabel1 = new JLabel();
    accountLabel1.setText("Menu");
    accountLabel1.setFont(new Font( "Serif", Font.PLAIN, 35));
    accountLabel1.setHorizontalTextAlignment(JLabel.LEFT);
    accountLabel1.setVerticalTextAlignment(JLabel.CENTER);

    JLabel balanceLabel = new JLabel();
    balanceLabel.setText("View My Balance");
    balanceLabel.setFont(new Font( "Serif", Font.PLAIN, 30));
    balanceLabel.setHorizontalTextAlignment(JLabel.CENTER);
    balanceLabel.setVerticalTextAlignment(JLabel.CENTER);

    JLabel withdrawLabel = new JLabel();
    withdrawLabel.setText("Withdraw Cash");
    withdrawLabel.setFont(new Font( "Serif", Font.PLAIN, 30));
    withdrawLabel.setHorizontalTextAlignment(JLabel.CENTER);
    withdrawLabel.setVerticalTextAlignment(JLabel.CENTER);

    JLabel transferLabel = new JLabel();
    transferLabel.setText("Transfer");
    transferLabel.setFont(new Font( "Serif", Font.PLAIN, 30));
    transferLabel.setHorizontalTextAlignment(JLabel.CENTER);
    transferLabel.setVerticalTextAlignment(JLabel.CENTER);

    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.add(accountLabel);
    accountPanelBlank2.setLayout(new BorderLayout());
    accountPanelBlank2.add(accountLabel1);

    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(balanceLabel);
    accountPanel3.setLayout(new BorderLayout());
    accountPanel3.add(withdrawLabel);
    accountPanel5.setLayout(new BorderLayout());
    accountPanel5.add(transferLabel);

    chequePanel.add(accountPanelBlank1);
    chequePanel.add(accountPanelBlank2);
    chequePanel.add(accountPanel1);
    chequePanel.add(accountPanel2);
    chequePanel.add(accountPanel3);
    chequePanel.add(accountPanel4);
    chequePanel.add(accountPanel5);
    chequePanel.add(accountPanel6);
    chequePanel.add(accountPanel7);
    chequePanel.add(accountPanel8);

    subPanel2.add(chequePanel, BorderLayout.CENTER);
    chequePanel.setVisible(true);
    subPanel2.revalidate();
}

```

Here is the GUI displaying chequeaccount. We use 5 JLabel to display the message.

### Displayaccount:

```

private int displayAccount()
{
    //return exit when the user withdrew the money or pressed cancel
    if(withdrawalconfirm == 2 || cancelPressed == true)
        return 3;

    screen.displayMessageLine( "\nAccount menu: " );
    screen.displayMessageLine( "1 - View my Saving account" );
    screen.displayMessageLine( "2 - View my Cheque account" );
    screen.displayMessageLine( "3 - Exit\n" );
    screen.displayMessage( "Enter a choice: " );

    while(chooseAccount == true || index == -1) { //stop until user choose the options
        try {
            accountMenu();
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {
        }
    }

    accountindex = index; //record the choice of account
    return index;
} // end method displayMainMenu

```

If the account number and PIN is correct.

It calls the accountMenu() to let user choose the saving account or cheque account.

And save the choice of user into accountindex, then return index(the choice of user.)

Performaccount:

```
private void performAccount()//display account menu
{
    boolean userExited = false; // user has not chosen to exit
    // loop while user has not chosen option to exit system
    while ( !userExited )
    {
        x = displayAccount();
        switch (x) {
            case 1, 2 -> {
                currentAccountNumber =currentAccountNumber*1000+x;
                performTransactions();
            }
            // user chose to terminate session
            case 3 -> {
                screen.displayMessageLine("\nExiting the system...");
                userExited = true; // this ATM session should end
            }
            default -> {
                invalidMessage();
                enteReturn = true;
                while(enteReturn == true){
                    try {
                        Thread.sleep( millis: 100 );
                    } catch (InterruptedException ex) {

                    }
                }
                screen.displayMessageLine(
                    "\nYou did not enter a valid selection. Try again.");
            }
        } // end switch
    } // end while
} // end method performTransactions
```

After the user choose the saving account or cheque account, it will return a value into x (return index).

Choosing saving account will go to case 1

Choosing cheque account will go to case 2

Press cancel button will go to case 3

OR, it will display the wrong message.

## DisplayMainMenu:

```

private int displayMainMenu()
{
    reset();
    //return exit when the user withdrew the money or pressed cancel
    if(withdrawalconfirm == 2 || cancelPressed == true)
        return 4;
    //showing the menu of saving account or cheque account
    if(accountindex == 1)
        savingMenu();
    else if(accountindex == 2)
        chequeMenu();

    screen.displayMessageLine( "\nMain menu:" );
    screen.displayMessageLine( "1 - View my balance" );
    screen.displayMessageLine( "2 - Withdraw cash" );
    screen.displayMessageLine( "3 - Transfer" );
    screen.displayMessageLine( "4 - Exit\n" );
    screen.displayMessage( "Enter a choice: " );

    if(accountindex == 1){
        while(chooseSaving == true) {//stop until user choose the options
            try {
                Thread.sleep( millis: 100 );
            } catch (InterruptedException ex) {
            }
        }
    }
    else if(accountindex == 2){
        while(chooseCheque == true) {//stop until user choose the options
            try {
                Thread.sleep( millis: 100 );
            } catch (InterruptedException ex) {
            }
        }
    }
    return index;// return user's selection
}

```

If user choose saving account, it calls the savingMenu().

Else if user choose cheque account, it calls the chequeMenu().

Saving the choice of savingMenu() / chequeMenu() into index, then return index.

## CreateTransaction:

```

switch ( type )
{
    case BALANCE_INQUIRY: // create new BalanceInquiry transaction
        temp = new BalanceInquiry(
            currentAccountNumber, screen, bankDatabase, subPanel2);
        break;
    case WITHDRAWAL: // create new Withdrawal transaction
        temp = new Withdrawal( currentAccountNumber, screen,
            bankDatabase, keypad, cashDispenser, subPanel2, subPanel);
        break;
    case TRANSFER: // create new Transfer transaction
        temp = new Transfer( currentAccountNumber, screen,
            bankDatabase, keypad, subPanel2, subPanel);
        break;
} // end switch

```

Here we add that put in subPanel2 and subPanel to the BalanceInquiry and Withdrawal and Transfer so that can re-add the button and change the display.

```

public BalanceInquiry( int userAccountNumber, Screen atmScreen,
                      BankDatabase atmBankDatabase, JPanel Panel )
{
    super( userAccountNumber, atmScreen, atmBankDatabase );
    subPanel2 = Panel;
} // end BalanceInquiry constructor

public Withdrawal(int userAccountNumber, Screen atmScreen,
                  BankDatabase atmBankDatabase, Keypad atmKeypad,
                  CashDispenser atmCashDispenser, JPanel panel ,JPanel panel1)
{
    // initializing variables
// Deposit constructor
public Transfer(int userAccountNumber, Screen atmScreen,
                BankDatabase atmBankDatabase, Keypad atmKeypad, JPanel panel ,JPanel panel1 )
{
    // initializing variables
}

```

Performtransaction:

```

private void performTransactions()
{
    // local variable to store transaction currently being processed
    Transaction currentTransaction = null;
    boolean userExited = false; // user has not chosen to exit
    // loop while user has not chosen option to exit system
    while ( !userExited )
    {
        // show main menu and get user selection
        int mainMenuSelection = displayMainMenu();

        // decide how to proceed based on user's menu selection
        switch ( mainMenuSelection )
        {
            // user chose to perform one of three transaction types
            case BALANCE_INQUIRY:
            case WITHDRAWAL:
            case TRANSFER:
                // initialize as new object of chosen type
                currentTransaction =
                    createTransaction( mainMenuSelection );
                currentTransaction.execute(); // execute transaction

                //determine the user press cancel or not during withdrawing or transferring money
                if(mainMenuSelection == WITHDRAWAL || mainMenuSelection == TRANSFER){
                    currentTransaction.takevalue();
                    System.out.println(currentTransaction.takevalue());
                }
                resetButtons();
                enteReturn = true;
                while(enteReturn == true){
                    try {
                        Thread.sleep( millis: 100);
                    } catch (InterruptedException ex) {
                    }
                }
        }
    }
}

```

The index of **DisplayMainMenu()** will be saved into **mainMenuSelection**.

If it is 1, it will go to case **BALANCE\_INQUIRY**

2, it will go to case **WITHDRAWAL**

3, it will go to case **TRANSFER**

If user pressed cancel in **WITHDRAWAL** or **TRANSFER**, takevalue will be 1, else it will be 0.

```

//let user choose to print the advice after withdrawing money
if(mainMenuSelection == 2){
    withdrawalconfirm = 1;
    while(withdrawalconfirm == 1){
        try {
            if(currentTransaction.takevalue() == 1)
                withdrawalconfirm = 2;
            else
                advice();
            Thread.sleep( millis: 100);
        } catch (InterruptedException ex) {
        }
    }

    if(currentTransaction.takevalue() == 0) {
        //Asking user take the card
        takeCard = true;
        while(takeCard ==true){
            try {
                CardAndMoney();
                Thread.sleep( millis: 100);
            } catch (InterruptedException ex) {
            }
        }
        //Asking user take the money
        takeMoney = true;
        while(takeMoney ==true){
            try {
                CardAndMoney();
                Thread.sleep( millis: 100);
            } catch (InterruptedException ex) {
            }
        }
    }
    //return the card after the user press cancel
    else if(currentTransaction.takevalue() == 1) {
        checkCardStatus = false;
        checkButtons[0].setBackground(null);
    }
}

```

In the end of the WITHDRAW, it calls the advice(), and let the user choose to print the advice or not.

The user has to take the card first, then take the money.

When takeCard is true, CardAndMoney will show the text and ask user to take the card.

When takeMoney is true, CardAndMoney will show the text and ask user to take money.

After that, the checkcardstatus will be false and the “Card” button will turn off the red light.

```

//let user choose to print the advice after transferring money
if(mainMenuSelection == 3) {
    withdrawalconfirm = 1;
    while(withdrawalconfirm == 1) {
        try {
            if (currentTransaction.takevalue() == 1)
                withdrawalconfirm = 2;
            else
                advice();
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {
        }
    }
    reset();
    checkCardStatus = false;
    checkButtons[0].setBackground(null);
}

break;

case EXIT: // user chose to terminate session

    screen.displayMessageLine( "\nExiting the system..." );
    userExited = true; // this ATM session should end
    currentAccountNumber=(currentAccountNumber-x)/1000;

    break;

default: // user did not enter an integer from 1-4
{
    invalidMessage();
    enteReturn = true;
    while(enteReturn == true){
        try {
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {

        }
    }
    screen.displayMessageLine(
        "\nYou did not enter a valid selection. Try again.");
}
break;

```

In the end of the TRANSFER, it also calls the advice(), and let the user choose to print the advice or not.

After that, the checkcardstatus will be false and the “Card” button will turn off the red light.

If you pressed cancel, it will go to case EXIT.

Or it will show the wrong message.

```

public void invalidMessage(){
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel invalidPanel = new JPanel();
    invalidPanel.setLayout(new GridLayout( 3, 1));
    invalidPanel.setBackground(atmBlue);

    JLabel invalidLabel = new JLabel();
    invalidLabel.setText("Invalid selection.");
    invalidLabel.setFont(new Font( "Serif", Font.PLAIN, 30));
    invalidLabel.setHorizontalTextPosition(JLabel.CENTER);
    invalidLabel.setVerticalTextPosition(JLabel.CENTER);
    accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    invalidPanel.add(accountPanel1);

    JLabel invalidLabel2 = new JLabel();
    invalidLabel2.setText("Please try again.");
    invalidLabel2.setFont(new Font( "Serif", Font.PLAIN, 30));
    invalidLabel2.setHorizontalTextPosition(JLabel.CENTER);
    invalidLabel2.setVerticalTextPosition(JLabel.CENTER);
    accountPanel3 = new JPanel();
    accountPanel3.setLayout(new BorderLayout());
    invalidPanel.add(accountPanel3);

    JLabel invalidLabel3 = new JLabel();
    invalidLabel3.setText("Please press \'Enter\' to continue.");
    invalidLabel3.setFont(new Font( "Serif", Font.PLAIN, 30));
    invalidLabel3.setHorizontalTextPosition(JLabel.RIGHT);
    invalidLabel3.setVerticalTextPosition(JLabel.CENTER);
    accountPanel5 = new JPanel();
    accountPanel5.setLayout(new BorderLayout());
    invalidPanel.add(accountPanel5);

    public void invalidpin(){
        reset();
        BorderLayout layout = (BorderLayout)subPanel2.getLayout();
        subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
        JPanel errorPanel = new JPanel();
        errorPanel.setLayout(new GridLayout( 3, 1));
        errorPanel.setBackground(atmBlue);

        JLabel errorLabel = new JLabel();
        errorLabel.setText("Invalid account number or PIN");
        errorLabel.setFont(new Font( "Serif", Font.PLAIN, 35));
        errorLabel.setHorizontalTextPosition(JLabel.CENTER);
        errorLabel.setVerticalTextPosition(JLabel.CENTER);
        accountPanel1.setLayout(new BorderLayout());
        errorPanel.add(accountPanel1);

        JLabel errorLabel1 = new JLabel();
        errorLabel1.setText("Please try again");
        errorLabel1.setFont(new Font( "Serif", Font.PLAIN, 35));
        errorLabel1.setHorizontalTextPosition(JLabel.CENTER);
        errorLabel1.setVerticalTextPosition(JLabel.CENTER);
        accountPanel3.setLayout(new BorderLayout());
        errorPanel.add(accountPanel3);

        accountPanel1.setLayout(new BorderLayout());
        accountPanel1.add(errorLabel);
        accountPanel3.setLayout(new BorderLayout());
        accountPanel3.add(errorLabel1);

        subPanel2.add(errorPanel, BorderLayout.CENTER);
        errorPanel.setVisible(true);
        subPanel2.revalidate();
    }
}

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

Here is setting two JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment, at last put it in a panel which set to gridlayout to display.

```

public void advice(){
    reset();
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));

    JPanel advicePanel = new JPanel();
    advicePanel.setLayout(new GridLayout( 5, 2, 10, 10));
    advicePanel.setBackground(atmBlue);

    accountPanel3.setBackground(choiceBlue);
    accountPanel4.setBackground(choiceBlue);

    JLabel blankTitleLabel1 = new JLabel();
    blankTitleLabel1.setText("Your action");
    blankTitleLabel1.setFont(new Font("Serif", Font.PLAIN, 35));
    blankTitleLabel1.setHorizontalTextPosition(JLabel.RIGHT);
    blankTitleLabel1.setVerticalTextPosition(JLabel.CENTER);

    JLabel blankTitleLabel2 = new JLabel();
    blankTitleLabel2.setText("is processed.");
    blankTitleLabel2.setFont(new Font("Serif", Font.PLAIN, 35));
    blankTitleLabel2.setHorizontalTextPosition(JLabel.LEFT);
    blankTitleLabel2.setVerticalTextPosition(JLabel.CENTER);

    JLabel importantLabel1 = new JLabel();
    importantLabel1.setText("Please");
    importantLabel1.setFont(new Font("Serif", Font.PLAIN, 35));
    importantLabel1.setHorizontalTextPosition(JLabel.RIGHT);
    importantLabel1.setVerticalTextPosition(JLabel.CENTER);

    JLabel importantLabel2 = new JLabel();
    importantLabel2.setText("Select");
    importantLabel2.setFont(new Font("Serif", Font.PLAIN, 35));
    importantLabel2.setHorizontalTextPosition(JLabel.LEFT);
    importantLabel2.setVerticalTextPosition(JLabel.CENTER);

    JLabel step1Label = new JLabel();
    step1Label.setText("Print Advice");
    step1Label.setFont(new Font("Serif", Font.PLAIN, 30));
    step1Label.setHorizontalTextPosition(JLabel.CENTER);
    step1Label.setVerticalTextPosition(JLabel.CENTER);

    JLabel step2Label = new JLabel();
    step2Label.setText("Without Advice");
    step2Label.setFont(new Font("Serif", Font.PLAIN, 30));
    step2Label.setHorizontalTextPosition(JLabel.CENTER);
    step2Label.setVerticalTextPosition(JLabel.CENTER);

    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.add(blankTitleLabel1);
    accountPanelBlank2.setLayout(new BorderLayout());
    accountPanelBlank2.add(blankTitleLabel2);
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(importantLabel1);
    accountPanel2.setLayout(new BorderLayout());
    accountPanel2.add(importantLabel2);
    accountPanel3.setLayout(new BorderLayout());
    accountPanel3.add(step1Label);
    accountPanel4.setLayout(new BorderLayout());
    accountPanel4.add(step2Label);
    accountPanel5.setLayout(new BorderLayout());

    advicePanel.add(accountPanelBlank1);
    advicePanel.add(accountPanelBlank2);
    advicePanel.add(accountPanel1);
    advicePanel.add(accountPanel2);
    advicePanel.add(accountPanel3);
    advicePanel.add(accountPanel4);

    subPanel2.add(advicePanel, BorderLayout.CENTER);
    advicePanel.setVisible(true);
    subPanel2.revalidate();
}

```

Here is setting six JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment, at last put it in a panel which set to gridlayout to display the advice whether print it or not.

```

public void CardAndMoney(){
    reset();
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));

    JPanel CardAndMoneyPanel = new JPanel();
    CardAndMoneyPanel.setLayout(new GridLayout( rows: 5, cols: 2, hgap: 10, vgap: 10));
    CardAndMoneyPanel.setBackground(atmBlue);

    JLabel CMLabel = new JLabel();

    if(takeCard == true){
        CMLabel.setText("Step 1: Please take your card.");
        CMLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 40));
        CMLabel.setHorizontalTextPosition(JLabel.CENTER);
        CMLabel.setVerticalTextPosition(JLabel.CENTER);
        accountPanel1.setLayout(new BorderLayout());
        CardAndMoneyPanel.add(accountPanel1);
    }

    if(takeMoney == true){
        CMLabel.setText("Step 2: Please take the money.");
        CMLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 40));
        CMLabel.setHorizontalTextPosition(JLabel.CENTER);
        CMLabel.setVerticalTextPosition(JLabel.CENTER);
        accountPanel1.setLayout(new BorderLayout());
        CardAndMoneyPanel.add(accountPanel1);
    }

    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(CMLabel);

    CardAndMoneyPanel.add(accountPanelBlank1);
    CardAndMoneyPanel.add(accountPanelBlank2);
    CardAndMoneyPanel.add(accountPanel1);

    subPanel2.add(CardAndMoneyPanel, BorderLayout.CENTER);
    CardAndMoneyPanel.setVisible(true);
    subPanel2.revalidate();
}

```

Step 2: Please take the money.



Here is set the two CMLabel and put it in the panel. Also. the panel set layout to borderlayout so it will display in the right alignment, at last put it in a panel which set to gridlayout to display the two steps of take money. For the if-statement, user must press the red button to act like really take card and money in real life.

```

public void goodbye(){
    reset();
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout( 3, 1));
    errorPanel.setBackground(atmBlue);

    JLabel errorLabel = new JLabel();
    errorLabel.setText("Thank You");
    errorLabel.setFont(new Font( "Serif", Font.PLAIN, 35));
    errorLabel.setHorizontalTextPosition(JLabel.CENTER);
    errorLabel.setVerticalTextPosition(JLabel.CENTER);
    accountPanel1.setLayout(new BorderLayout());
    errorPanel.add(accountPanel1);

    JLabel errorLabel2 = new JLabel();
    errorLabel2.setText("Please using our ATM next time.");
    errorLabel2.setFont(new Font( "Serif", Font.PLAIN, 35));
    errorLabel2.setHorizontalTextPosition(JLabel.CENTER);
    errorLabel2.setVerticalTextPosition(JLabel.CENTER);
    accountPanel2.setLayout(new BorderLayout());
    errorPanel.add(accountPanel2);

    JLabel errorLabel3 = new JLabel();
    errorLabel3.setText("Please wait 5 seconds!");
    errorLabel3.setFont(new Font( "Serif", Font.PLAIN, 35));
    errorLabel3.setHorizontalTextPosition(JLabel.RIGHT);
    errorLabel3.setVerticalTextPosition(JLabel.CENTER);
    accountPanel3.setLayout(new BorderLayout());
    errorPanel.add(accountPanel3);

    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(errorLabel);
    accountPanel2.setLayout(new BorderLayout());
    accountPanel2.add(errorLabel2);
    accountPanel3.setLayout(new BorderLayout());
    accountPanel3.add(errorLabel3);

    subPanel2.add(errorPanel, BorderLayout.CENTER);
    errorPanel.setVisible(true);
    subPanel2.revalidate();
}

try {
    Thread.sleep( 5000);
} catch (InterruptedException ex) {
}
subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment, at last put it in a panel which set to gridlayout to display.

Here is set when display this GUI will wait 5 seconds then remove the subPanel2 then in the ATM system after show this screen will go to welcome.

```

public class ButtonHandler implements ActionListener
{
    public void actionPerformed(ActionEvent event )
    {
        //the action of pressing "cancel" button.
    }
}

```

Here is the ButtonHandler

```

//the action of pressing "cancel" button.
//leave the gui and return the card.
if(event.getActionCommand().equals("Cancel") && (enterPassReturn == true || enterPINReturn == true || chooseAccount == true || chooseSaving == true || chooseCheque == true)){
    cancelPressed = true;
    enterPassReturn = false;
    enterPINReturn = false;
    if(chooseAccount == true)
        index = 3; //exit
    chooseAccount = false;
    if(chooseSaving == true || chooseCheque == true)
        index = 4; //exit
    chooseSaving = false;
    chooseCheque = false;
    checkCardStatus = false;
    checkButtons[0].setBackground(null);
}
else{// if user does not press the "cancel" button
}

```

Cancel

Here is once the cancel button clicked then will set different variable to the value, we want so that it can return to welcome pages.

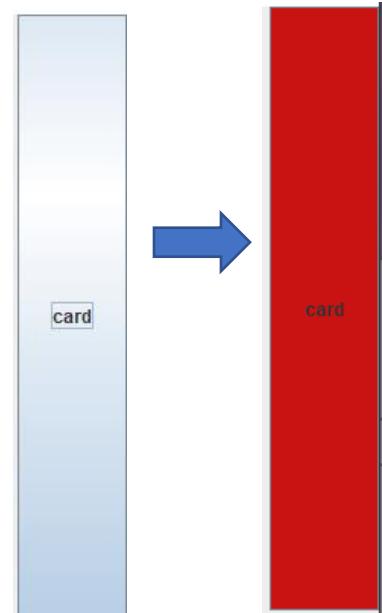
Below is when not clicking cancel button

```

if(event.getSource().equals(checkButtons[0])){//"card" button
    subPanel3.setVisible(false);
    checkCardStatus = true;
    checkButtons[0].setBackground(choiceRed);
}

```

Here is when the card button clicked will change the color to red. In another words means that the user insert card.



```

if(enteReturn == true ){//balance enquiry press enter to return
    if(event.getSource().equals(extraButtons[1]))
        enteReturn = false;
}

```

Here is once Enter button click will set enteReturn to false so that let the system know.

```

if(chooseAccount == true && cancelPressed == false)//account menu button
{
    if(event.getSource().equals(leftButtons[0])){//saving account button
        index = 1;
        chooseAccount = false;
    }
    else if(event.getSource().equals(rightButtons[0])){//cheque account button
        index = 2;
        chooseAccount = false;
    }
    else //other buttons
        index = -1;
}

if(chooseSaving == true)//saving account's button
{
    if(event.getSource().equals(leftButtons[0])){//balance enquiry button
        index = 1;
        chooseSaving = false;
    }
    else if(event.getSource().equals(leftButtons[1])){//withdraw button
        index = 2;
        chooseSaving = false;
    }
    else if(event.getSource().equals(leftButtons[2])){//transfer button
        index = 3;
        chooseSaving = false;
    }
    else //other buttons
        index = -1;
}

if(chooseCheque == true)//cheque account's button
{
    if(event.getSource().equals(leftButtons[0])){//balance enquiry button
        index = 1;
        chooseCheque = false;
    }
    else if(event.getSource().equals(leftButtons[1])){//withdraw button
        index = 2;
        chooseCheque = false;
    }
    else if(event.getSource().equals(leftButtons[2])){//transfer button
        index = 3;
        chooseCheque = false;
    }
    else //other buttons
        index = -1;
}

```

Account Menu

Saving account      Cheque account

Here is when user choosing the account then they will press the left or right button.

Saving Menu

View My Balance  
Withdraw Cash  
Transfer

Here is when user in saving account then they will press the left button to choose.

Cheque Menu

View My Balance  
Withdraw Cash  
Transfer

Here is when user in cheque account then they will press the left button to choose.

```

if(withdrawalconfirm == 1)//print advice's button
{
    if(event.getSource().equals(leftButtons[1])){//print advice
        withdrawalconfirm = 2;
    }
    else if(event.getSource().equals(rightButtons[1])){//not print advice
        withdrawalconfirm = 2;
    }
}

```

Your action is processed.

Please Select

Print Advice

Without Advice

Here is when user in success take money then they will press the button to choose.

```

if(takeCard == true)//take card button
{
    if(event.getSource().equals(checkButtons[0])){//card click
        checkButtons[0].setBackground(null);
        checkCardStatus = false;
        checkButtons[1].setBackground(choiceRed);
        takeCard = false;
    }
}

```

Step 1: Please take your card.

Here is when user in take the card then the card wil change the colour back.

```

if(takeMoney == true)//take money button
{
    if(event.getSource().equals(checkButtons[1])){//card click
        checkButtons[1].setBackground(null);
        takeMoney = false;
    }
}

```

Step 2: Please take the money.

Here is after user take the card then the money button will change to red at last money took the colour will change back the colour.

```

if(enterPassReturn == true) { //account number gui enter button
    for (int i = 0; i <= 11; i++) {
        if ( event.getSource().equals(extraButtons[2])) {
            passTextField.setText("");
            line1 = "";
        } else if ( event.getActionCommand().equals(names[i])) {
            line1 = line1.concat(event.getActionCommand());
            passTextField.setText(line1); // display line1 in jTextField
        } else {
            line1 = line1.concat("");
        }
    }

    if (enterPassReturn == true && event.getSource().equals(extraButtons[1])) {
        String input = passTextField.getText(); //the input account name
        accountNumberInput = Integer.parseInt(input);
        enterPassReturn = false;
    }
}

//pin
if(enterPINReturn == true) { //PIN gui enter button

    for (int i = 0; i <= 11; i++) { //pin gui
        if ( event.getSource().equals(extraButtons[2])) {
            passwordText.setText("");
            linePass = "";
        } else if (event.getActionCommand().equals(names[i])) {
            linePass = linePass.concat(event.getActionCommand());
            passwordText.setText(linePass); // display line1 in jTextField
        } else {
            linePass = linePass.concat("");
        }
    }

    if ( event.getSource().equals(extraButtons[1])) {
        pinInput = passwordText.getText(); //the input account name

        enterPINReturn = false;
        passwordText.setText("");
        linePass = "";
        passTextField.setText("");
        line1 = "";
    }
}

if(invalidConfirm == true && event.getSource().equals(extraButtons[1])){
    invalidConfirm = false;
    reset();
}

```



These two are to show the text field. The for loop is to check the number pad clicked then if “clear” clicked it will set text to nothing same as string. If the number pad buttons clicked the string will concatenation it otherwise clicked other buttons will concatenation nothing. And the below one’ for-loop is for setting the pin number text field. At last, “Enter” buttons clicked it will continue and if press ”cancel” it will set canceledNum = 1 which is for exit the atm system.

From the Balance Inquiry,

```
//set the border layout of the whole panel
BorderLayout layout = (BorderLayout)subPanel2.getLayout();
subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));

JPanel balancePanel = new JPanel(); //set the panel name
balancePanel.setLayout(new GridLayout(4, 1)); // set the panel layout

JLabel title = new JLabel("Balance Information"); //set the output
title.setFont(new Font("Serif", Font.PLAIN, 30)); // set the font
title.setHorizontalAlignment(JLabel.CENTER); //set the position
title.setVerticalAlignment(JLabel.CENTER); //set the position
JPanel accountPanelBlank1 = new JPanel(); //set panel name
accountPanelBlank1.setLayout(new BorderLayout()); //set back the panel
accountPanelBlank1.add(title); //add the panel using
accountPanelBlank1.setBackground(atmBlue); //add the panel color

JLabel first = new JLabel("Available balance: $" + availableBalance); //set the output
first.setFont(new Font("Serif", Font.PLAIN, 30)); // set the font
first.setHorizontalAlignment(JLabel.CENTER); //set the position
first.setVerticalAlignment(JLabel.CENTER); //set the position
JPanel accountPanel1 = new JPanel(); //set panel name
accountPanel1.setLayout(new BorderLayout()); //set back the panel
accountPanel1.setBackground(choiceBlue); //add the panel using
accountPanel1.add(first); //add the panel color

JLabel second = new JLabel("Total balance: $" + totalBalance); //set the output
second.setFont(new Font("Serif", Font.PLAIN, 30)); // set the font
second.setHorizontalAlignment(JLabel.CENTER); //set the position
second.setVerticalAlignment(JLabel.CENTER); //set the position

JPanel accountPanel2 = new JPanel(); //set panel name
accountPanel2.setLayout(new BorderLayout()); //set back the panel
accountPanel2.setBackground(choiceBlue); //add the panel color
accountPanel2.add(second); //add the panel using

JLabel messageLabel = new JLabel("Please press \"Enter\" to back."); //set the output
messageLabel.setFont(new Font("Serif", Font.PLAIN, 25)); // set the font
messageLabel.setHorizontalAlignment(JLabel.RIGHT); //set the position
messageLabel.setVerticalAlignment(JLabel.CENTER); //set the position

JPanel accountPanel3 = new JPanel(); //set panel name
accountPanel3.setLayout(new BorderLayout()); //set back the panel
accountPanel3.setBackground(atmBlue); //add the panel color
accountPanel3.add(messageLabel); //add the panel using

balancePanel.add(accountPanelBlank1); //add the panel using
balancePanel.add(accountPanel1); //add the panel using
balancePanel.add(accountPanel2); //add the panel using
balancePanel.add(accountPanel3); //add the panel using

subPanel2.add(balancePanel, BorderLayout.CENTER); //add the panel using
balancePanel.setVisible(true);
subPanel2.revalidate();
```

The GUI set required with different coding. Each set of JLabel is for setting a text and set the alignment and set font size and show it. For the whole coding is for showing. Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour at last put it in a panel which set to gridlayout to display.

From Withdrawal and Transfer both same method

<pre>public abstract int takevalue();</pre>	added an abstract in transaction class
<pre>public int takevalue(){     return canceledNum; }</pre>	Which it is use for cancelling to exit the atm system back to welcome.
<pre>public JLabel setLabelTitle(String name){     JLabel assignTitleLabel = new JLabel(); //set the panel name     assignTitleLabel.setText(name); // set the font     assignTitleLabel.setFont(new Font("name: "Serif", Font.PLAIN, size: 30)); // set the font     assignTitleLabel.setHorizontalTextPosition(JLabel.CENTER); //set the position     assignTitleLabel.setVerticalTextPosition(JLabel.CENTER); //set the position     return assignTitleLabel; }</pre>	This method is to set the label with the font serif ,font size 30 and set alignment to center.

In both class we add the set of buttons so that we can call the buttons to use.

<pre>BorderLayout layout = (BorderLayout)subPanel2.getLayout(); subPanel2.remove(layout.getLayoutComponent(BorderLayout.WEST)); subPanel2.remove(layout.getLayoutComponent(BorderLayout.EAST)); BorderLayout layout1 = (BorderLayout)subPanel.getLayout(); subPanel.remove(layout1.getLayoutComponent(BorderLayout.EAST)); subPanel.remove(layout1.getLayoutComponent(BorderLayout.CENTER));  leftButtons = new JButton[4]; // create array of JButtons leftButtonJPanel = new JPanel(); // set up panel leftButtonJPanel.setLayout(new GridLayout( rows: 5, cols: 1, hgap: 0, vgap: 10)); //set GridLayout JLabel blankLabel = new JLabel(); //set panel leftButtonJPanel.add(blankLabel); for (int count = 0; count &lt; 4; count++) {     leftButtons[count] = new JButton();     leftButtonJPanel.add(leftButtons[count]); // add button to JFrame } // end for //set left buttons leftButtons[0].setActionCommand("B0"); leftButtons[0].setText("&gt;&gt;"); leftButtons[1].setActionCommand("B1"); leftButtons[1].setText("&gt;&gt;"); leftButtons[2].setActionCommand("B2"); leftButtons[2].setText("&gt;&gt;"); leftButtons[3].setActionCommand("B3"); leftButtons[3].setText("&gt;&gt;");  rightButtons = new JButton[4]; // create array of JButtons rightButtonJPanel = new JPanel(); // set up panel rightButtonJPanel.setLayout(new GridLayout( rows: 5, cols: 1, hgap: 0, vgap: 10)); JLabel blankLabel1 = new JLabel(); rightButtonJPanel.add(blankLabel1); for (int count = 0; count &lt; 4; count++) {     rightButtons[count] = new JButton();     rightButtonJPanel.add(rightButtons[count]); // add button to JFrame } // end for //set right buttons rightButtons[0].setActionCommand("B4"); rightButtons[0].setText("&lt;&lt;"); rightButtons[1].setActionCommand("B5"); rightButtons[1].setText("&lt;&lt;"); rightButtons[2].setActionCommand("B6"); rightButtons[2].setText("&lt;&lt;"); rightButtons[3].setActionCommand("B7"); rightButtons[3].setText("&lt;&lt;");</pre>	Here we remove the old buttons first.
	<p>These are the left buttons.</p> 

These are the right buttons.



```

extraButtons = new JButton[3]; // create array of JButtons
extraButtonJPanel = new JPanel(); // set up panel
extraButtonJPanel.setLayout(new GridLayout( rows: 3, cols: 1, hgap: 0, vgap: 3));

for (int count = 0; count < 3; count++) {
    extraButtons[count] = new JButton();
    extraButtonJPanel.add(extraButtons[count]); // add button to JFrame
} // end for

//set three buttons
extraButtons[0].setActionCommand("Cancel");
extraButtons[0].setText("Cancel");
extraButtons[0].setBackground(Color.red);
extraButtons[1].setActionCommand("\n");
extraButtons[1].setText("Enter");
extraButtons[1].setBackground(Color.green);
extraButtons[2].setText("Clear");
extraButtons[2].setText("Clear");
extraButtons[2].setBackground(Color.yellow);

buttons = new JButton[names.length]; // create array of JButtons
buttonJPanel = new JPanel(); // set up panel
buttonJPanel.setLayout(new GridLayout( rows: 4, buttons.length, hgap: 3, vgap: 3));
for (int count = 0; count < names.length; count++) {
    buttons[count] = new JButton(names[count]);
    buttonJPanel.add(buttons[count]); // add button to JFrame
} // end for

```

These are the extra buttons.

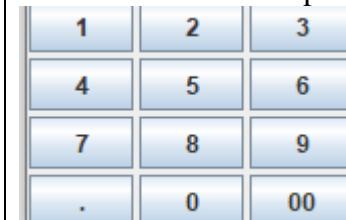


```

handlerT = new ButtonHandler();
subPanel2.add(leftButtonJPanel, BorderLayout.WEST); //add the panel with layout
subPanel2.add(rightButtonJPanel, BorderLayout.EAST); //add the panel with layout
subPanel.add(extraButtonJPanel, BorderLayout.EAST); //add the panel with layout
subPanel.add(buttonJPanel, BorderLayout.CENTER); //add the panel with layout
// register event handlers
for (int i = 0; i <= 11; i++) {
    buttons[i].addActionListener(handlerT);
}
for (int i = 0; i <= 3; i++) {
    leftButtons[i].addActionListener(handlerT);
    rightButtons[i].addActionListener(handlerT);
}
for (int i = 0; i <= 2; i++) {
    extraButtons[i].addActionListener(handlerT);
}

```

These are the number pad buttons



Here is to add the handler to each button then add back to the panels to display.

## From the Withdrawal,

```

public class Withdrawal extends Transaction
{
    private static final String[] names =
        { "1", "2", "3", "4", "5", "6", "7", "8", "9", ".", "0", "00" };
    Color atmBlue = new Color( 179, 178, 209 );
    Color choiceBlue = new Color( 63, 139, 166 );
    private JButton[] leftButtons; // array of buttons
    private JPanel leftButtonJPanel; // panel to hold buttons

    private JButton[] rightButtons; // array of buttons
    private JPanel rightButtonJPanel; // panel to hold buttons

    private JButton[] extraButtons; // array of buttons
    private JPanel extraButtonJPanel; // panel to hold buttons

    private JButton[] buttons; // array of buttons
    private JPanel buttonJPanel; // panel to hold buttons

    private String lineamount = ""; //for enter amount
    private JTextField enterTextField = new JTextField( columns, 50 );

    private int amount; // amount to withdraw
    private double check = -1; // amount to withdraw

    private Keypad keypad; // reference to keypad
    private CashDispenser cashDispenser; // reference to cash dispenser
    private JPanel subPanel2;
    private JPanel subPanel;
    private ButtonHandler handler;
    private boolean withdrawalReturn = false;
    private boolean checkerror = false;
    private boolean enterGUIReturn = false;

    private int edit = -1;
    private int canceledNum = 0;
}

```

```

private int displayMenuOfAmounts()
{
    int userChoice = 0; // local variable to store return value

    boolean checkCase = true;
    Screen screen = getScreen(); // get screen reference

    // array of amounts to correspond to menu numbers
    int amounts[] = { 0, 100, 500, 1000, 0 };

    // loop while no valid choice has been made
    while ( userChoice == 0 )
    {
        // display the menu with hkd
        screen.displayMessageLine( "\nWithdrawal Menu:" );
        screen.displayMessageLine( "1 - HKD100" );
        screen.displayMessageLine( "2 - HKD500" );
        screen.displayMessageLine( "3 - HKD1000" );
        screen.displayMessageLine( "4 - Enter the amounts" );
        screen.displayMessageLine( "5 - Cancel transaction" );
        screen.displayMessage( "\nChoose a withdrawal amount: " );

        withdrawalChooseGUI();
        //int input = keypad.getInput(); // get user input through keypad
        while( withdrawalReturn == true){
            try {
                Thread.sleep( millis, 100 );
            } catch (InterruptedException ex) {
            }
        }
        // determine how to proceed based on the input value
    }
}

```

Initializing the variable.

Here is the interface in the ATM system and here is withdrawalChooseGUI to call the GUI with this statement.

```

public void execute()
{
    boolean cashDispensed = false; // cash was not dispensed yet
    double availableBalance; // amount available for withdrawal
    int num1 = 0, num2 = 0, num3 = 0; // set num1 is HKD100 num2 HKD 500 num3 HKD 1000
    // get references to bank database and screen
    BankDatabase bankDatabase = getBankDatabase();
    Screen screen = getScreen();

    // loop until cash is dispensed or the user cancels
    do
    {
        // obtain a chosen withdrawal amount from the user
        amount = displayMenuOfAmounts();
        // check whether user enter a wrong amount
        if(amount == ERROR){//if error show error message
            errorMessage();
            while(checkerror == true){
                try {
                    Thread.sleep( millis: 1000 );
                } catch (InterruptedException ex) {
                }
            }
            //error message
            screen.displayMessageLine( "\nERROR transaction..." );
            screen.displayMessageLine( "\nOnly accept integer." );
            screen.displayMessageLine( "\nPlease try enter." );
        }

        // check whether the user has enough money in the account
        if ( amount <= availableBalance )
        {
            // check whether the cash dispenser has enough money
            if ( cashDispenser.isSufficientCashAvailable( amount ) )
            {...} // end if
            else // cash dispenser does not have enough cash
            {
                nomoneyatm();
                while (checkerror == true) {
                    try {
                        Thread.sleep( millis: 1000 );
                    } catch (InterruptedException ex) {
                    }
                }
                screen.displayMessageLine("//error message
                    "\nInsufficient cash available in the ATM." +
                    "\nPlease choose a smaller amount.");
            }
        } // end if
    }

    // update the account involved to reflect withdrawal
    bankDatabase.debit( getAccountNumber(), amount );

    cashDispenser.dispenseCash( amount ); // dispense cash
    cashDispensed = true; // cash was dispensed
    do{//calculate output how many piece(s) of $100,$500,$1000
        if(amount>=1000){ //if the input value larger or equal $1000
            amount-=1000;//the amount of input will minus $1000
            num3++; //the $1000 will add one
        }
        else if(amount>=500){ // if the input value larger or equal $500
            amount-=500;//the amount of input will minus $500
            num2++; //the $500 will add one
        }
        else if(amount >=100){ // if the input value larger or equal $100
            amount-=100;//the amount of input will minus $100
            num1++; //the $100 will add one
        }
    }while(amount!=0);
}

```

Here is the method to implement the check cases and setting error messages. Then, it uses the “try and catch” manners to add the insufficient case output. Besides, it uses the do while loop for changing the amount and num values.

```

screen.displayMessageLine("\nPlease take your cash now." );//display message of how many cash the user takes,
                                                        // if 0 number of that cash dollar, the piece will not show
if(num1>0)// Display the message of $100
    screen.displayMessageFd( message: "\nNumber of HKD 100: %d piece(s)", num1 );
if(num2>0)// Display the message of $500
    screen.displayMessageFd( message: "\nNumber of HKD 500: %d piece(s)", num2 );
if(num3>0)// Display the message of $1000
    screen.displayMessageFd( message: "\nNumber of HKD 1000: %d piece(s)", num3 );

{...}

} // end if

```

Here is to check the num numbers and set the display confirm page messages.

<pre> //set JPanel checkerror=true; BorderLayout layout = (BorderLayout)subPanel2.getLayout(); subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER)); JPanel numberPanel = new JPanel(); numberPanel.setLayout(new GridLayout( rows: 5, cols: 1)); numberPanel.setBackground(atmBlue);  JPanel accountPanel1 = new JPanel(); accountPanel1.setLayout(new BorderLayout()); accountPanel1.setBackground(atmBlue);  JPanel accountPanel3 = new JPanel(); accountPanel3.setLayout(new BorderLayout()); accountPanel3.setBackground(atmBlue);  JPanel accountPanel4 = new JPanel(); accountPanel4.setLayout(new BorderLayout()); accountPanel4.setBackground(atmBlue);  if(num1&gt;0)// Display the message of \$100 {     oneHun.setText("Number of HKD 100: " +num1+ " piece(s)");     oneHun.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));     oneHun.setHorizontalAlignment(JLabel.CENTER);     oneHun.setVerticalAlignment(JLabel.CENTER); }  numberPanel.add(accountPanel1);  if(num3&gt;0)// Display the message of \$1000 {     oneTh.setText("Number of HKD 1000: " +num3+ " piece(s)");     oneTh.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));     oneTh.setHorizontalAlignment(JLabel.CENTER);     oneTh.setVerticalAlignment(JLabel.CENTER); }  numberPanel.add(accountPanel4); </pre>	<pre> //set JLabel JLabel title = new JLabel(); title.setText("Number of Money"); title.setFont(new Font( name: "Serif", Font.PLAIN, size: 35)); title.setHorizontalAlignment(JLabel.CENTER); title.setVerticalAlignment(JLabel.CENTER); JPanel accountPanelBlank1 = new JPanel(); accountPanelBlank1.setLayout(new BorderLayout()); accountPanelBlank1.setBackground(atmBlue); numberPanel.add(accountPanelBlank1);  JLabel oneHun = new JLabel( text: ""); JLabel fiveHun = new JLabel( text: ""); JLabel oneTh = new JLabel( text: "");  Here is to set the variables </pre>
<pre> accountPanelBlank1.setLayout(new BorderLayout()); accountPanelBlank1.add(title); accountPanel1.setLayout(new BorderLayout()); accountPanel1.add(oneHun); accountPanel3.setLayout(new BorderLayout()); accountPanel3.add(fiveHun); accountPanel4.setLayout(new BorderLayout()); accountPanel4.add(oneTh); accountPanel5.setLayout(new BorderLayout()); accountPanel5.add(Label3); //add panel subPanel2.add(numberPanel, BorderLayout.CENTER); numberPanel.setVisible(true); subPanel2.revalidate(); return; }  } // end if </pre>	<pre> if(num2&gt;0)// Display the message of \$500 {     fiveHun.setText("Number of HKD 500: " +num2+ " piece(s)");     fiveHun.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));     fiveHun.setHorizontalAlignment(JLabel.CENTER);     fiveHun.setVerticalAlignment(JLabel.CENTER); } numberPanel.add(accountPanel3);  JLabel Label3 = new JLabel(); Label3.setText("Please press \"Enter\" to continue."); Label3.setFont(new Font( name: "Serif", Font.PLAIN, size: 30)); Label3.setHorizontalAlignment(JLabel.RIGHT); Label3.setVerticalAlignment(JLabel.CENTER); JPanel accountPanel5 = new JPanel(); accountPanel5.setLayout(new BorderLayout()); accountPanel5.setBackground(atmBlue); numberPanel.add(accountPanel5);  Here is the GUI displaying the page let the user confirm the amount of withdrawal. We use 3 JLabel to display the \$100,\$500 and \$1000 message. Then, add back the JLabels to panel. </pre>

<pre> else // not enough money available in user's account {     nomoneyacc(); //use method     while (checkerror == true) {         try {             Thread.sleep( millis: 1000 );         } catch (InterruptedException ex) {          }     }     screen.displayMessageLine( //error message         "\nInsufficient funds in your account." +         "\n\nPlease choose a smaller amount." ); } // end else } // end if //set JLabel public JLabel setLabelTitle(String name){     JLabel assignTitleLabel = new JLabel();     assignTitleLabel.setText(name);     assignTitleLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));     assignTitleLabel.setHorizontalTextPosition(JLabel.CENTER);     assignTitleLabel.setVerticalTextPosition(JLabel.CENTER);     return assignTitleLabel; } </pre>	<pre> else // user chose cancel menu option {     screen.displayMessageLine( "\nCanceling transaction..." );//cancel message     withCancel();     return; // return to main menu because user canceled } // end else } while ( !cashDispensed ); } // end method execute </pre>
--	--

Here is the case of outputting the error messages of cash dispenser does not have enough cash or there are not enough money available in user's account.

```

public void withdrawalChooseGUI() //withdrawal choose GUI
{
    withdrawalReturn = true;
    //declare layout
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    //declare JPanel
    JPanel withdrawalPanel = new JPanel();
    withdrawalPanel.setLayout(new GridBagLayout());
    withdrawalPanel.setBackground(atmBlue);
}

```

```

//declare JLabel
JLabel title = setLabelTitle("Select the amount to withdrawal");
JPanel accountPanelBlank1 = new JPanel();
accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(title);
//declare JLabel
JLabel blank = new JLabel( text: "" );
JPanel accountPanelBlank2 = new JPanel();
accountPanelBlank2.setLayout(new BorderLayout());
accountPanelBlank2.add(blank);
//declare JLabel
JLabel first = setLabelTitle("HKD $100");
JPanel accountPanel1 = new JPanel();
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(first);
//declare JLabel
JLabel second = setLabelTitle("HKD $500");
JPanel accountPanel2 = new JPanel();
accountPanel2.setLayout(new BorderLayout());
accountPanel2.add(second);
//declare JLabel
JLabel three = setLabelTitle("HKD $1000");
JPanel accountPanel3 = new JPanel();
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(three);

```

```

//declare JLabel
JLabel bottom = setLabelTitle("Or");
JPanel accountPanel4 = new JPanel();
accountPanel4.setLayout(new BorderLayout());
accountPanel4.add(bottom);
//declare JLabel
JLabel bottom1 = setLabelTitle("Press \'Enter\' continue to input the amount.");
JPanel accountPanel5 = new JPanel();
accountPanel5.setLayout(new BorderLayout());
accountPanel5.add(bottom1);
//set background color
accountPanelBlank1.setBackground(atmBlue);
accountPanelBlank2.setBackground(atmBlue);
accountPanel1.setBackground(choiceBlue);
accountPanel2.setBackground(choiceBlue);
accountPanel3.setBackground(choiceBlue);
accountPanel4.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);

```

The GUI set required with different coding. Here is setting JLabels. Each set of JLabel is setting a text and set the alignment, font size and background color. For the whole coding is for showing.

```

//set position
GridBagConstraints c0 = new GridBagConstraints();
c0.insets = new Insets( top: 0, left: 0, bottom: 1, right: 0);
c0.gridx = 0;
c0.gridy = 0;
c0.gridwidth = 2;
c0.gridheight = 1;
c0.weightx = 0.5;
c0.weighty = 0.5;
c0.fill = GridBagConstraints.BOTH;
c0.anchor = GridBagConstraints.NORTH;
withdrawalPanel.add(accountPanelBlank1, c0);
//set position
GridBagConstraints c2 = new GridBagConstraints();
c2.insets = new Insets( top: 3, left: 5, bottom: 8, right: 0);
c2.gridx = 1;
c2.gridy = 1;
c2.gridwidth = 1;
c2.gridheight = 1;
c2.weightx = 0.47;
c2.weighty = 0.47;
c2.fill = GridBagConstraints.BOTH;
c2.anchor = GridBagConstraints.EAST;
withdrawalPanel.add(accountPanel2, c2);

//set position
GridBagConstraints c4 = new GridBagConstraints();
c4.insets = new Insets( top: 0, left: 5, bottom: 8, right: 0);
c4.gridx = 1;
c4.gridy = 2;
c4.gridwidth = 1;
c4.gridheight = 1;
c4.weightx = 0.47;
c4.weighty = 0.47;
c4.fill = GridBagConstraints.BOTH;
c4.anchor = GridBagConstraints.SOUTH;
withdrawalPanel.add(accountPanelBlank2, c4);

//set position
GridBagConstraints c6 = new GridBagConstraints();
c6.gridx = 0;
c6.gridy = 5;
c6.gridwidth = 2;
c6.gridheight = 1;
c6.weightx = 0.5;
c6.weighty = 0.5;
c6.fill = GridBagConstraints.BOTH;
c6.anchor = GridBagConstraints.WEST;
withdrawalPanel.add(accountPanel5, c6);

//set position
subPanel2.add(withdrawalPanel, BorderLayout.CENTER);
withdrawalPanel.setVisible(true);
subPanel2.revalidate();
subPanel2.repaint();

}

```

After setting the JLabels, here are set the position and size. Besides, putting them into the panel with set layout to border layout. Therefore, it will fit in the right alignment, put it in a panel which set to grid layout to display, and set it to be visible eventually.

<pre> public void invalidMessage(){//invalid message     checkeror=true;     BorderLayout layout = (BorderLayout)subPanel2.getLayout();     subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));     JPanel invalidPanel = new JPanel();     invalidPanel.setLayout(new GridLayout( rows: 4, cols: 1));     invalidPanel.setBackground(atmBlue); } </pre>	
<pre> //declare JLabel JLabel invalidTitle = setLabelTitle("Withdrawal"); JPanel accountPanelBlank1 = new JPanel(); accountPanelBlank1.setLayout(new BorderLayout()); invalidPanel.add(accountPanelBlank1);  JLabel invalidLabel = setLabelTitle("Invalid selection."); JPanel accountPanel1 = new JPanel(); accountPanel1.setLayout(new BorderLayout()); invalidPanel.add(accountPanel1);  JLabel invalidLabel2 = setLabelTitle("Please try again."); JPanel accountPanel3 = new JPanel(); accountPanel3.setLayout(new BorderLayout()); invalidPanel.add(accountPanel3);  -----, //add accountPanelBlank1.setLayout(new BorderLayout()); accountPanelBlank1.add(invalidTitle); accountPanelBlank1.setBackground(atmBlue); accountPanel1.setLayout(new BorderLayout()); accountPanel1.add(invalidLabel); accountPanel1.setBackground(atmBlue); accountPanel3.setLayout(new BorderLayout()); accountPanel3.add(invalidLabel2); accountPanel3.setBackground(atmBlue); accountPanel5.setLayout(new BorderLayout()); accountPanel5.add(invalidLabel3); accountPanel5.setBackground(atmBlue);  subPanel2.add(invalidPanel, BorderLayout.CENTER); invalidPanel.setVisible(true); subPanel2.revalidate(); subPanel2.repaint(); </pre>	<pre> JLabel invalidLabel3 = new JLabel(); invalidLabel3.setText("Please press \"Enter\" to continue."); invalidLabel3.setFont(new Font( name: "Serif", Font.PLAIN, size: 30)); invalidLabel3.setHorizontalTextPosition(JLabel.RIGHT); invalidLabel3.setVerticalTextPosition(JLabel.CENTER); JPanel accountPanel5 = new JPanel(); accountPanel5.setLayout(new BorderLayout()); invalidPanel.add(accountPanel5); </pre> <p>Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.</p>

```

public void enterGUI(){//enterGUI method
    enterGUIReturn = true;

    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    //declare panel
    JPanel enterPanel = new JPanel();
    enterPanel.setLayout(new GridBagLayout());
    enterPanel.setBackground(atmBlue);

    //declare JLabel
    JLabel titleLabel = new JLabel( text: "Withdrawal");
    titleLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 25));
    titleLabel.setHorizontalTextPosition(JLabel.CENTER);
    titleLabel.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.setBackground(atmBlue);
    accountPanelBlank1.add(titleLabel);

    JLabel blankLabel = new JLabel( text: "");
    JPanel accountPanel3 = new JPanel();
    accountPanel3.setLayout(new BorderLayout());
    accountPanel3.add(blankLabel);
    accountPanel3.setBackground(atmBlue);
    //set position
    GridBagConstraints c0 = new GridBagConstraints();
    c0.insets = new Insets( top: 0, left: 0, bottom: 1, right: 0 );
    c0.gridx = 0;
    c0.gridy = 0;
    c0.gridwidth = 2;
    c0.gridheight = 1;
    c0.weightx = 0.5;
    c0.weighty = 0.5;
    c0.fill = GridBagConstraints.BOTH;
    c0.anchor = GridBagConstraints.NORTH;
    enterPanel.add(accountPanelBlank1, c0);

    GridBagConstraints c2 = new GridBagConstraints();
    c2.insets = new Insets( top: 3, left: 5, bottom: 8, right: 0 );
    c2.gridx = 0;
    c2.gridy = 2;
    c2.gridwidth = 2;
    c2.gridheight = 1;
    c2.weightx = 0.47;
    c2.weighty = 0.47;
    c2.fill = GridBagConstraints.NONE;
    c2.anchor = GridBagConstraints.WEST;
    enterPanel.add(accountPanel2, c2);

    JLabel titleLabel2 = new JLabel( text: "Enter the amount(HKD $)");
    titleLabel2.setFont(new Font( name: "Serif", Font.PLAIN, size: 25));
    titleLabel2.setHorizontalAlignment(JLabel.CENTER);
    titleLabel2.setVerticalAlignment(JLabel.CENTER);
    JPanel accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.setBackground(choiceBlue);
    accountPanel1.add(titleLabel2);

    // declaration of textArea for displaying output
    enterTextField.setEditable(false);
    enterTextField.setText(lineamount); // display line1 in textArea
    lineamount = "";
    enterTextField.setText("");
    //declare panel
    JPanel accountPanel2 = new JPanel();
    accountPanel2.setLayout(new BorderLayout());
    //add textfield
    accountPanel2.add(enterTextField);

    GridBagConstraints c1 = new GridBagConstraints();
    c1.insets = new Insets( top: 3, left: 0, bottom: 8, right: 5 );
    c1.gridx = 0;
    c1.gridy = 1;
    c1.gridwidth = 2;
    c1.gridheight = 1;
    c1.weightx = 0.47;
    c1.weighty = 0.47;
    c1.fill = GridBagConstraints.BOTH;
    c1.anchor = GridBagConstraints.WEST;
    enterPanel.add(accountPanel1, c1);

    GridBagConstraints c3 = new GridBagConstraints();
    c3.insets = new Insets( top: 0, left: 0, bottom: 8, right: 5 );
    c3.gridx = 0;
    c3.gridy = 3;
    c3.gridwidth = 2;
    c3.gridheight = 2;
    c3.weightx = 0.47;
    c3.weighty = 0.47;
    c3.fill = GridBagConstraints.BOTH;
    c3.anchor = GridBagConstraints.WEST;
    enterPanel.add(accountPanel3, c3);

    subPanel2.add(enterPanel, BorderLayout.CENTER);
    enterPanel.setVisible(true);
    subPanel2.revalidate();
    subPanel2.repaint();
}
}

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display. Since there are some button for the user to click, so require to use 4 GridBagConstraints to the choice button.

```

public void errorMessage(){
    checkeror=true;

    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout( rows: 4, cols: 1));
    errorPanel.setBackground(atmBlue);

    JLabel errortitle = new JLabel();
    errortitle.setText("Withdrawal");
    errortitle.setFont(new Font( name: "Serif", Font.PLAIN, size: 35));
    errortitle.setHorizontalTextPosition(JLabel.CENTER);
    errortitle.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.setBackground(atmBlue);
    errorPanel.add(accountPanelBlank1);

    JLabel errorLabel = new JLabel();
    errorLabel.setText("Error input, please input integer only.");
    errorLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel.setHorizontalTextPosition(JLabel.CENTER);
    errorLabel.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.setBackground(atmBlue);
    errorPanel.add(accountPanel1);

    JLabel errorLabel2 = new JLabel();
    errorLabel2.setText("Please press \"Enter\" to continue.");
    errorLabel2.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel2.setHorizontalTextPosition(JLabel.RIGHT);
    errorLabel2.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel3 = new JPanel();
    accountPanel3.setLayout(new BorderLayout());
    accountPanel3.setBackground(atmBlue);
    errorPanel.add(accountPanel3);
}

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(errortitle);
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(errorLabel);
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(errorLabel2);

subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();
}
}

```

```

public void wrongMessage(){
    checkeror=true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout( rows: 4, cols: 1));
    errorPanel.setBackground(atmBlue);

    JLabel errortitle = new JLabel();
    errortitle.setText("Withdrawal");
    errortitle.setFont(new Font( name: "Serif", Font.PLAIN, size: 35));
    errortitle.setHorizontalTextPosition(JLabel.CENTER);
    errortitle.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    errorPanel.add(accountPanelBlank1);
}
}

```

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

JLabel errorLabel = new JLabel();
errorLabel.setText("Wrong input");
errorLabel.setFont(new Font("Serif", Font.PLAIN, 30));
errorLabel.setHorizontalTextPosition(JLabel.CENTER);
errorLabel.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel1 = new JPanel();
accountPanel1.setLayout(new BorderLayout());
errorPanel.add(accountPanel1);

JLabel errorLabel2 = new JLabel();
errorLabel2.setText("Please enter the amount with multiples of HKD100.");
errorLabel2.setFont(new Font("Serif", Font.PLAIN, 30));
errorLabel2.setHorizontalTextPosition(JLabel.CENTER);
errorLabel2.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel3 = new JPanel();
accountPanel3.setLayout(new BorderLayout());
errorPanel.add(accountPanel3);

JLabel errorLabel3 = new JLabel();
errorLabel3.setText("Please press \"Enter\" to continue.");
errorLabel3.setFont(new Font("Serif", Font.PLAIN, 30));
errorLabel3.setHorizontalTextPosition(JLabel.RIGHT);
errorLabel3.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel5 = new JPanel();
accountPanel5.setLayout(new BorderLayout());
errorPanel.add(accountPanel5);

accountPanelBlank1.setBackground(atmBlue);
accountPanel1.setBackground(atmBlue);
accountPanel3.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);

accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(errorTitle);
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(errorLabel);
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(errorLabel2);
accountPanel5.setLayout(new BorderLayout());
accountPanel5.add(errorLabel3);

subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();
}
}

```

```

public void nomoneyatm(){
    checker=true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout(4, 1));
    errorPanel.setBackground(atmBlue);

    JLabel errorTitle = new JLabel();
    errorTitle.setText("Withdrawal");
    errorTitle.setFont(new Font("Serif", Font.PLAIN, 35));
    errorTitle.setHorizontalTextPosition(JLabel.CENTER);
    errorTitle.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    errorPanel.add(accountPanelBlank1);

    JLabel errorLabel = new JLabel();
    errorLabel.setText("Insufficient cash available in the ATM.");
    errorLabel.setFont(new Font("Serif", Font.PLAIN, 30));
    errorLabel.setHorizontalTextPosition(JLabel.CENTER);
    errorLabel.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    errorPanel.add(accountPanel1);
}

```

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

JLabel errorLabel2 = new JLabel();
errorLabel2.setText("Please choose a smaller amount.");
errorLabel2.setFont(new Font("Serif", Font.PLAIN, size: 30));
errorLabel2.setHorizontalTextPosition(JLabel.CENTER);
errorLabel2.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel3 = new JPanel();
accountPanel3.setLayout(new BorderLayout());
errorPanel.add(accountPanel3);

JLabel errorLabel3 = new JLabel();
errorLabel3.setText("Please press \"Enter\" to continue.");
errorLabel3.setFont(new Font("Serif", Font.PLAIN, size: 30));
errorLabel3.setHorizontalTextPosition(JLabel.RIGHT);
errorLabel3.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel5 = new JPanel();
accountPanel5.setLayout(new BorderLayout());
errorPanel.add(accountPanel5);

accountPanelBlank1.setBackground(atmBlue);
accountPanel1.setBackground(atmBlue);
accountPanel3.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);

accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(errorTitle);
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(errorLabel);
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(errorLabel2);
accountPanel5.setLayout(new BorderLayout());
accountPanel5.add(errorLabel3);

subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();

}

```

```

public void nomoneyacc(){
    checkerror=true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout(rows: 4, cols: 1));
    errorPanel.setBackground(atmBlue);
}

```

```

JLabel errorLabel2 = new JLabel();
errorLabel2.setText("Please choose a smaller amount.");
errorLabel2.setFont(new Font("Serif", Font.PLAIN, size: 30));
errorLabel2.setHorizontalTextPosition(JLabel.CENTER);
errorLabel2.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel3 = new JPanel();
accountPanel3.setLayout(new BorderLayout());
errorPanel.add(accountPanel3);

JLabel errorLabel3 = new JLabel();
errorLabel3.setText("Please press \"Enter\" to continue.");
errorLabel3.setFont(new Font("Serif", Font.PLAIN, size: 30));
errorLabel3.setHorizontalTextPosition(JLabel.RIGHT);
errorLabel3.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel5 = new JPanel();
accountPanel5.setLayout(new BorderLayout());
errorPanel.add(accountPanel5);

subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();
}

```

```

JLabel errorTitle = new JLabel();
errorTitle.setText("Withdrawal");
errorTitle.setFont(new Font("Serif", Font.PLAIN, size: 35));
errorTitle.setHorizontalTextPosition(JLabel.CENTER);
errorTitle.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanelBlank1 = new JPanel();
accountPanelBlank1.setLayout(new BorderLayout());
errorPanel.add(accountPanelBlank1);

JLabel errorLabel = new JLabel();
errorLabel.setText("Insufficient cash available in your account.");
errorLabel.setFont(new Font("Serif", Font.PLAIN, size: 30));
errorLabel.setHorizontalTextPosition(JLabel.CENTER);
errorLabel.setVerticalTextPosition(JLabel.CENTER);
 JPanel accountPanel1 = new JPanel();
accountPanel1.setLayout(new BorderLayout());
errorPanel.add(accountPanel1);

accountPanelBlank1.setBackground(atmBlue);
accountPanel1.setBackground(atmBlue);
accountPanel3.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);

accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(errorTitle);
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(errorLabel);
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(errorLabel2);
accountPanel5.setLayout(new BorderLayout());
accountPanel5.add(errorLabel3);

```

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

public void withCancel(){
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.setLayout(layout.getComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout( 4, 1));
    errorPanel.setBackground(atmBlue);

    JLabel errortitle = new JLabel();
    errortitle.setText("Withdrawal");
    errortitle.setFont(new Font( name: "Serif", Font.PLAIN, size: 35));
    errortitle.setHorizontalTextAlignment(JLabel.CENTER);
    errortitle.setVerticalTextAlignment(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    errorPanel.add(accountPanelBlank1);

    JLabel errorLabel = new JLabel();
    errorLabel.setText("The withdrawal has canceled.");
    errorLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel.setHorizontalTextAlignment(JLabel.CENTER);
    errorLabel.setVerticalTextAlignment(JLabel.CENTER);
    JPanel accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    errorPanel.add(accountPanel1);

    JLabel errorLabel2 = new JLabel();
    errorLabel2.setText("");
    errorLabel2.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel2.setHorizontalTextAlignment(JLabel.CENTER);
    errorLabel2.setVerticalTextAlignment(JLabel.CENTER);
    JPanel accountPanel3 = new JPanel();
    accountPanel3.setLayout(new BorderLayout());
    errorPanel.add(accountPanel3);

    JLabel errorLabel3 = new JLabel();
    errorLabel3.setText("Please press \"Enter\" to continue.");
    errorLabel3.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel3.setHorizontalTextAlignment(JLabel.RIGHT);
    errorLabel3.setVerticalTextAlignment(JLabel.CENTER);
    JPanel accountPanel5 = new JPanel();
    accountPanel5.setLayout(new BorderLayout());
    errorPanel.add(accountPanel5);

    accountPanelBlank1.setBackground(atmBlue);
    accountPanel1.setBackground(atmBlue);
    accountPanel3.setBackground(atmBlue);
    accountPanel5.setBackground(atmBlue);
}

accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(errortitle);
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(errorLabel);
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(errorLabel2);
accountPanel5.setLayout(new BorderLayout());
accountPanel5.add(errorLabel3);

subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();
}

```

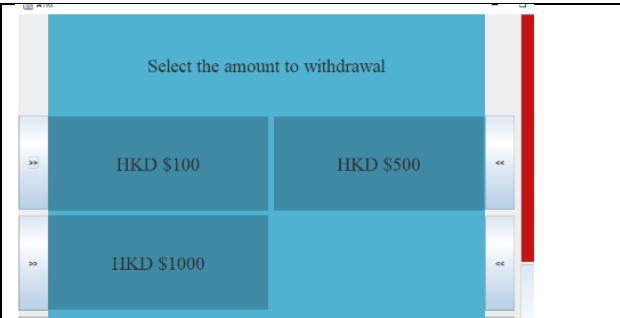
Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

Here is the ButtonHandler

```
public class ButtonHandler implements ActionListener {
    public void actionPerformed(ActionEvent e) {
```

```
        public class ButtonHandler implements ActionListener {
            public void actionPerformed(ActionEvent e) {

                String[] buttonsNames =
                    {"B0", "B1", "B2", "B3", "B4", "B5", "B6", "B7"};
                if (withdrawalReturn == true) { //page withdrawal
                    if (e.getSource().equals(extraButtons[1])) {
                        edit = 4; //set index will be 9 when enter press
                    } else if (e.getActionCommand().equals("B4")) { //B4 ie invalid selection
                        edit = 2;
                    } else if (e.getActionCommand().equals("B0")) {
                        edit = 1;
                    } else if (e.getActionCommand().equals("B1")) {
                        edit = 3;
                    } else edit = -1;
                    withdrawalReturn = false;
                }
            }
        }
    }
}
```

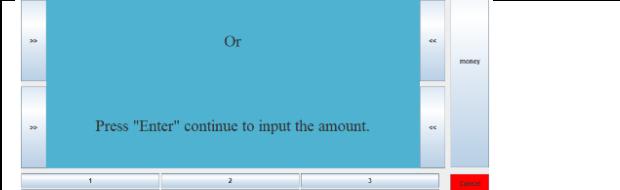


Here is setting the withdrawal button with withdrawal different kinds of money options.

```
        if (checkerror == true) { //page invalid selection
            if (e.getSource().equals(extraButtons[1])) {

                checkerror = false;

            }
        }
    }
}
```



Here is the setting the enter button before get in the GUI for inputting the amount.

```
for(int i=0;i<=11;i++){//enter gui
    if (e.getSource().equals(extraButtons[2])) {
        enterTextField.setText("");
        lineamount = "";
    }
    else if(e.getActionCommand().equals(names[i])) {
        lineamount=lineamount.concat( e.getActionCommand());
        enterTextField.setText(lineamount); // display line1 in textArea
    }
    else{
        lineamount=lineamount.concat("");
    }
}
```

```
if(EnterGUIReturn == true && e.getSource().equals(extraButtons[1])){
    String input = enterTextField.getText(); //the input
    System.out.println(input);
    check = Double.parseDouble(input); //for return the value back to withdrawal
    System.out.println(check);
    enterGUIReturn = false;
}
}
}

} // end class Withdrawal
```

Here is set the keypad of the GUI. The for loop is set for the number pad and the decimal for user's input, which pursue the real ATM system.

Also, user clicked then if "clear" clicked it will set text to nothing same as string. If the number pad buttons clicked the string will concatenation it otherwise clicked other buttons will concatenation nothing. At last, once "Enter" buttons clicked it will continue and if press "cancel" which is for exit the atm system.

## From the Transfer,

```
public class Transfer extends Transaction
{
    private static final String[] names =
        { "1", "2", "3", "4", "5", "6", "7", "8", "9", ".", "0", "00" };
    Color atmBlue = new Color(  c 79,  g 178,  b 209);
    Color choiceBlue = new Color(  c 63,  g 139,  b 166);
    private JButton[] leftButtons; // array of buttons
    private JPanel leftButtonJPanel; // panel to hold buttons

    private JButton[] rightButtons; // array of buttons
    private JPanel rightButtonJPanel; // panel to hold buttons

    private JButton[] extraButtons; // array of buttons
    private JPanel extraButtonJPanel; // panel to hold buttons

    private JButton[] buttons; // array of buttons
    private JPanel buttonJPanel; // panel to hold buttons

    private String line1 = "";//for enter amount
    private String line2 = "";//for enter amount

    private JTextField transferTextfield2 =new JTextField( columns, 50);//set for text
    private JTextField transferTextfield = new JTextField( columns, 50);//set for text

    private JPanel subPanel2;//set for panel
    private JPanel subPanel1;//set for panel
    private ButtonHandler handler1;//set for handler
    private boolean transRecretun= false;//set for boolean
    private boolean transerror = false;//set for boolean
    private boolean transGUIReturn = false;//set for boolean
    private boolean transGUIReturn1 = false;//set for boolean
    private boolean confirm = false;//set for boolean
    private boolean transRecretunchoose = false;//set for boolean

    //set for the variables
    private int edit = -1;
    private int choice = -1;
    private int canceledNum = 0;
    private int receiverAc =-1;
    private double amount; // amount to transfer
    private double check = -1; // amount to transfer
    private final Keypad keypad; // reference to keypad

    private final static int CANCELED = 0; // constant for cancel option
```

Initializing the variable.

```

//gui method
public void transInvalidMessage(){
    transerror=true;//set boolean
    BorderLayout layout = (BorderLayout)subPanel2.getLayout(); //set layout
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER)); //set the position
    JPanel invalidPanel = new JPanel(); //set the panel name
    invalidPanel.setLayout(new GridLayout(4, 1)); //set the panel row
    invalidPanel.setBackground(atmBlue); //set the panel color

    JLabel invalidTitle = setLabelTitle("Transaction"); //set the output
    JPanel accountPanelBlank1 = new JPanel(); //set the panel name
    accountPanelBlank1.setLayout(new BorderLayout()); //set the panel
    invalidPanel.add(accountPanelBlank1); //set the panel

    JLabel invalidLabel = setLabelTitle("Invalid selection."); //set the output
    JPanel accountPanel1 = new JPanel(); //set the panel name
    accountPanel1.setLayout(new BorderLayout()); //set the panel
    invalidPanel.add(accountPanel1); //set the panel

    JLabel invalidLabel2 = setLabelTitle("Please try again."); //set the output
    JPanel accountPanel3 = new JPanel(); //set the panel
    accountPanel3.setLayout(new BorderLayout()); //set the panel
    invalidPanel.add(accountPanel3); //add the panel

    JLabel invalidLabel3 = new JLabel(); //set the panel
    invalidLabel3.setText("Please press \"Enter\" to continue."); //set the output
    invalidLabel3.setFont(new Font("Serif", Font.PLAIN, 30)); //set the font
    invalidLabel3.setHorizontalAlignment(JLabel.RIGHT); //set the position
    invalidLabel3.setVerticalAlignment(JLabel.CENTER); //set the position
    JPanel accountPanel5 = new JPanel(); //set back the panel
    accountPanel5.setLayout(new BorderLayout()); //add the panel using
    invalidPanel.add(accountPanel5); //add the panel using

    accountPanelBlank1.setLayout(new BorderLayout()); //add the panel using
    accountPanelBlank1.add(invalidTitle); //add the panel
    accountPanelBlank1.setBackground(atmBlue); //add the panel color
    accountPanel1.setLayout(new BorderLayout()); //add the panel using
    accountPanel1.add(invalidLabel); //add the panel
    accountPanel1.setBackground(atmBlue); //add the panel color
    accountPanel3.setLayout(new BorderLayout()); //add the panel using
    accountPanel3.add(invalidLabel2); //add the panel
    accountPanel3.setBackground(atmBlue); //add the panel color
    accountPanel5.setLayout(new BorderLayout()); //add the panel using
    accountPanel5.add(invalidLabel3); //add the panel
    accountPanel5.setBackground(atmBlue); //add the panel color

    //add the panel using
    subPanel2.add(invalidPanel, BorderLayout.CENTER);
    invalidPanel.setVisible(true);
    subPanel2.revalidate();
    subPanel2.repaint();
}

```

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

do{
    transferChoose(); //call the gui
    while(transRecreturnchoose == true){
        try {
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {

        }
    }
    x = edit;
    System.out.println(x);
    // x = keypad.getInput();

    screen.displayMessage("\nInValid input. Please try again.");
    if(x != 1 && x != 2) {
        transInvalidMessage(); //call the gui
        while (transerror == true) {
            try {
                Thread.sleep( millis: 100 );
            } catch (InterruptedException ex) {
            }
        }
    }
}

}while(x != 1 && x != 2);

```

```

//gui method
public void receivevnum(){
    transerror=true;//set boolean
    BorderLayout layout = (BorderLayout)subPanel2.setLayout(); //set layout
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER)); //set the position
    JPanel invalidPanel = new JPanel(); //set the panel name
    invalidPanel.setLayout(new GridLayout( rows: 4, cols: 1)); //set the panel row
    invalidPanel.setBackground(atmBlue); //set the panel color

    JLabel invalidLabel = setLabelTitle("Receive number cannot be your account."); //set the output
    JPanel accountPanel1 = new JPanel(); //set panel name
    accountPanel1.setLayout(new BorderLayout()); //set panel name
    invalidPanel.add(accountPanel1); //add the panel using

    JLabel invalidLabel2 = setLabelTitle("Please press \"Enter\" to return."); //set the output
    JPanel accountPanel3 = new JPanel(); //set panel name
    accountPanel3.setLayout(new BorderLayout()); //set panel name
    invalidPanel.add(accountPanel3); //add the panel using

    accountPanel1.setLayout(new BorderLayout()); //add the panel using
    accountPanel1.add(invalidLabel); //add the panel using
    accountPanel1.setBackground(atmBlue); //add the panel using
    accountPanel3.setLayout(new BorderLayout()); //add the panel using
    accountPanel3.add(invalidLabel2); //add the panel using
    accountPanel3.setBackground(atmBlue); //add the panel using

    //add the panel using
    subPanel2.add(invalidPanel, BorderLayout.CENTER);
    invalidPanel.setVisible(true);
    subPanel2.revalidate();
    subPanel2.repaint();
}

```

Here is how to call the previous gui method and have a while loop waiting the button hander.

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

if(TransferAccountNumber == getAccountNumber()){
    receivenum(); //call the gui
    while(transerror == true) {
        try {
            Thread.sleep( millis: 100);
        } catch (InterruptedException ex) {

        }
    }
    screen.displayMessage( "\nReceiver's account cannot be your account." );
}

```

Here is how to call the previous gui method and have a while loop waiting the button hander.

```

public void transInvalid(){
    transerror=true;//set boolean
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();//set layout
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));//set the position
    JPanel invalidPanel = new JPanel();//set the panel name
    invalidPanel.setLayout(new GridLayout( rows: 4, cols: 1));//set the panel row
    invalidPanel.setBackground(atmBlue);//set the panel color

    JLabel invalidLabel = setLabelTitle("Invalid account number.");//set the output
    JPanel accountPanel1 = new JPanel();// set the font
    accountPanel1.setLayout(new BorderLayout());//add the panel using
    invalidPanel.add(accountPanel1);//add the panel using

    JLabel invalidLabel2 = setLabelTitle("Please press \'Enter\' return to menu.");//set the output
    JPanel accountPanel3 = new JPanel();// set the font
    accountPanel3.setLayout(new BorderLayout());//add the panel using
    invalidPanel.add(accountPanel3);//add the panel using

    accountPanel1.setLayout(new BorderLayout());//add the panel using
    accountPanel1.add(invalidLabel);//add the panel using
    accountPanel1.setBackground(atmBlue);//add the panel color
    accountPanel3.setLayout(new BorderLayout());//add the panel using
    accountPanel3.add(invalidLabel2);//add the panel using
    accountPanel3.setBackground(atmBlue);//add the panel color

    //add the panel using
    subPanel2.add(invalidPanel, BorderLayout.CENTER);
    invalidPanel.setVisible(true);
    subPanel2.revalidate();
    subPanel2.repaint();
}

```

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, moreover here have a textfield displaying. At last put it in a panel which set to gridbaglayout to display the c0 c1 c2 c3 are to set the row column width height.

```

//each label size
GridBagConstraints c1 = new GridBagConstraints();
c1.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0);
c1.gridx = 0;
c1.gridy = 1;
c1.gridwidth = 4;
c1.gridheight = 1;
c1.weightx = 0.47;
c1.weighty = 0.47;
c1.fill = GridBagConstraints.BOTH;
c1.anchor = GridBagConstraints.WEST;
transferenterReceivePanel.add(accountPanel1, c1);
//each label size
GridBagConstraints c2 = new GridBagConstraints();
c2.insets = new Insets( top: 0, left: 0, bottom: 0, right: 5);
c2.gridx = 0;
c2.gridy = 2;
c2.gridwidth = 1;
c2.gridheight = 1;
c2.weightx = 0.50;
c2.weighty = 0.50;
c2.fill = GridBagConstraints.HORIZONTAL;
c2.anchor = GridBagConstraints.WEST;
transferenterReceivePanel.add(accountPanel2, c2);
//each label size

```

```

transferTextField2.setEditable(false);
line2 = "";
transferTextField2.setText(line2); // display line1 in textField

GridBagConstraints c3 = new GridBagConstraints();
c3.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0);
c3.gridx = 1;
c3.gridy = 2;
c3.gridwidth = 3;
c3.gridheight = 1;
c3.weightx = 0.50;
c3.weighty = 0.50;
c3.fill = GridBagConstraints.HORIZONTAL;
c3.anchor = GridBagConstraints.CENTER;
transferenterReceivePanel.add(transfertextfield2, c3);
//each label size

//add panel
subPanel2.add(transferenterReceivePanel, BorderLayout.CENTER);
transferenterReceivePanel.setVisible(true);
subPanel2.revalidate();
subPanel2.repaint();

```

```

transferReceive(); //call the gui
while(transGUIReturn== true){
    try {
        Thread.sleep( millis: 100);
    } catch (InterruptedException ex) {

    }
}
//TransferAccountNumber = keypad.getInput();
TransferAccountNumber = receiverAc;

```

Here is how to call the previous gui method and have a while loop waiting the button hander.

```

public void transferAmount(){
    //set layout
    transGUIReturn1= true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel userPanel = new JPanel();
    userPanel.setLayout(new GridBagLayout());
    userPanel.setBackground(atmBlue);
    JPanel accountPanelBlank1 = new JPanel();
    JPanel accountPanel1 = new JPanel();
    JPanel accountPanel2 = new JPanel();
    transferTextField = new JTextField();

    //set color
    accountPanelBlank1.setBackground(atmBlue);
    accountPanel1.setBackground(atmBlue);
    accountPanel2.setBackground(choiceBlue);

    //declare JLabel
    JLabel messageLabel1 = new JLabel();
    messageLabel1.setText("Please enter the transfer amount");
    messageLabel1.setFont(new Font( "Serif", Font.PLAIN, 15));
    messageLabel1.setHorizontalTextAlignment(JLabel.CENTER);
    messageLabel1.setVerticalTextAlignment(JLabel.CENTER);
    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.add(messageLabel1);

    //declare JLabel
    JLabel messageLabel2 = new JLabel();
    messageLabel2.setText("After input the amount then press \"Enter\"");
    messageLabel2.setFont(new Font( "Serif", Font.PLAIN, 15));
    messageLabel2.setHorizontalTextAlignment(JLabel.CENTER);
    messageLabel2.setVerticalTextAlignment(JLabel.CENTER);
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(messageLabel2);

    //declare JLabel
    JLabel userLabel = new JLabel();
    userLabel.setText("transfer amount");
    userLabel.setFont(new Font( "Serif", Font.PLAIN, 15));
    userLabel.setHorizontalTextAlignment(JLabel.RIGHT);
    userLabel.setVerticalTextAlignment(JLabel.CENTER);
    accountPanel2.setLayout(new BorderLayout());
    accountPanel2.add(userLabel);

    //declare JLabel
    GridBagConstraints c0 = new GridBagConstraints();
    c0.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0);
    c0.gridx = 0;
    c0.gridy = 0;
    c0.gridwidth = 4;
    c0.gridheight = 1;
    c0.weightx = 0.47;
    c0.weighty = 0.47;
    c0.fill = GridBagConstraints.BOTH;
    c0.anchor = GridBagConstraints.WEST;
    userPanel.add(accountPanelBlank1, c0);
}
//declare JLabel
GridBagConstraints c1 = new GridBagConstraints();
c1.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0);
c1.gridx = 0;
c1.gridy = 1;
c1.gridwidth = 4;
c1.gridheight = 1;
c1.weightx = 0.47;
c1.weighty = 0.47;
c1.fill = GridBagConstraints.BOTH;
c1.anchor = GridBagConstraints.WEST;
userPanel.add(accountPanel1, c1);

//declare JLabel
GridBagConstraints c2 = new GridBagConstraints();
c2.insets = new Insets( top: 0, left: 0, bottom: 0, right: 5);
c2.gridx = 0;
c2.gridy = 2;
c2.gridwidth = 1;
c2.gridheight = 1;
c2.weightx = 0.50;
c2.weighty = 0.50;
c2.fill = GridBagConstraints.HORIZONTAL;
c2.anchor = GridBagConstraints.WEST;
userPanel.add(accountPanel2, c2);

transferTextField.setEditable(false);
line1 = "";
transferTextField.setText(line1); // display line1 in textField

//declare JLabel
GridBagConstraints c3 = new GridBagConstraints();
c3.insets = new Insets( top: 0, left: 0, bottom: 0, right: 0);
c3.gridx = 1;
c3.gridy = 2;
c3.gridwidth = 3;
c3.gridheight = 1;
c3.weightx = 0.50;
c3.weighty = 0.50;
c3.fill = GridBagConstraints.HORIZONTAL;
c3.anchor = GridBagConstraints.CENTER;
userPanel.add(transferTextField, c3);
//declare JLabel
accountPanelBlank1.setBackground(atmBlue);
accountPanel1.setBackground(choiceBlue);
accountPanel2.setBackground(choiceBlue);

subPanel2.add(userPanel, BorderLayout.CENTER);
userPanel.setVisible(true);
subPanel2.revalidate();
}
}

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, moreover there have a textfield displaying, at last put it in a panel which set to gridbaglayout to display the c0 c1 c2 c3 are to set the row column width height.

```

transferAmount(); //call gui
while(transGUIReturn1== true){
    try {
        Thread.sleep( millis: 100);
    } catch (InterruptedException ex) {
    }
}

```

Here is how to call the previous gui method and have a while loop waiting the button hander.

```

public void transCancel(){
    //set layout
    transerror=true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout( 4, 1));
    errorPanel.setBackground(atmBlue);

    //declare JLabel
    JLabel errortitle = new JLabel();
    errortitle.setText("Transaction");
    errortitle.setFont(new Font( "Serif", Font.PLAIN, 35));
    errortitle.setHorizontalTextPosition(JLabel.CENTER);
    errortitle.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    errorPanel.add(accountPanelBlank1);

    //declare JLabel
    JLabel errorLabel = new JLabel();
    errorLabel.setText("The ATM has canceled your transaction.");
    errorLabel.setFont(new Font( "Serif", Font.PLAIN, 30));
    errorLabel.setHorizontalAlignment(JLabel.CENTER);
    errorLabel.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    errorPanel.add(accountPanel1);

    //declare JLabel
    JLabel errorLabel2 = new JLabel();
    errorLabel2.setText("");
    errorLabel2.setFont(new Font( "Serif", Font.PLAIN, 30));
    errorLabel2.setHorizontalTextPosition(JLabel.CENTER);
    errorLabel2.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel3 = new JPanel();
    accountPanel3.setLayout(new BorderLayout());
    errorPanel.add(accountPanel3);

    //declare JLabel
    JLabel errorLabel3 = new JLabel();
    errorLabel3.setText("Please press \nEnter\ to continue.");
    errorLabel3.setFont(new Font( "Serif", Font.PLAIN, 30));
    errorLabel3.setHorizontalTextPosition(JLabel.RIGHT);
    errorLabel3.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel5 = new JPanel();
    accountPanel5.setLayout(new BorderLayout());
    errorPanel.add(accountPanel5);

    //add JLabel color
    accountPanelBlank1.setBackground(atmBlue);
    accountPanel1.setBackground(atmBlue);
    accountPanel3.setBackground(atmBlue);
    accountPanel5.setBackground(atmBlue);
}

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

//add panel
accountPanel.add(accountPanelBlank1);
accountPanel.add(accountPanelBlank2);
accountPanel.add(accountPanel1);
accountPanel.add(accountPanel2);
accountPanel.add(accountPanel3);
accountPanel.add(accountPanel4);
accountPanel.add(accountPanel5);
accountPanel.add(accountPanel6);
accountPanel.add(accountPanel7);
accountPanel.add(accountPanel8);

//add panel color
accountPanelBlank1.setBackground(atmBlue);
accountPanelBlank2.setBackground(atmBlue);
accountPanel.setBackground(atmBlue);
accountPanel1.setBackground(choiceBlue);
accountPanel2.setBackground(choiceBlue);
accountPanel3.setBackground(atmBlue);
accountPanel4.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);
accountPanel6.setBackground(atmBlue);
accountPanel7.setBackground(atmBlue);
accountPanel8.setBackground(atmBlue);

subPanel2.add(accountPanel, BorderLayout.CENTER);
accountPanel.setVisible(true);
subPanel2.revalidate();
subPanel2.repaint();

screen.displayMessageLine( "\nCanceling transaction..." );
transCancel(); //call the gui

```

```

public void transNomoneyacc(){
    //set layout
    transerror=true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout( rows: 4, cols: 1));
    errorPanel.setBackground(atmBlue);

    //declare JLabel
    JLabel errortitle = new JLabel();
    errortitle.setText("Transaction");
    errortitle.setFont(new Font( name: "Serif", Font.PLAIN, size: 35));
    errortitle.setHorizontalTextPosition(JLabel.CENTER);
    errortitle.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    errorPanel.add(accountPanelBlank1);

    //declare JLabel
    JLabel errorLabel = new JLabel();
    errorLabel.setText("Insufficient cash available in your account.");
    errorLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel.setHorizontalTextPosition(JLabel.CENTER);
    errorLabel.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    errorPanel.add(accountPanel1);

    //declare JLabel
    JLabel errorLabel2 = new JLabel();
    errorLabel2.setText("Please choose a smaller amount.");
    errorLabel2.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel2.setHorizontalTextPosition(JLabel.CENTER);
    errorLabel2.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel3 = new JPanel();
    accountPanel3.setLayout(new BorderLayout());
    errorPanel.add(accountPanel3);

    //declare JLabel
    JLabel errorLabel3 = new JLabel();
    errorLabel3.setText("Please press \\"Enter\\" to continue.");
    errorLabel3.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
    errorLabel3.setHorizontalTextPosition(JLabel.RIGHT);
    errorLabel3.setVerticalTextPosition(JLabel.CENTER);
    JPanel accountPanel5 = new JPanel();
    accountPanel5.setLayout(new BorderLayout());
    errorPanel.add(accountPanel5);

    //add panel color
    accountPanelBlank1.setBackground(atmBlue);
    accountPanel1.setBackground(atmBlue);
    accountPanel3.setBackground(atmBlue);
    accountPanel5.setBackground(atmBlue);

    if (input > availableBalance) {
        screen.displayMessageLine("\nInsufficient amount available in your account.\n");
        transNomoneyacc(); //call gui
        while(transerror== true){
            try {
                Thread.sleep( millis: 100);
            } catch (InterruptedException ex) {
            }
        }
    }
}

```

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

//add JLabel
accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(errortitle);
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(errorLabel);
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(errorLabel2);
accountPanel5.setLayout(new BorderLayout());
accountPanel5.add(errorLabel3);

//add JPanel
subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();
}

```

Here is how to call the previous gui method and have a while loop waiting the button handler.

```

public void transInvaenagain(){
    //set layout
    transerror=true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel errorPanel = new JPanel();
    errorPanel.setLayout(new GridLayout(4, 1));
    errorPanel.setBackground(atmBlue);

    //declare JLabel
    JLabel errortitle = new JLabel();
    errortitle.setText("Transaction");
    errortitle.setFont(new Font("Serif", Font.PLAIN, 35));
    errortitle.setHorizontalAlignment(JLabel.CENTER);
    errortitle.setVerticalAlignment(JLabel.CENTER);
    JPanel accountPanelBlank1 = new JPanel();
    accountPanelBlank1.setLayout(new BorderLayout());
    errorPanel.add(accountPanelBlank1);

    //declare JLabel
    JLabel errorLabel = new JLabel();
    errorLabel.setText("Invalid transfer number. ");
    errorLabel.setFont(new Font("Serif", Font.PLAIN, 30));
    errorLabel.setHorizontalAlignment(JLabel.CENTER);
    errorLabel.setVerticalAlignment(JLabel.CENTER);
    JPanel accountPanel1 = new JPanel();
    accountPanel1.setLayout(new BorderLayout());
    errorPanel.add(accountPanel1);

    //declare JLabel
    JLabel errorLabel2 = new JLabel();
    errorLabel2.setText("Please \"Enter\" to continue");
    errorLabel2.setFont(new Font("Serif", Font.PLAIN, 30));
    errorLabel2.setHorizontalAlignment(JLabel.RIGHT);
    errorLabel2.setVerticalAlignment(JLabel.CENTER);
    JPanel accountPanel3 = new JPanel();
    accountPanel3.setLayout(new BorderLayout());
    errorPanel.add(accountPanel3);

    //add JLabel color
    accountPanelBlank1.setBackground(atmBlue);
    accountPanel1.setBackground(atmBlue);
    accountPanel3.setBackground(atmBlue);

    //add JLabel
    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.add(errortitle);
    accountPanel1.setLayout(new BorderLayout());
    accountPanel1.add(errorLabel);
    accountPanel3.setLayout(new BorderLayout());
    accountPanel3.add(errorLabel2);

    else {
        screen.displayMessage("\nInvalid transfer number. Please enter again");
        transInvaenagain(); //call gui
        while(transerror== true){
            try {
                Thread.sleep(100);
            } catch (InterruptedException ex) {
            }
        }
    }
}

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

//add JLabel
subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();
}

```

Here is how to call the previous gui method and have a while loop waiting the button hander.

```

public void transOnly(){
    //set layout
    confirm = true;
    BorderLayout layout = (BorderLayout)subPanel2.getLayout();
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
    JPanel userPanel = new JPanel();
    userPanel.setLayout(new GridBagLayout());
    userPanel.setBackground(atmBlue);

    //declare JPanel
    JPanel accountPanelBlank1 = new JPanel();
    JPanel accountPanelBlank2 = new JPanel();
    JPanel accountPanel1 = new JPanel();
    JPanel accountPanel2 = new JPanel();
    JPanel accountPanel3 = new JPanel();
    JPanel accountPanel4 = new JPanel();
    JPanel accountPanel5 = new JPanel();
    JPanel accountPanel6 = new JPanel();
    JPanel accountPanel7 = new JPanel();
    JPanel accountPanel8 = new JPanel();
    JPanel accountPanel = new JPanel();
    accountPanel.setLayout(new GridLayout( 5, cols, 2, hgap, 10, vgap, 10));

    //declare JLabel
    JLabel accountLabel = new JLabel();
    accountLabel.setText(" Only accept ");
    accountLabel.setFont(new Font( name, "Serif", Font.PLAIN, size: 35));
    accountLabel.setHorizontalTextAlignment(JLabel.RIGHT);
    accountLabel.setVerticalTextAlignment(JLabel.CENTER);
    accountPanelBlank1.setLayout(new BorderLayout());
    accountPanelBlank1.add(accountLabel);

    //declare JPanel
    JLabel accountLabel1 = new JLabel();
    accountLabel1.setText("Confirm or Cancel");
    accountLabel1.setFont(new Font( name, "Serif", Font.PLAIN, size: 35));
    accountLabel1.setHorizontalTextAlignment(JLabel.LEFT);
    accountLabel1.setVerticalTextAlignment(JLabel.CENTER);
    accountPanelBlank2.setLayout(new BorderLayout());
    accountPanelBlank2.add(accountLabel1);

    //declare JPanel
    JLabel savingLabel = new JLabel();
    savingLabel.setText("Confirm");
    savingLabel.setFont(new Font( name, "Serif", Font.PLAIN, size: 30));
    savingLabel.setHorizontalTextAlignment(JLabel.CENTER);
    savingLabel.setVerticalTextAlignment(JLabel.CENTER);
    accountPanel.setLayout(new BorderLayout());
    accountPanel.add(savingLabel);
}

```

Here is setting three JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

//declare JPanel
JLabel chequeLabel = new JLabel();
chequeLabel.setText("Cancel");
chequeLabel.setFont(new Font( name, "Serif", Font.PLAIN, size: 30));
chequeLabel.setHorizontalTextAlignment(JLabel.CENTER);
chequeLabel.setVerticalTextAlignment(JLabel.CENTER);
accountPanel2.setLayout(new BorderLayout());
accountPanel2.add(chequeLabel);

//add JPanel
accountPanel.add(accountPanelBlank1);
accountPanel.add(accountPanelBlank2);
accountPanel.add(accountPanel1);
accountPanel.add(accountPanel2);
accountPanel.add(accountPanel3);
accountPanel.add(accountPanel4);
accountPanel.add(accountPanel5);
accountPanel.add(accountPanel6);
accountPanel.add(accountPanel7);
accountPanel.add(accountPanel8);

//add JPanel color
accountPanelBlank1.setBackground(atmBlue);
accountPanelBlank2.setBackground(atmBlue);
accountPanel.setBackground(atmBlue);
accountPanel1.setBackground(choiceBlue);
accountPanel2.setBackground(choiceBlue);
accountPanel3.setBackground(atmBlue);
accountPanel4.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);
accountPanel6.setBackground(atmBlue);
accountPanel7.setBackground(atmBlue);
accountPanel8.setBackground(atmBlue);

//add Panel
subPanel2.add(accountPanel, BorderLayout.CENTER);
accountPanel.setVisible(true);
subPanel2.revalidate();
subPanel2.repaint();

```

```

public void transInvalid(){
    transerror=true;//set boolean
    BorderLayout layout = (BorderLayout)subPanel2.getLayout(); //set layout
    subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER)); //set the position
    JPanel invalidPanel = new JPanel(); //set the panel name
    invalidPanel.setLayout(new GridLayout( rows: 4, cols: 1)); //set the panel row
    invalidPanel.setBackground(atmBlue); //set the panel color

    JLabel invalidLabel = setLabelTitle("Invalid account number."); //set the output
    JPanel accountPanel1 = new JPanel(); // set the font
    accountPanel1.setLayout(new BorderLayout()); //add the panel using
    invalidPanel.add(accountPanel1); //add the panel using

    JLabel invalidLabel2 = setLabelTitle("Please press \"Enter\" return to menu."); //set the output
    JPanel accountPanel3 = new JPanel(); // set the font
    accountPanel3.setLayout(new BorderLayout()); //add the panel using
    invalidPanel.add(accountPanel3); //add the panel using

    accountPanel1.setLayout(new BorderLayout()); //add the panel using
    accountPanel1.add(invalidLabel); //add the panel using
    accountPanel1.setBackground(atmBlue); //add the panel color
    accountPanel3.setLayout(new BorderLayout()); //add the panel using
    accountPanel3.add(invalidLabel2); //add the panel using
    accountPanel3.setBackground(atmBlue); //add the panel color

    //add the panel using
    subPanel2.add(invalidPanel, BorderLayout.CENTER);
    invalidPanel.setVisible(true);
    subPanel2.revalidate();
    subPanel2.repaint();
}

```

```

while(TransferReceived != 0 && TransferReceived != 1){

    screen.displayMessageLine( "\nOnly accept 0 or 1. Please try again: " );
    transOnly(); //call the gui
    while(confirm == true) {
        try {
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {

        }
    }
    //TransferReceived = keypad.getInput();
    TransferReceived = choice;
}

```

Here is how to call the previous gui method and have a while loop waiting the button hander.

Here is setting two JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

transInvlid(); //call the gui
while (transerror == true) {
    try {
        Thread.sleep( millis: 100);
    } catch (InterruptedException ex) {

    }
}

```

Here is how to call the previous gui method and have a while loop waiting the button hander.

```

public void transferAccept(){
//set layout
transerror=true;
BorderLayout layout = (BorderLayout)subPanel2.getLayout();
subPanel2.remove(layout.getLayoutComponent(BorderLayout.CENTER));
JPanel errorPanel = new JPanel();
errorPanel.setLayout(new GridLayout( rows: 4, cols: 1));
errorPanel.setBackground(atmBlue);

//declare JLabel
JLabel errortitle = new JLabel();
errortitle.setText("Transaction");
errortitle.setFont(new Font( name: "Serif", Font.PLAIN, size: 35));
errortitle.setHorizontalTextPosition(JLabel.CENTER);
errortitle.setVerticalTextPosition(JLabel.CENTER);
JPanel accountPanelBlank1 = new JPanel();
accountPanelBlank1.setLayout(new BorderLayout());
errorPanel.add(accountPanelBlank1);

//declare JLabel
JLabel errorLabel = new JLabel();
errorLabel.setText("Your transaction is successful.");
errorLabel.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
errorLabel.setHorizontalTextPosition(JLabel.CENTER);
errorLabel.setVerticalTextPosition(JLabel.CENTER);
JPanel accountPanel1 = new JPanel();
accountPanel1.setLayout(new BorderLayout());
errorPanel.add(accountPanel1);

//declare JLabel
JLabel errorLabel2 = new JLabel();
errorLabel2.setText("");
errorLabel2.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
errorLabel2.setHorizontalTextPosition(JLabel.CENTER);
errorLabel2.setVerticalTextPosition(JLabel.CENTER);
JPanel accountPanel3 = new JPanel();
accountPanel3.setLayout(new BorderLayout());
errorPanel.add(accountPanel3);

//declare JLabel
JLabel errorLabel3 = new JLabel();
errorLabel3.setText("Please press \"Enter\" to continue.");
errorLabel3.setFont(new Font( name: "Serif", Font.PLAIN, size: 30));
errorLabel3.setHorizontalTextPosition(JLabel.RIGHT);
errorLabel3.setVerticalTextPosition(JLabel.CENTER);
JPanel accountPanel5 = new JPanel();
accountPanel5.setLayout(new BorderLayout());
errorPanel.add(accountPanel5);

//add JLabel color
accountPanelBlank1.setBackground(atmBlue);
accountPanel1.setBackground(atmBlue);
accountPanel3.setBackground(atmBlue);
accountPanel5.setBackground(atmBlue);
}

```

Here is setting four JLabel and put it in the panel and the panel set layout to borderlayout so it will display in the right alignment also set the colour, at last put it in a panel which set to gridlayout to display.

```

//add JLabel
accountPanelBlank1.setLayout(new BorderLayout());
accountPanelBlank1.add(errortitle);
accountPanel1.setLayout(new BorderLayout());
accountPanel1.add(errorLabel);
accountPanel3.setLayout(new BorderLayout());
accountPanel3.add(errorLabel2);
accountPanel5.setLayout(new BorderLayout());
accountPanel5.add(errorLabel3);

//add JPanel
subPanel2.add(errorPanel, BorderLayout.CENTER);
errorPanel.setVisible(true);
subPanel2.revalidate();
}

```

```

if ( TransferReceived == 1)
{
    screen.displayMessageLine( "\nYour transaction has been accepted." );
    // credit account to reflect the transfer
    bankDatabase.debit( getAccountNumber(), amount );
    bankDatabase.debit( TransferAccountNumber, -amount );

    transferAccept(); //call the gui
}

// end if

```

Here is how to call the previous gui method.

```

public class ButtonHandler implements ActionListener {
    public void actionPerformed(ActionEvent e) {

```

Here is the ButtonHandler.

```

//TransferAccountNumber = keypad.getInput();
TransferAccountNumber = receiverAc;
if(TransferAccountNumber != -1){
    screen.displayMessage( "\n1. Receiver's Saving account" );
    screen.displayMessage( "\n2. Receiver's Cheque account\n" );
}
else
    break;
do{
    transferChoose(); //call the gui
    while(transRecreturnchoose == true){
        try {
            Thread.sleep( millis: 100 );
        } catch (InterruptedException ex) {

        }
    }
    x = edit;
    System.out.println(x);
    // x = keypad.getInput();

```

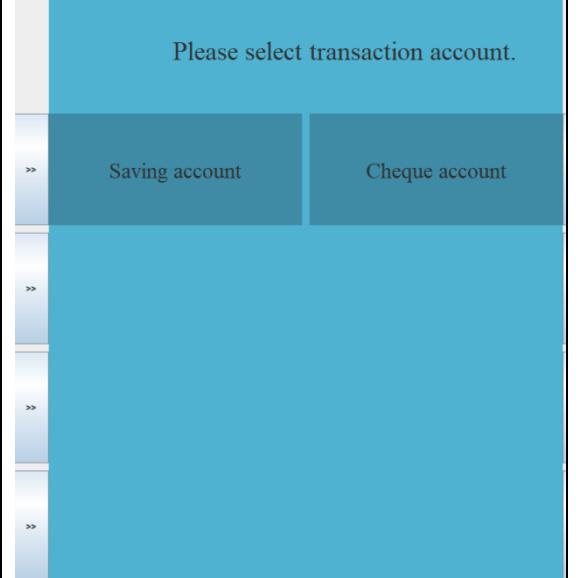
Here we use a while loop to wait the buttons clicked.

```

if (transRecreturnchoose == true) { //choose account
    if (e.getActionCommand().equals("B4")){
        edit = 2;
    } else if (e.getActionCommand().equals("B0")){
        edit = 1;
    }
    else edit = -1;
    transRecreturnchoose = false;
} //button for side

```

Here is means when enter right hand side GUI transRecreturnchoose become true so that run when the buttons clicked will run this line and when the buttons pressed the handler will set the edit then set back to false.



```

//button setting condition
if(transerror == true && e.getSource().equals(extraButtons[1])){
    transerror = false;
}
//button setting condition
if(transRe{return == true && e.getSource().equals(extraButtons[1])){
    transRe{return = false;
}

```

These two are use to wait user to press enter like right hand side.

```

//keypad setting condition
for(int i=0;i<=11;i++){//enter amount
    if (e.getSource().equals(extraButtons[2])) {
        transferTextfield2.setText("");
        line2 = "";// display line2 in textArea
    }
    else if(e.getActionCommand().equals(names[i])){
        line2=line2.concat( e.getActionCommand());
        transferTextfield2.setText(line2); // display line2 in textArea
    }
    else{
        line2=line2.concat("");
    }
}//button setting condition
if(transGUIGetReturn == true && e.getSource().equals(extraButtons[1])){
    String input = transferTextfield2.getText();//the input
    receiverAc = Integer.parseInt(input); //for return the value back to withdrawal
    transGUIGetReturn = false;
}

else if (transGUIGetReturn == true && e.getSource().equals(extraButtons[0])) { //cancel
    canceledNum = 1;
    receiverAc = -1; //for return the value back to withdrawal
    transGUIGetReturn = false;
}
//button setting condition
for(int i=0;i<=11;i++){//enter amount
    if (e.getSource().equals(extraButtons[2])) {
        transferTextfield.setText("");
        line1 = "";
    }
    else if(e.getActionCommand().equals(names[i])){
        line1 = line1.concat( e.getActionCommand());
        transferTextfield.setText(line1); // display line1 in textArea
    }
    else{
        line1=line1.concat("");
    }
}
if(transGUIGetReturn1 == true && e.getSource().equals(extraButtons[1])){
    String input = transferTextfield.getText();//the input
    check = Double.parseDouble(input); //for return the value back to withdrawal
    transGUIGetReturn1 = false;
}

else if (transGUIGetReturn1 == true && e.getSource().equals(extraButtons[0])) { //cancel
    canceledNum = 1;
    check = 0; //for return the value back to withdrawal
    transGUIGetReturn1 = false;
}

```

These two are to show the text field in the right-hand side. The for loop is to check the number pad clicked then if “clear” clicked it will set text to nothing same as string. If the number pad buttons clicked the string will concatenation it otherwise clicked other buttons will concatenation nothing. At last, once “Enter” buttons clicked it will continue and if press ”cancel” it will set canceledNum = 1 which is for exit the atm system.

Receive number cannot be your account.

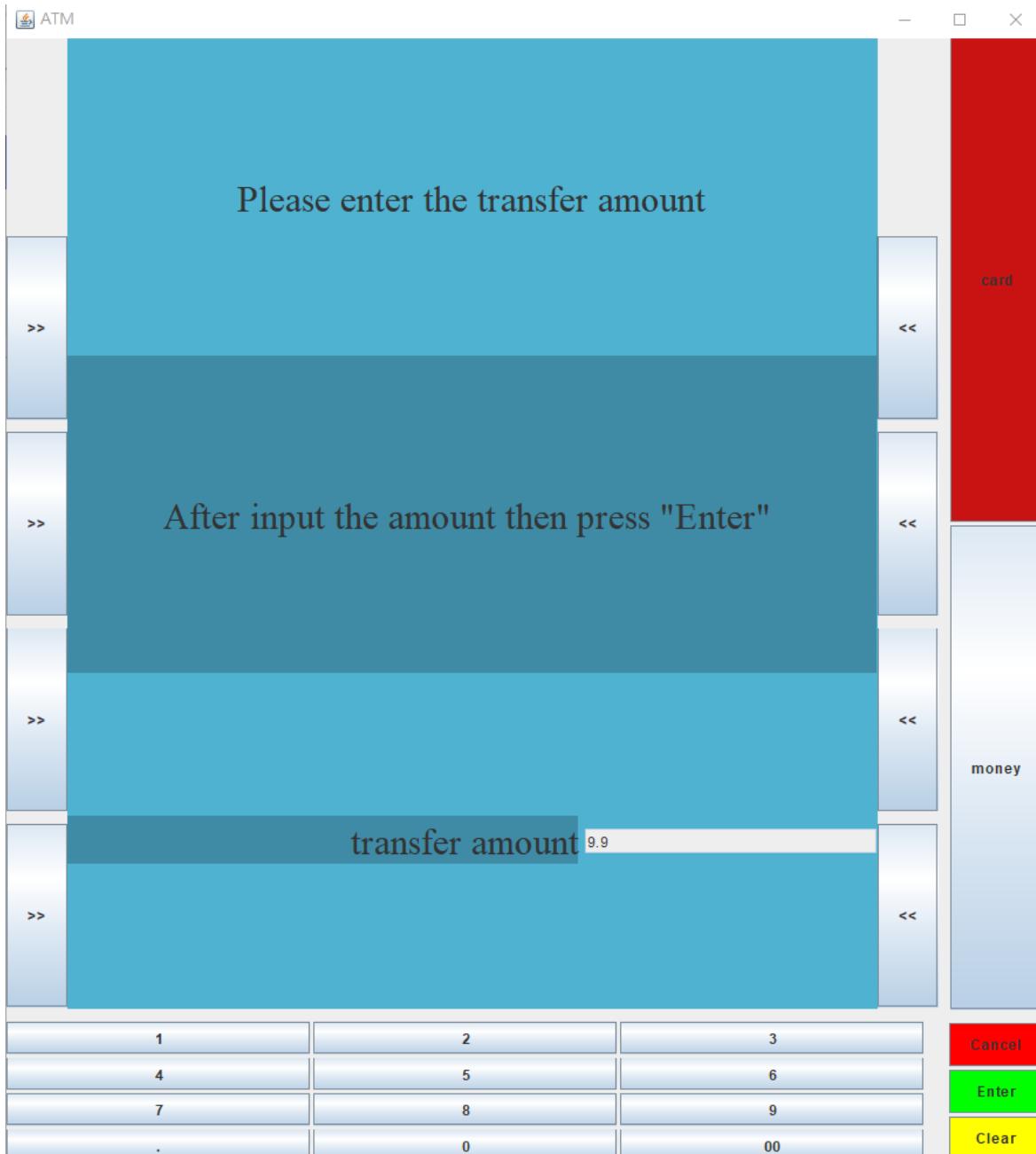
Please press "Enter" to return.

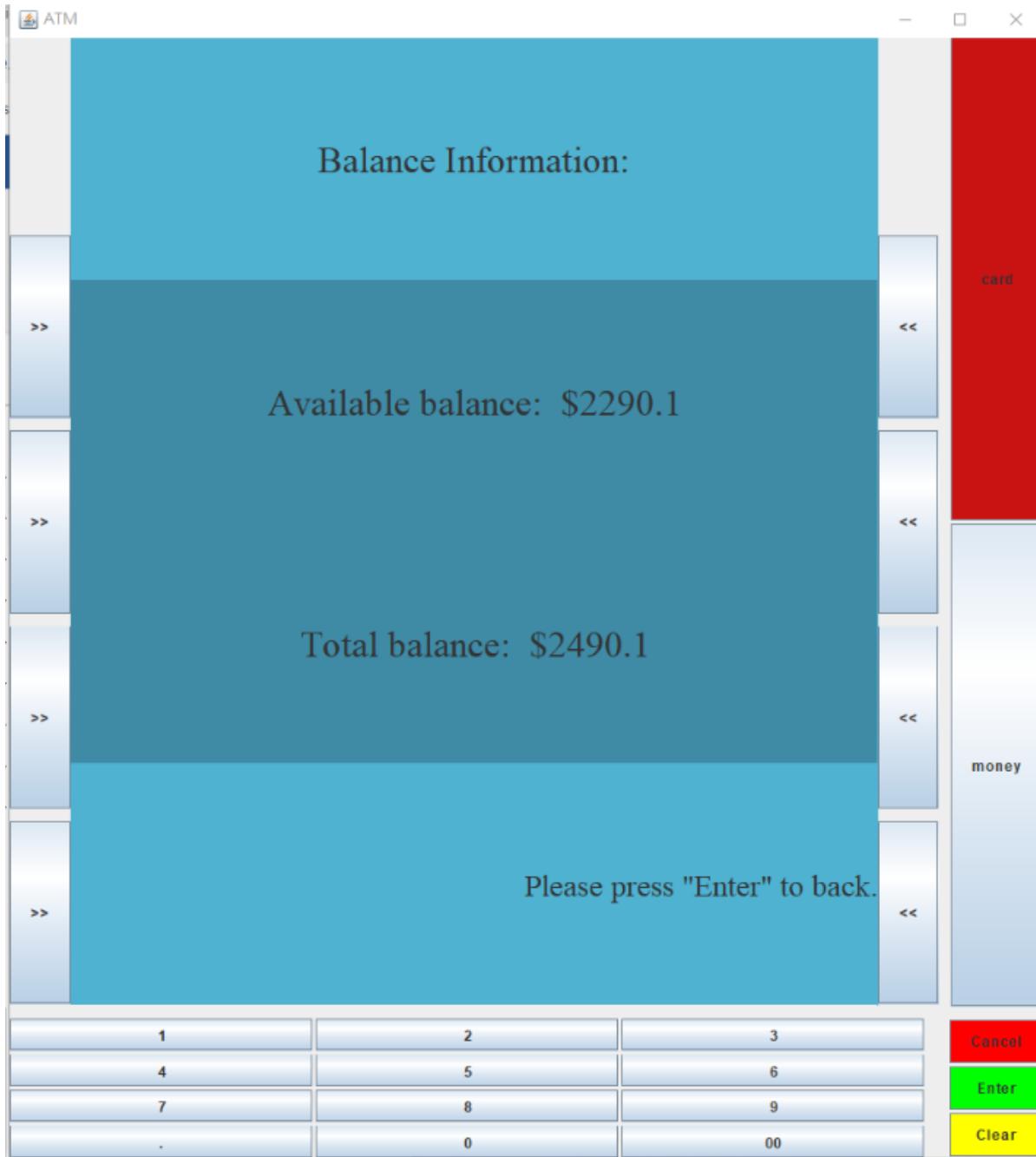
Account number

transfer amount

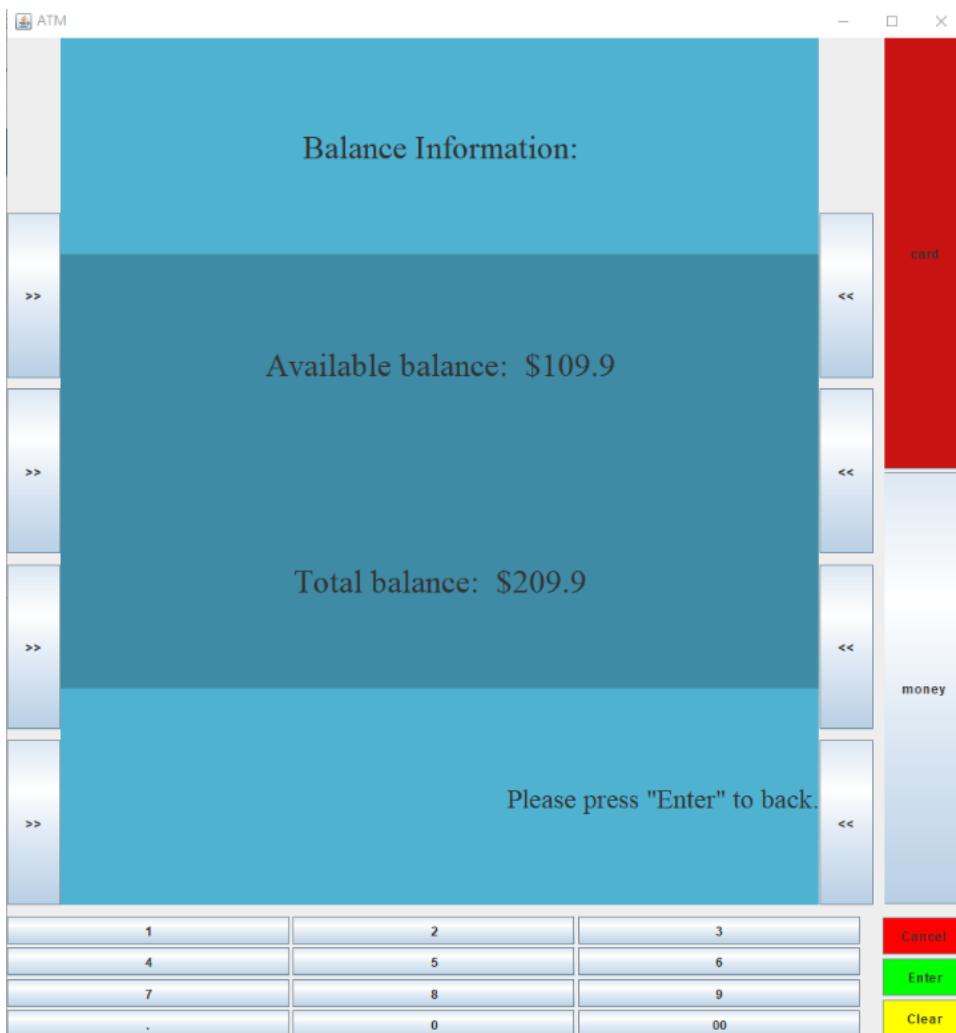
# Test cases

Test case 1: Transfer \$9.9





Saving Account of 12345 account.



Cheque Account of 12345 account, which increase \$9.9.

The balance information of 12345's saving account.

Account 12345001 (saving account)

$$\text{Available balance} = 2300 - 9.9 = 2290.1$$

$$\text{Total balance} = 2500 - 9.9 = 2490.1$$

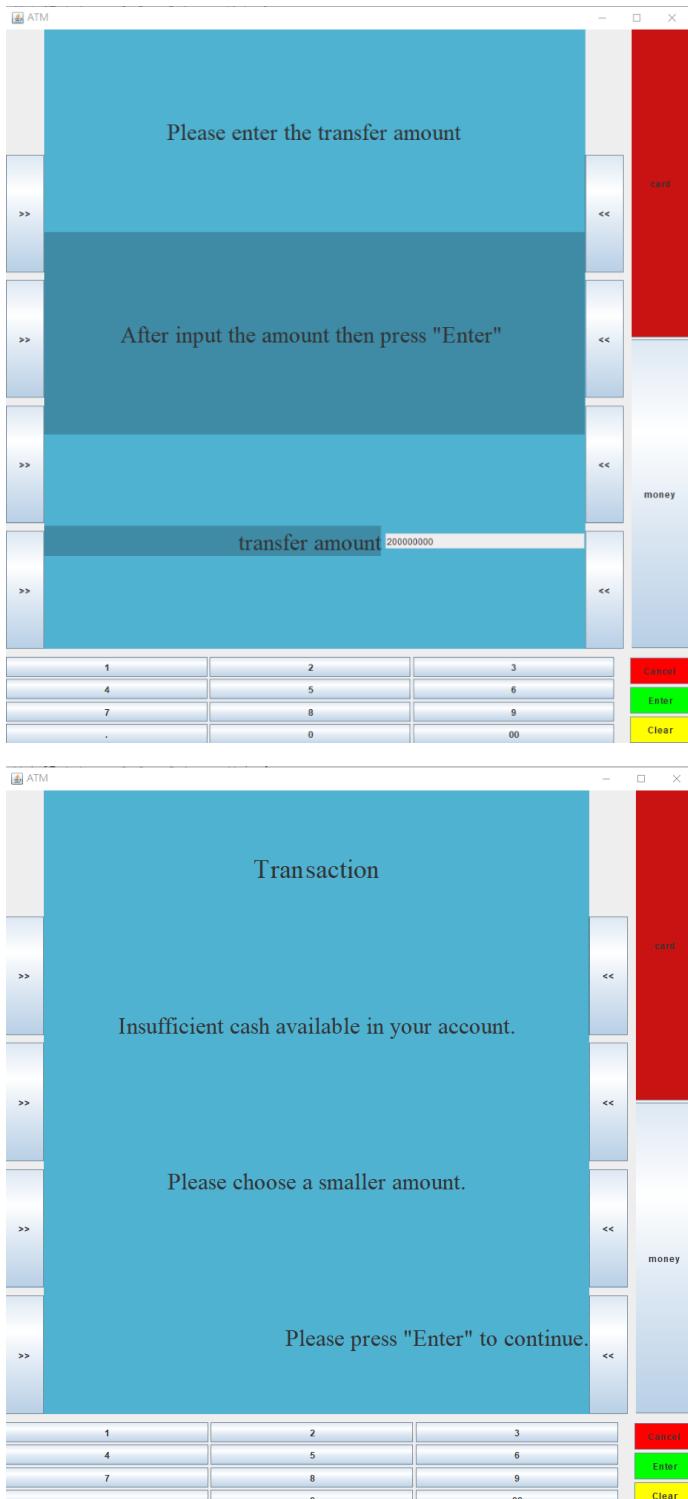
Account 12345002 (cheque account)

$$\text{Available balance} = 100 + 9.9 = 109.9$$

$$\text{Total balance} = 200 + 9.9 = 209.9$$

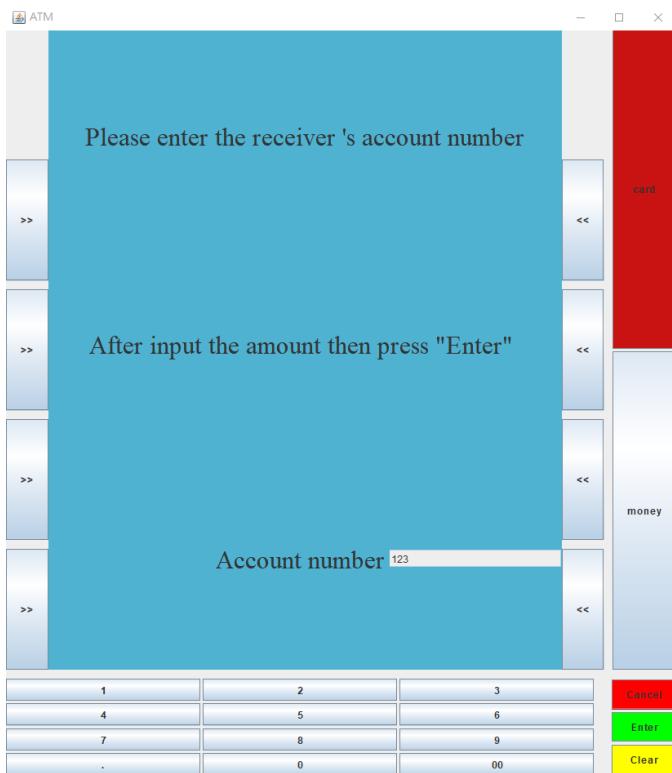
Thus, it is successful to transfer balance from 12345001 saving account to 98765001 saving account.

## Test case 2: Transfer amount > the available balance.

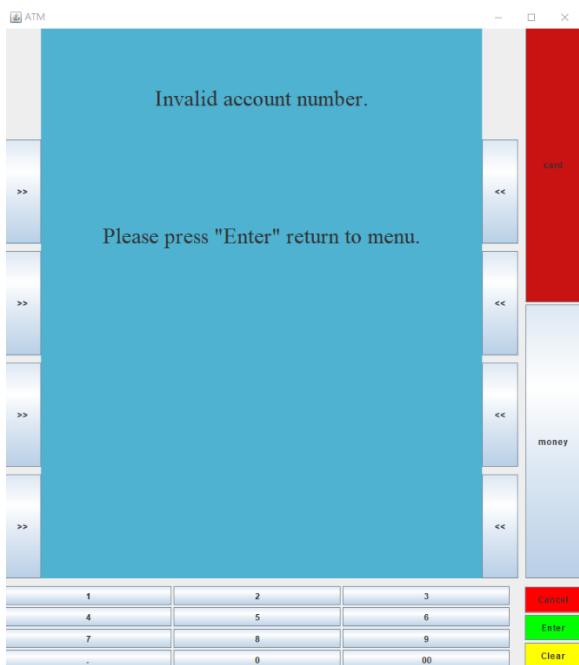


If the user input a amount larger than the available balance in user' account. This message will be shown.

### Test case 3: Transfer fund to an invalid bank account

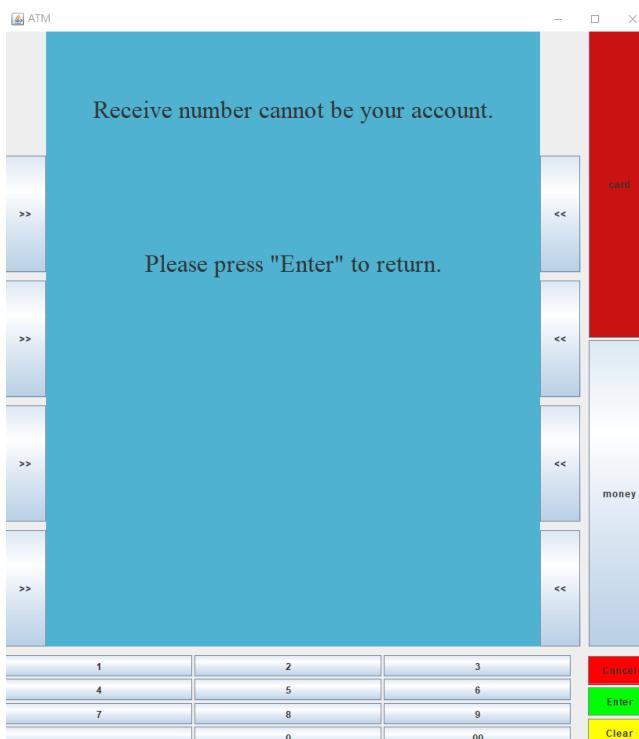
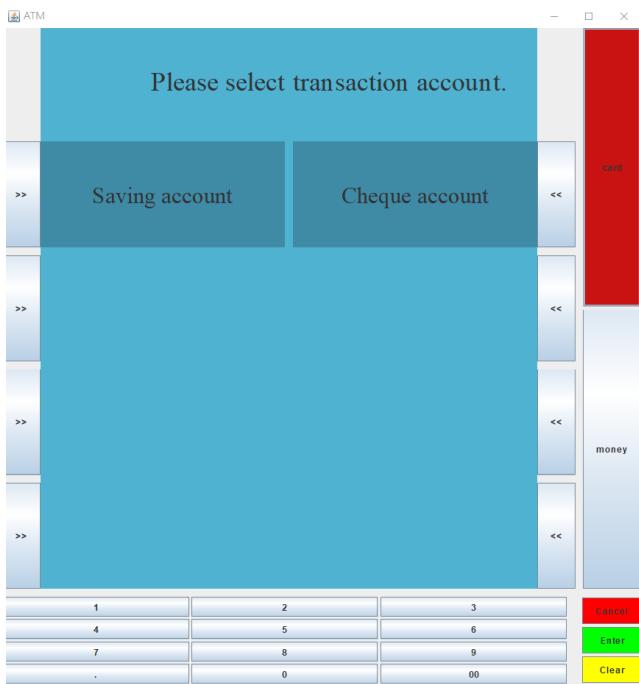


If the user input a invalid account number and didn't press the clear button to clear the input.

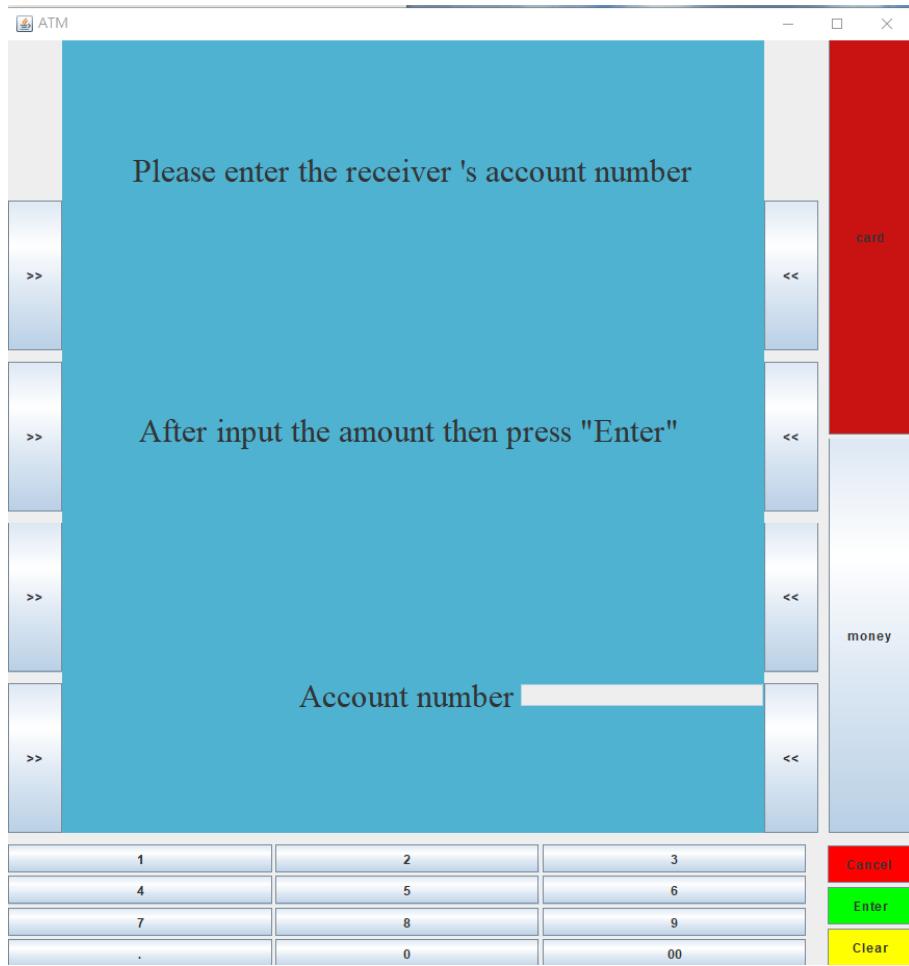


The message Invalid account number will be shown. And press “Enter” to back one step.

#### Test case 4: Transfer fund to the same bank account

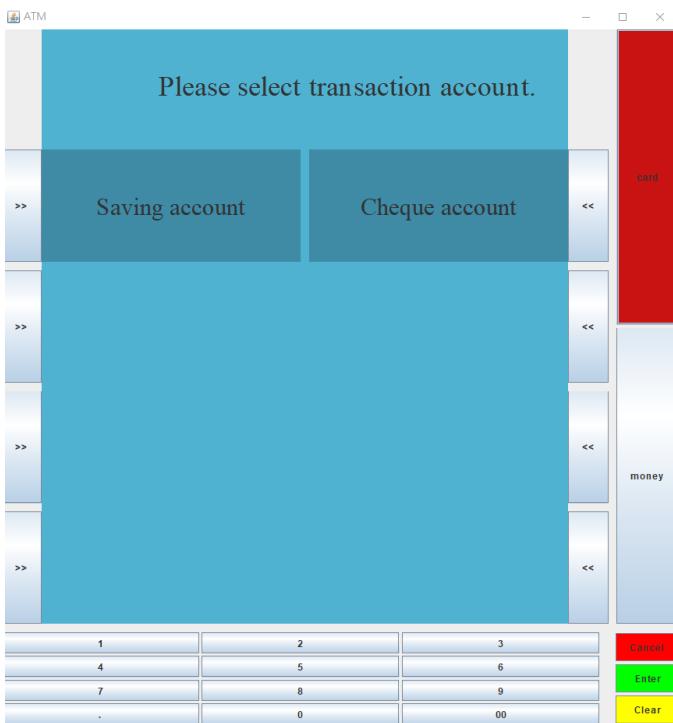
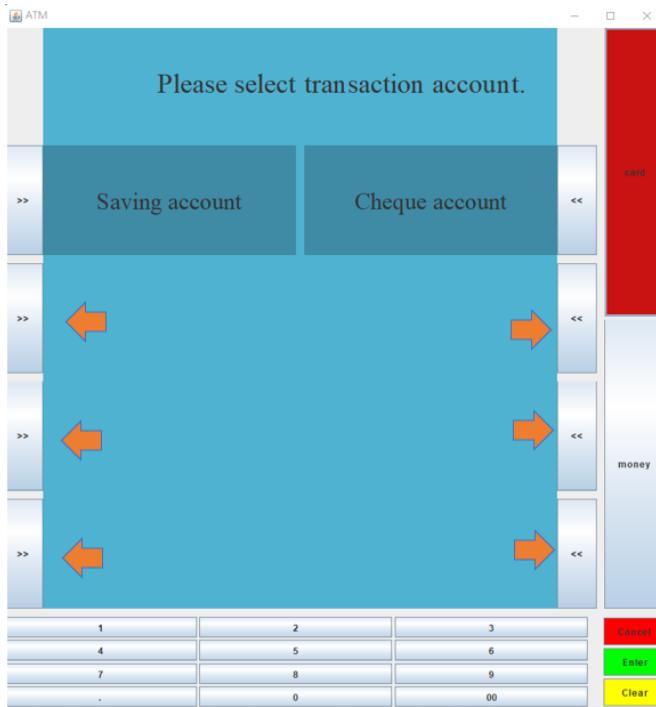


Since the current account is Saving account, so the user chooses the saving account to transfer the error will be shown.

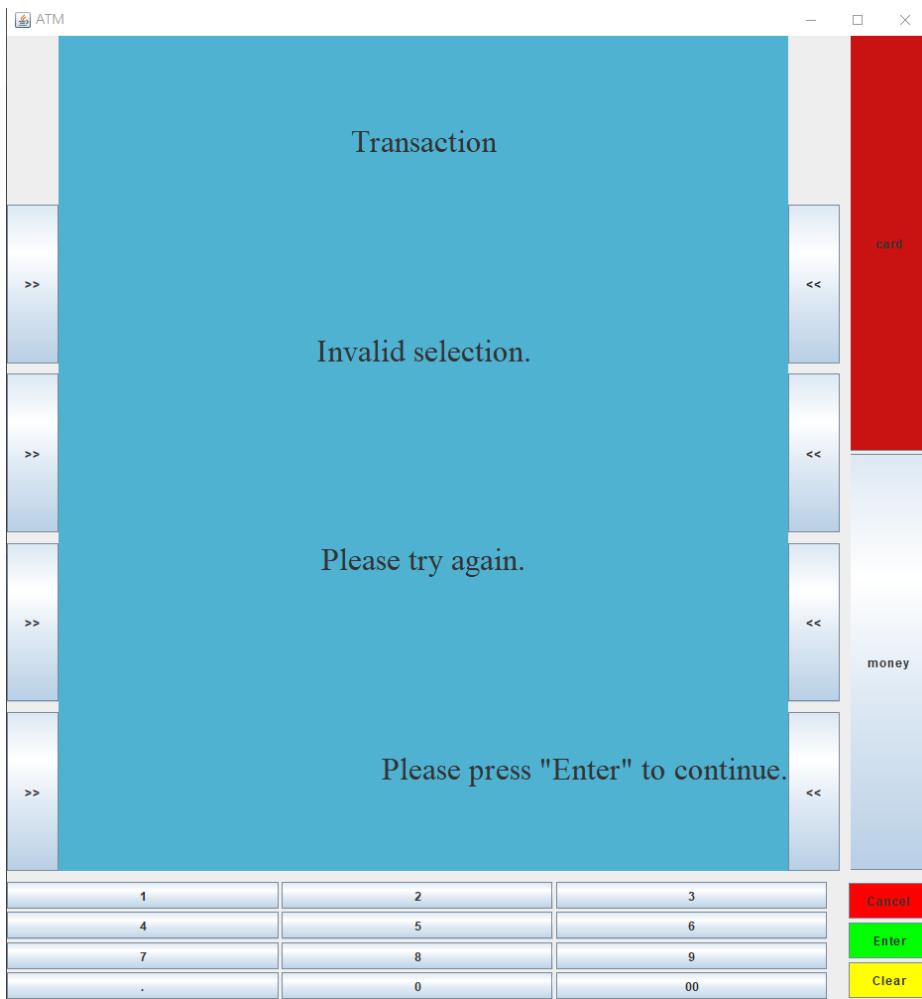


The system will require user to input the account number again like the ATM system we used before.

Test case 5: Press the wrong button on side.

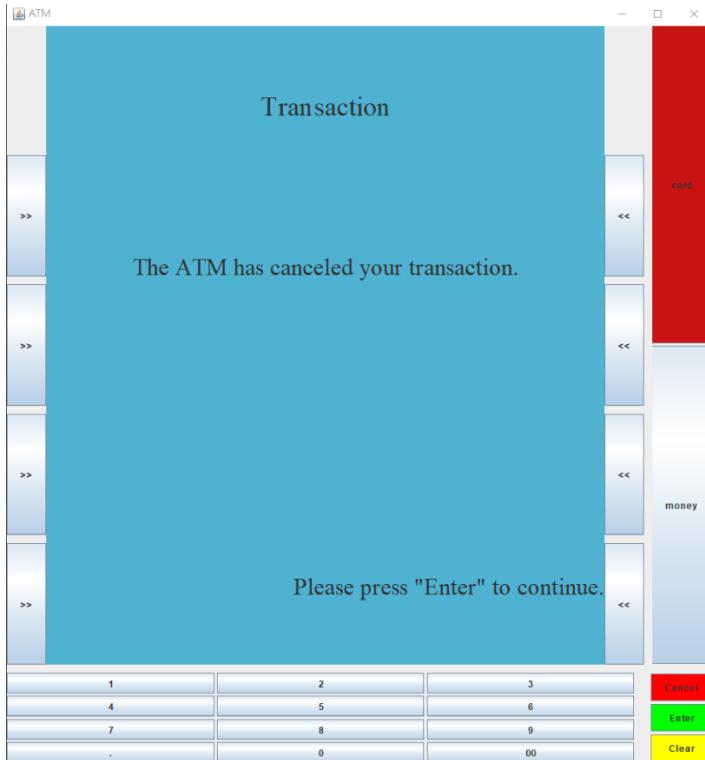
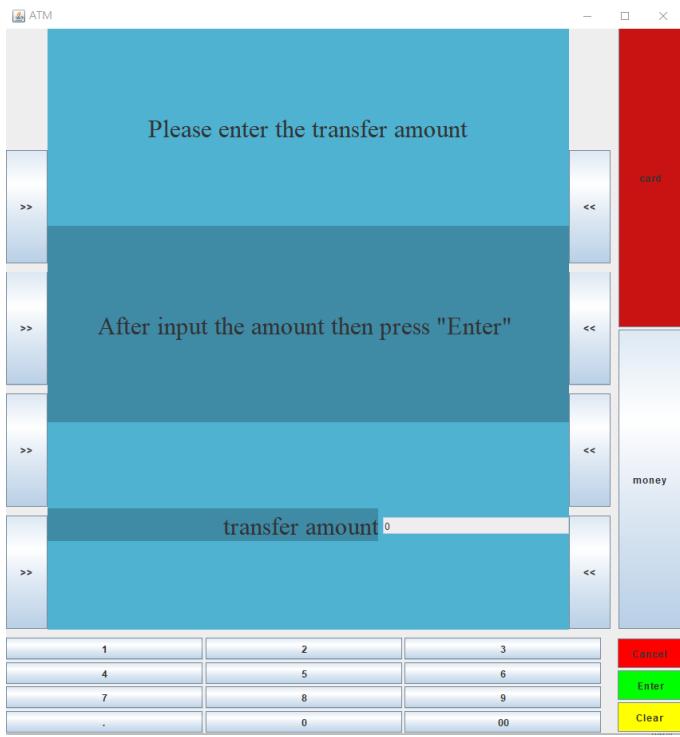


If the user presses the wrong button on side.



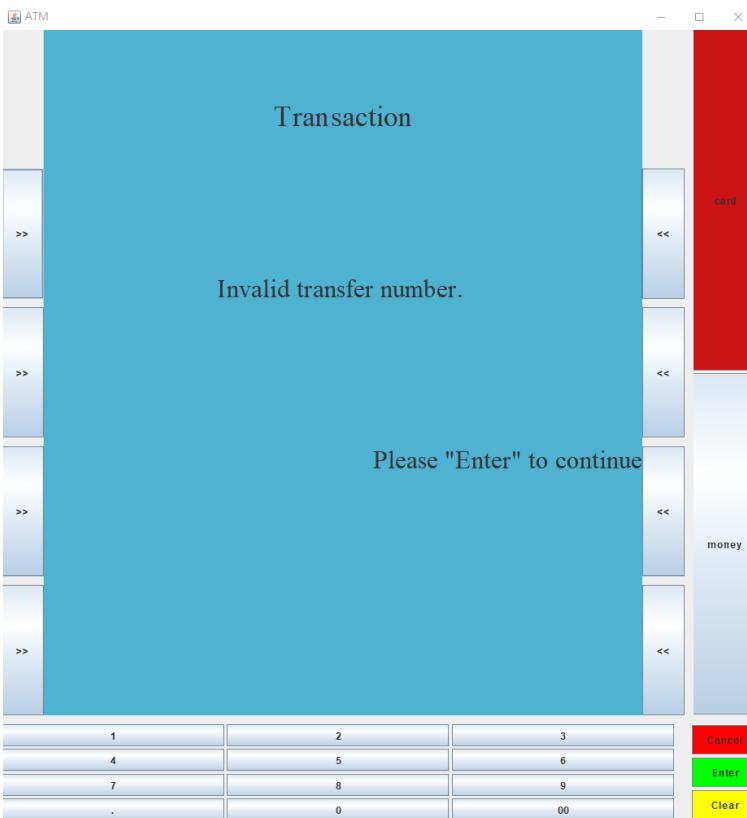
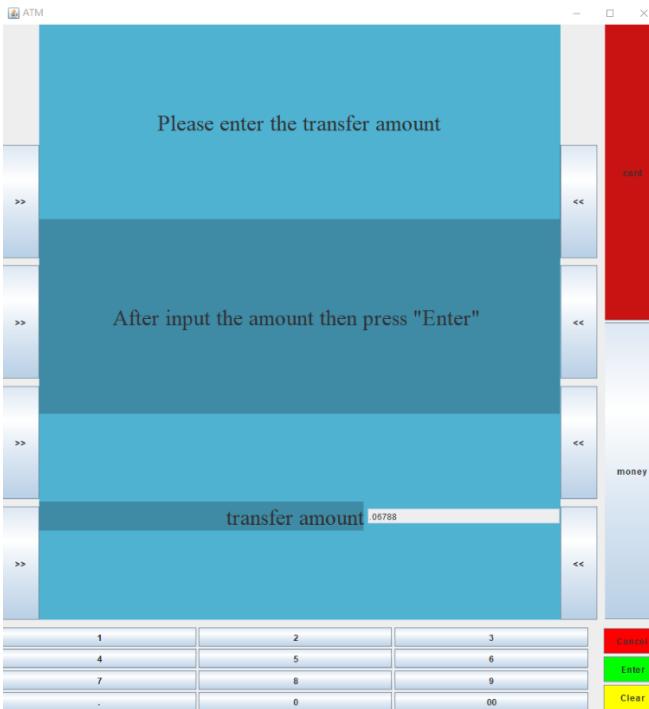
There will be a invalid selection message and user can press “enter” to choose the saving or cheque account again.

Test case 6: press 0 to cancel transaction when the user input zero.



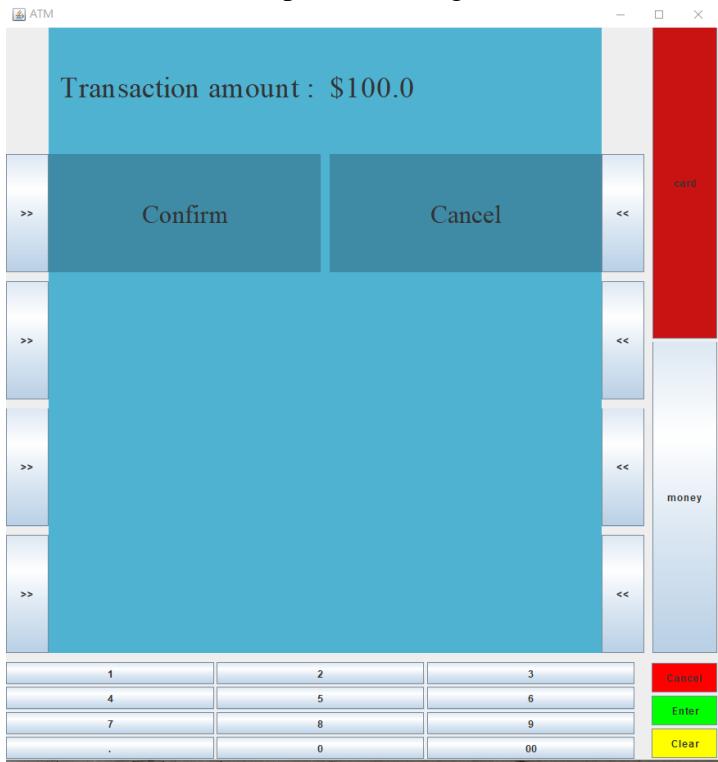
When the user input “0” the transaction will be cancelled.

### Test case 7: User input a invalid number as transfer amount



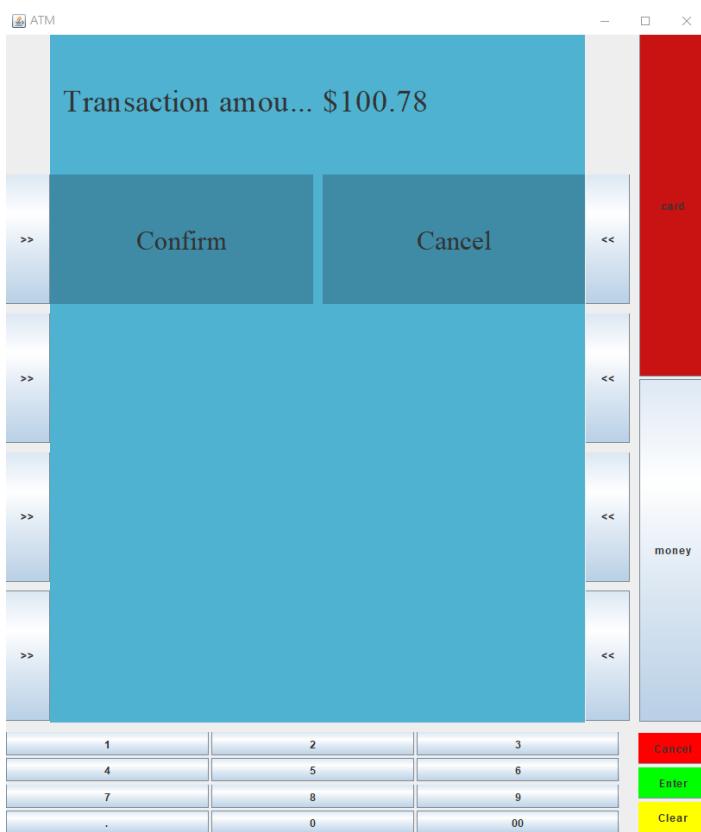
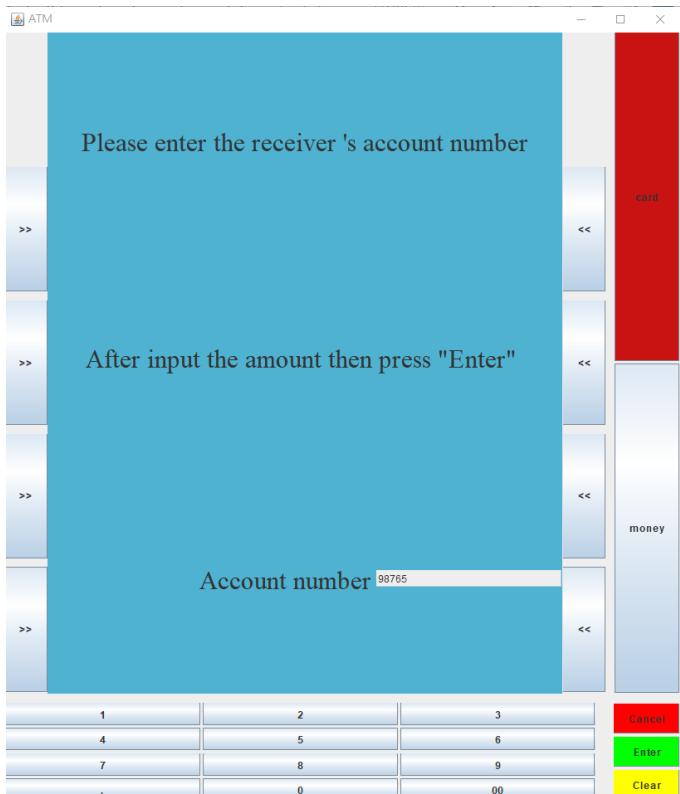
The message of Invalid transfer number will be shown.

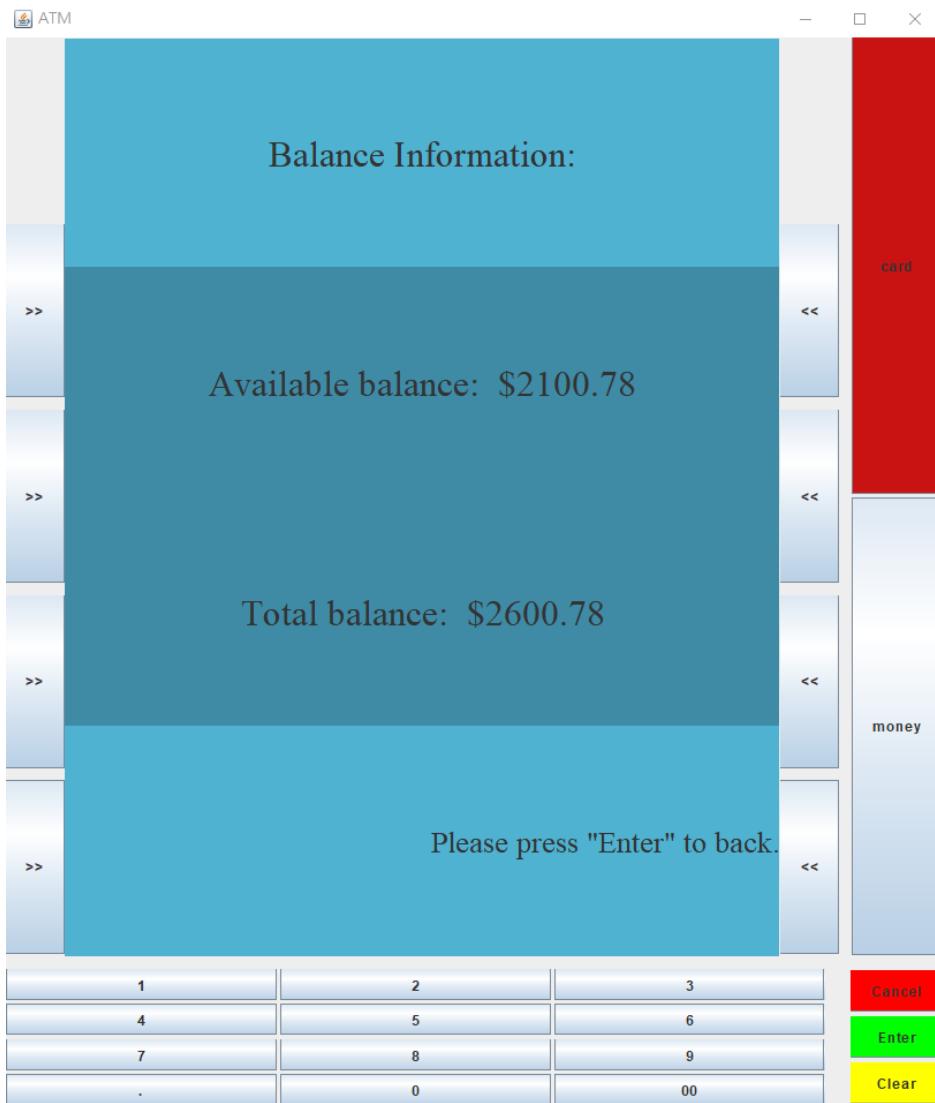
### Test case 8: The user presses wrong button when confirm or cancel step



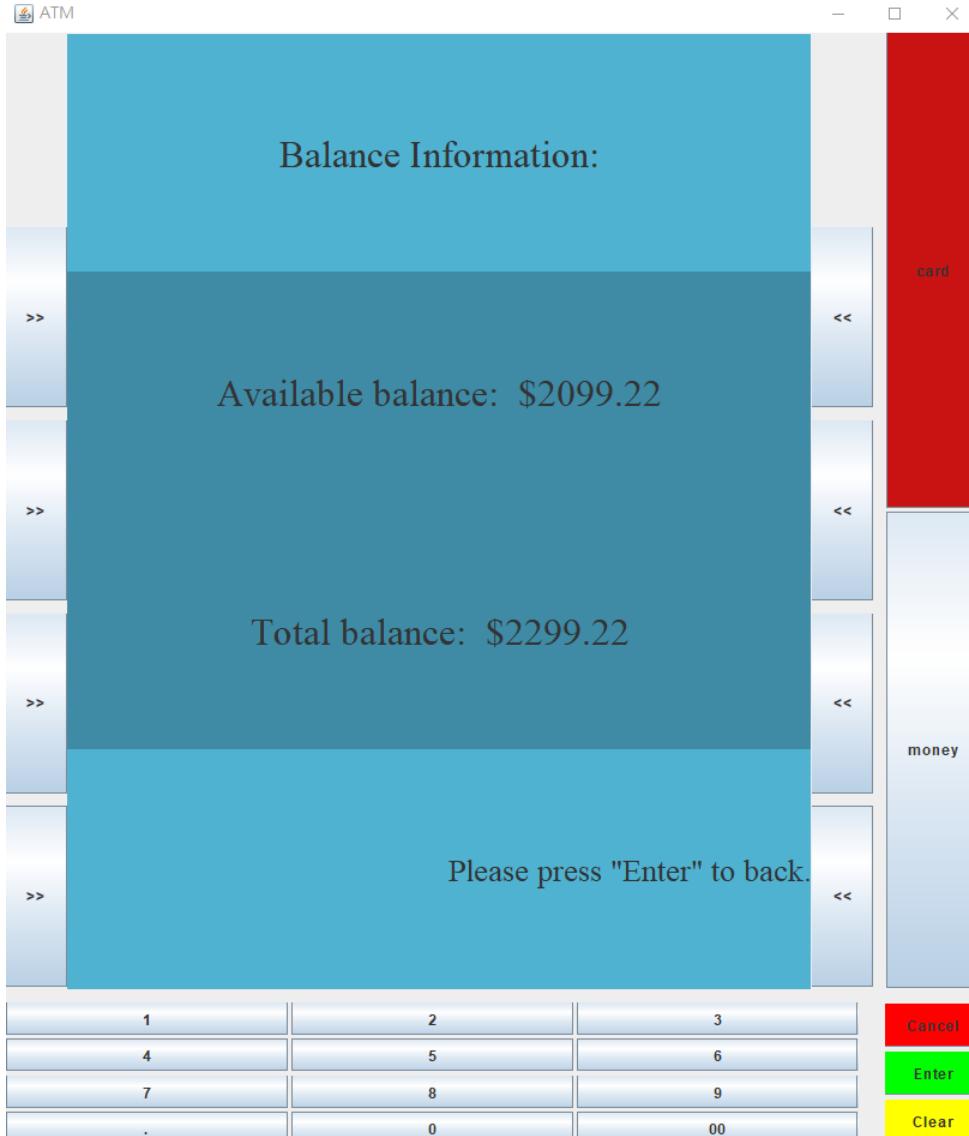
The system will request user to press again until user presses the correct button.

Test case 9: Account number 12345001(saving) transfer \$100.78 to 98765001(saving).





The balance information of 98765's saving account.



The balance information of 12345's saving account.

Account 12345001 (saving account)

Available balance =  $\$2200 - \$100.78 = \$2099.22$

Total balance =  $\$2400 - \$100.78 = \$2299.22$

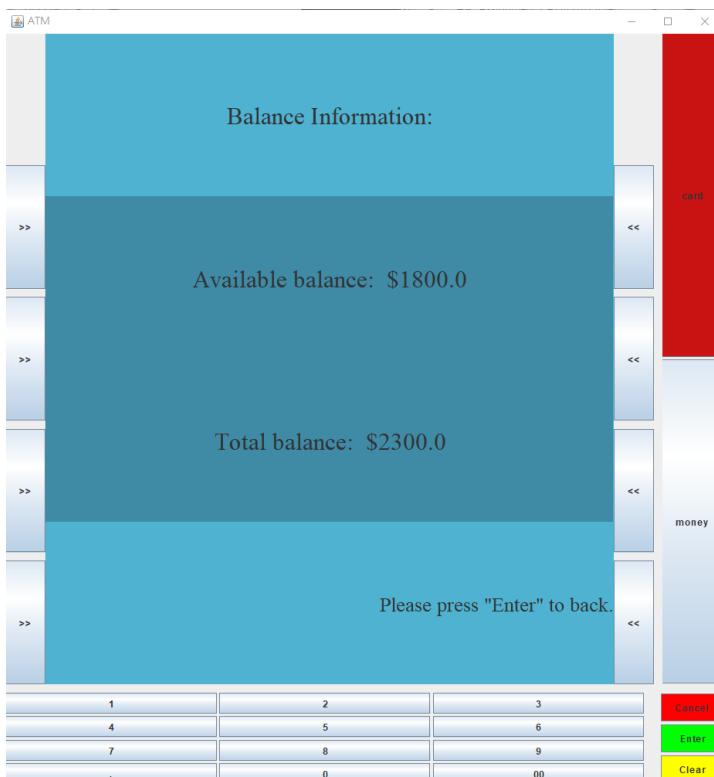
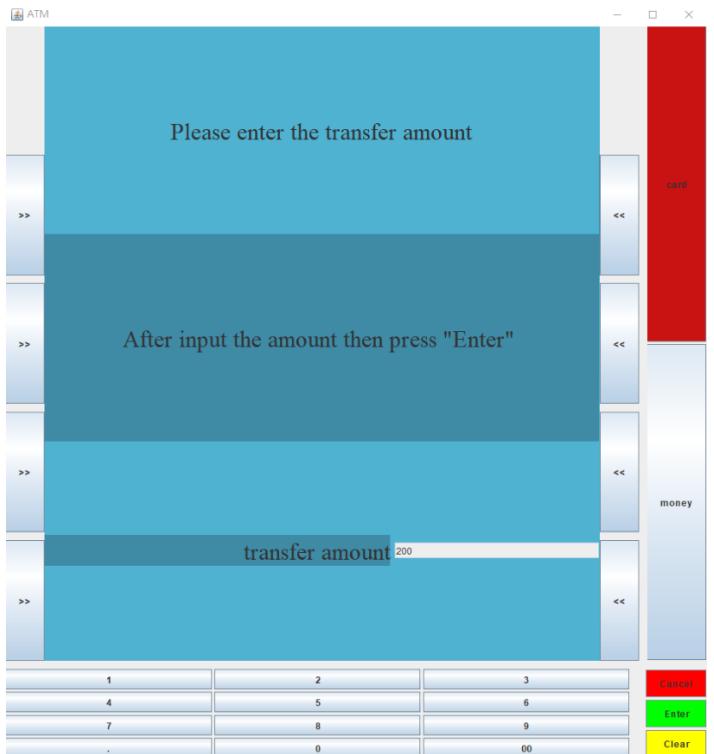
Account 98765001 (saving account)

Available balance =  $\$2000 + \$100.78 = \$2100.78$

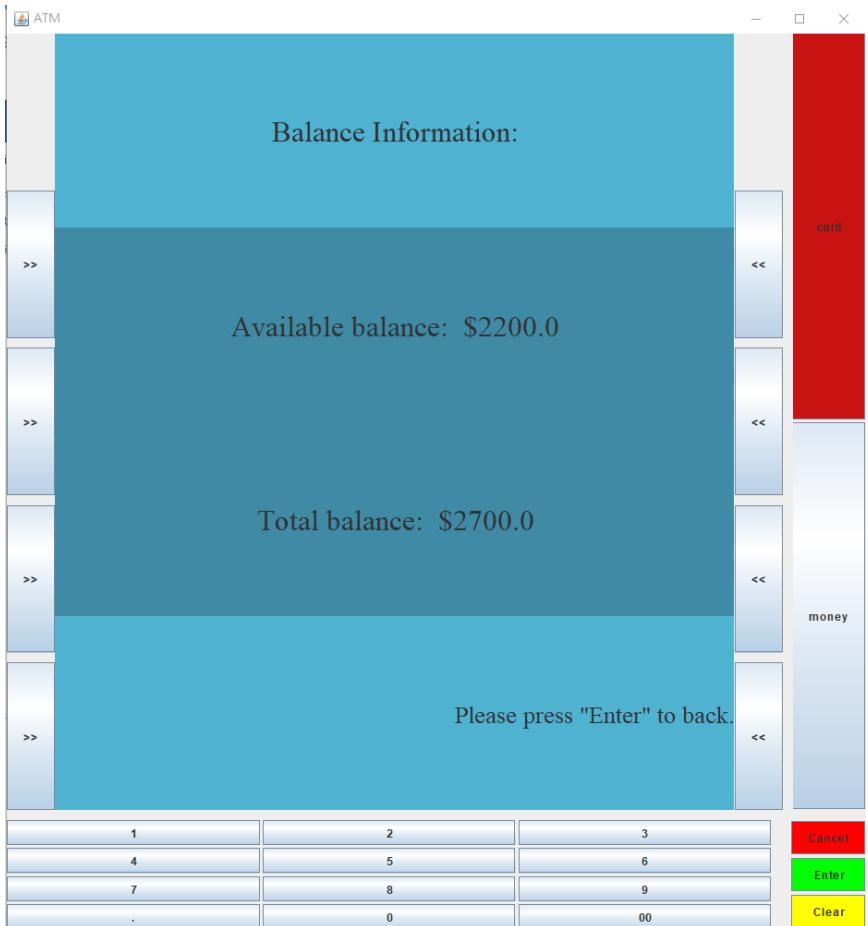
Total balance =  $\$2500 + \$100.78 = \$2600.78$

Thus, it is successful to transfer balance from 12345001 saving account to 98765001 saving account.

Test case 10: Account 98765 transfer \$200 from saving account to cheque account.



The balance information of 98765's saving account after decreasing.



The balance information of 98765's cheque account after increasing.

Account 98765001 (saving account)

Available balance = \$2000 - \$200 = \$1800

Total balance = \$2500 - \$200 = \$2300

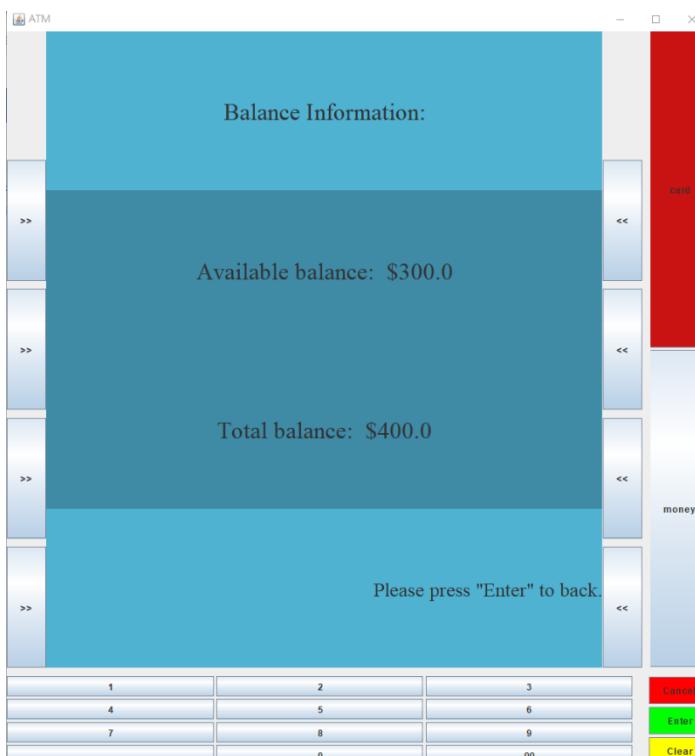
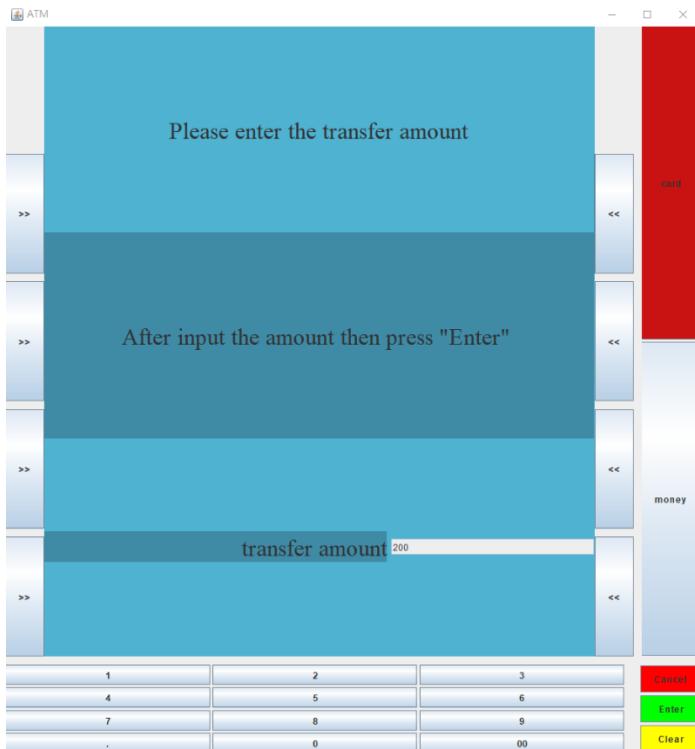
Account 98765002 (cheque account)

Available balance = \$2000 + \$200 = \$2200

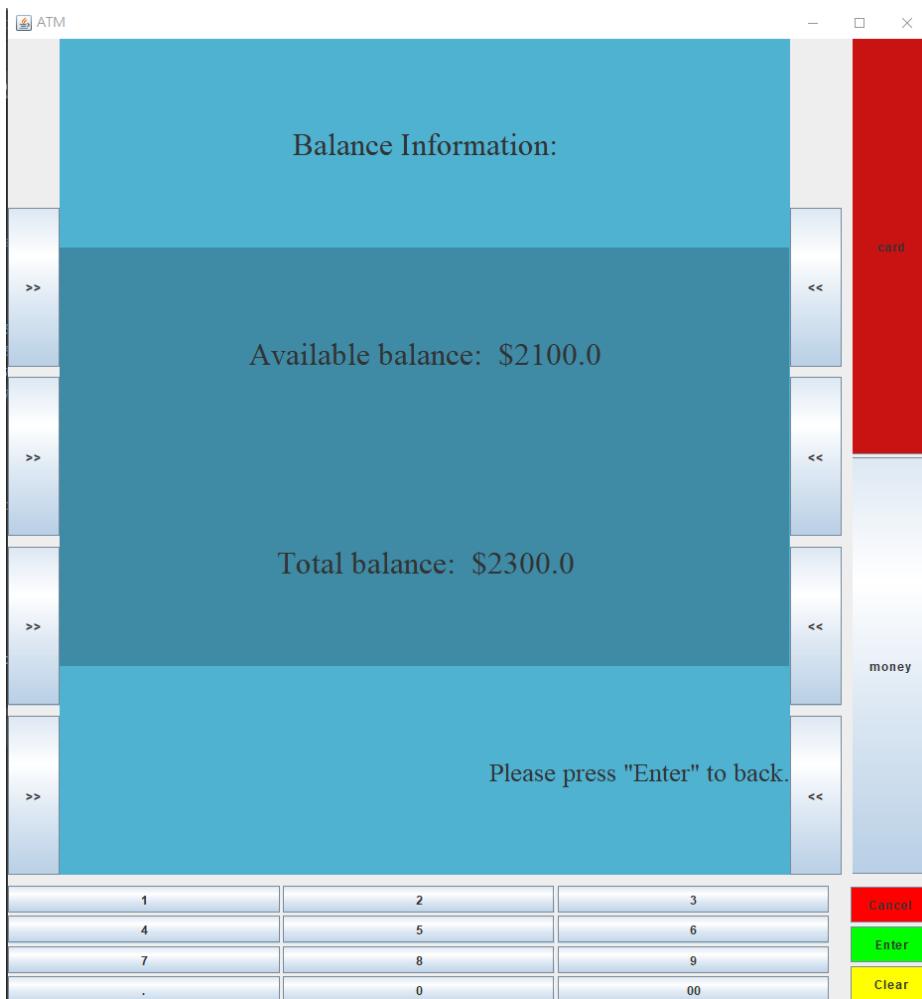
Total balance = \$2500 + \$200 = \$2700

Thus, it is successful to transfer balance from saving account to cheque account.

Test case 11: Account 12345 transfer \$100 from cheque account to saving account.



The balance information of 12345's cheque account after increasing.



The balance information of 12345's saving account after decreasing.

Account 12345001 (saving account)

Available balance = \$2300 - \$200 = \$2100

Total balance = \$2500 - \$200 = \$2300

Account 12345002 (cheque account)

Available balance = \$100 + \$200 = \$300

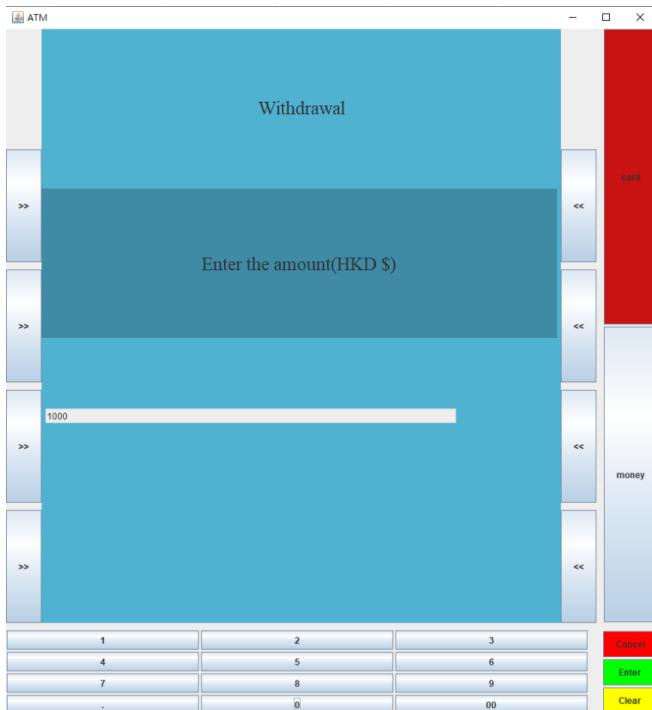
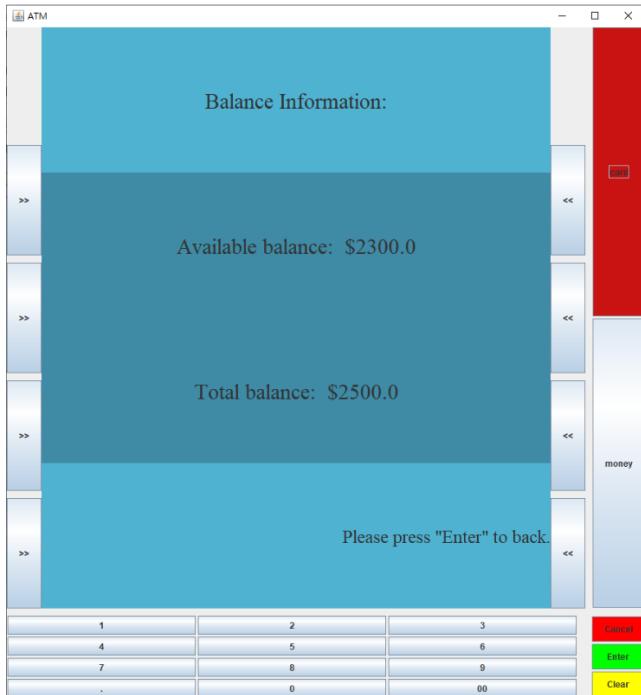
Total balance = \$200 + \$200 = \$400

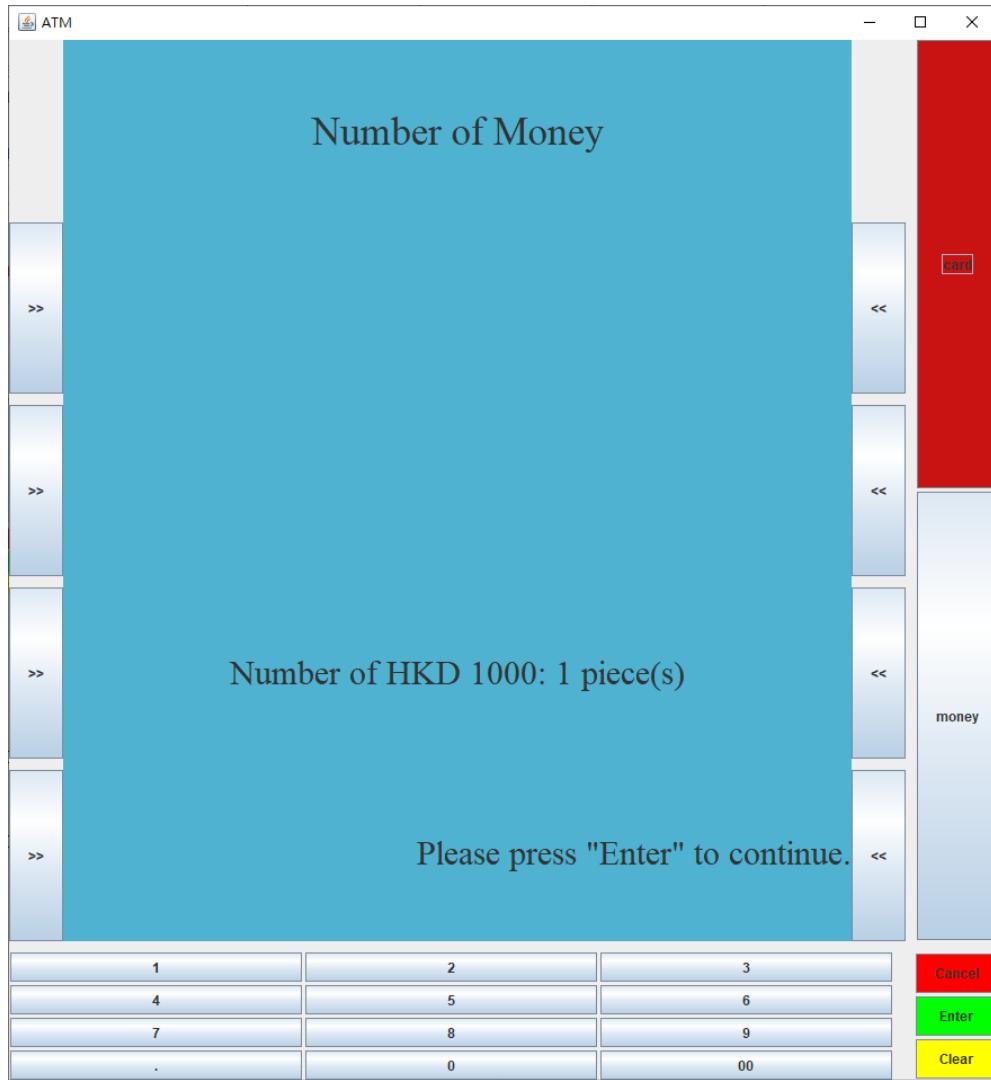
Thus, it is successful to transfer balance from saving account to cheque account.

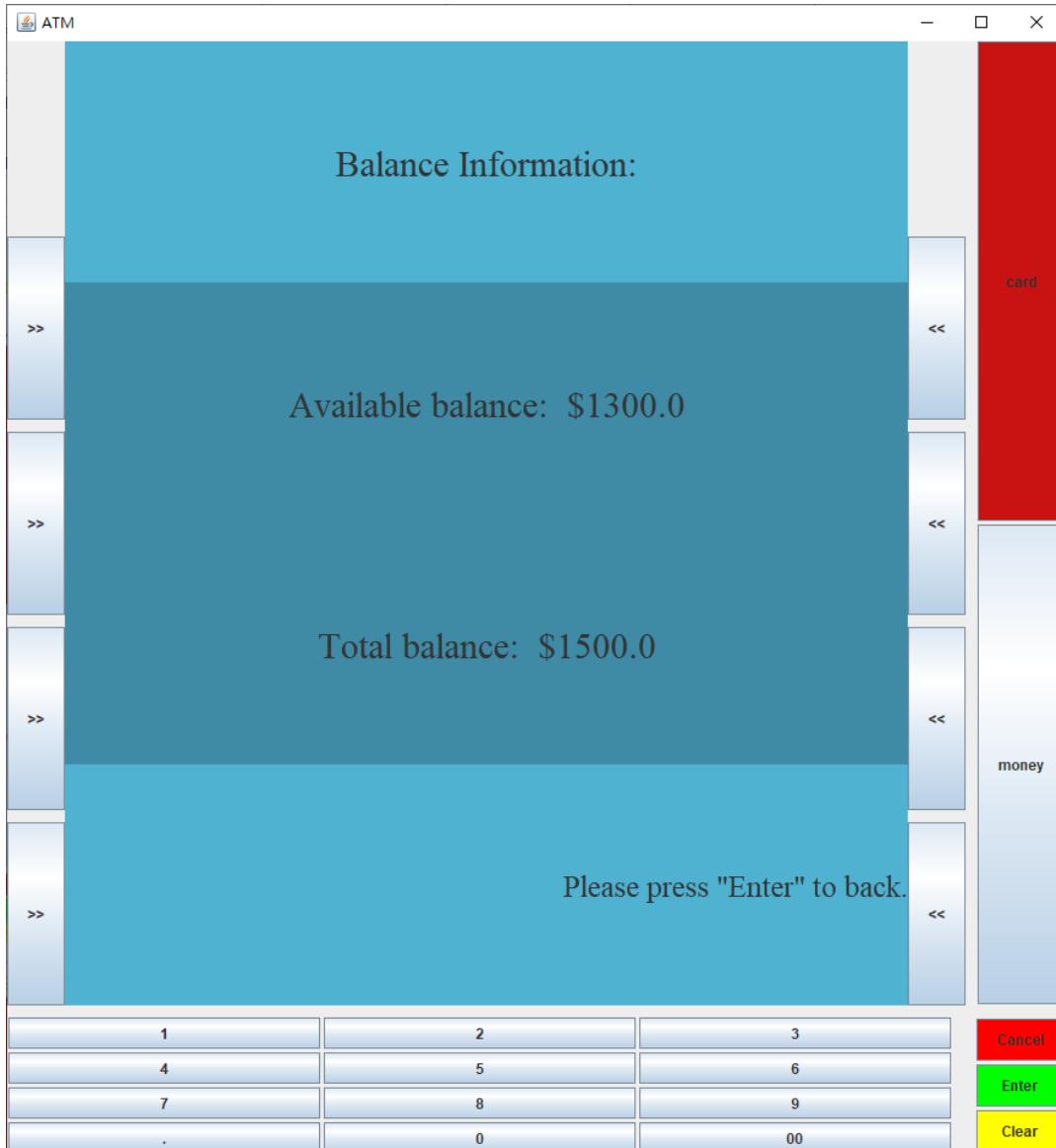
## Withdrawal

### Valid

Test case 1: account 12345(saving account) withdrawal \$1000







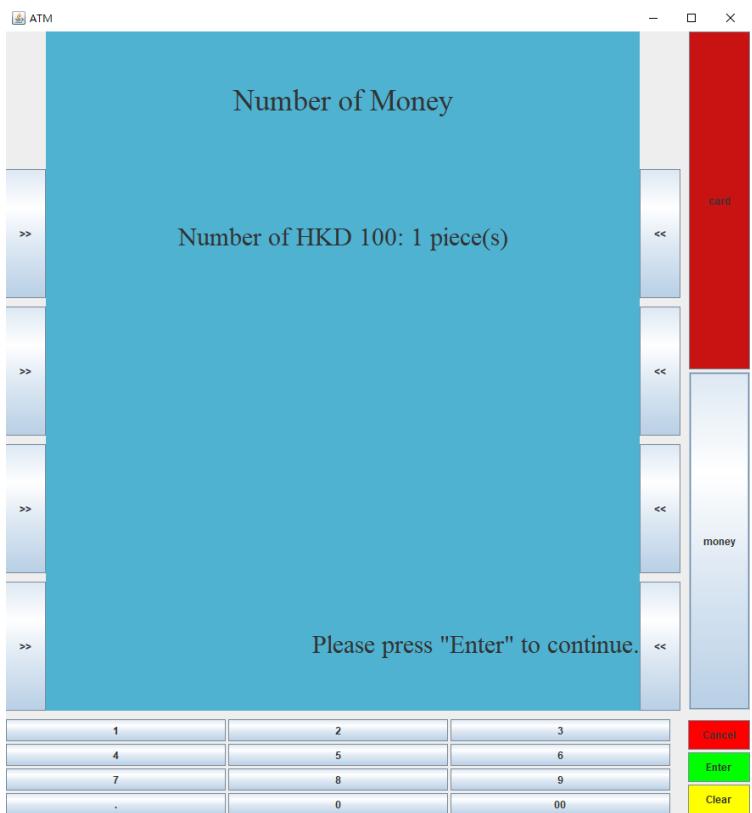
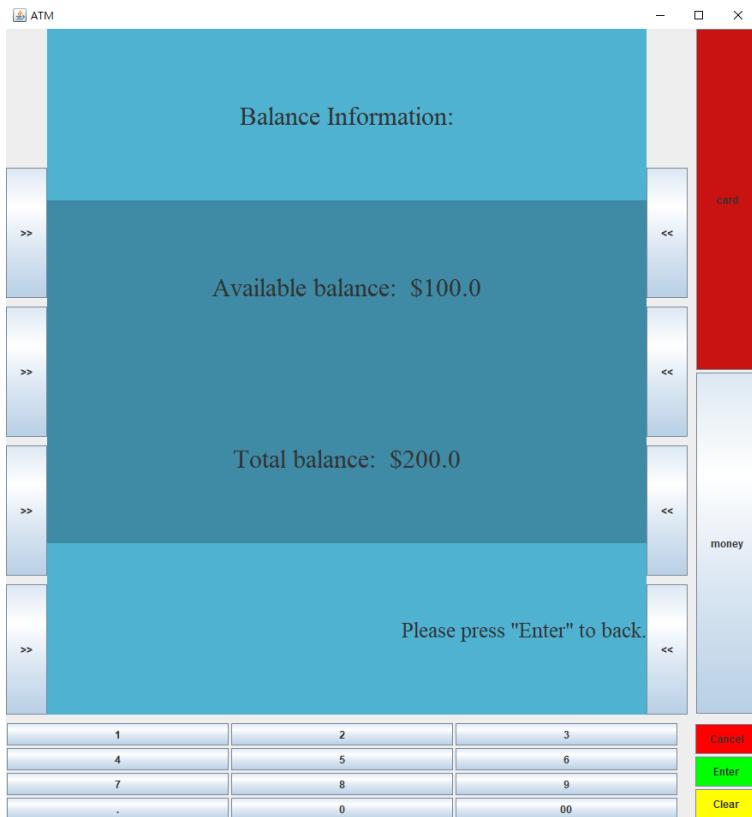
The balance information of 12345's saving account after withdrawal.

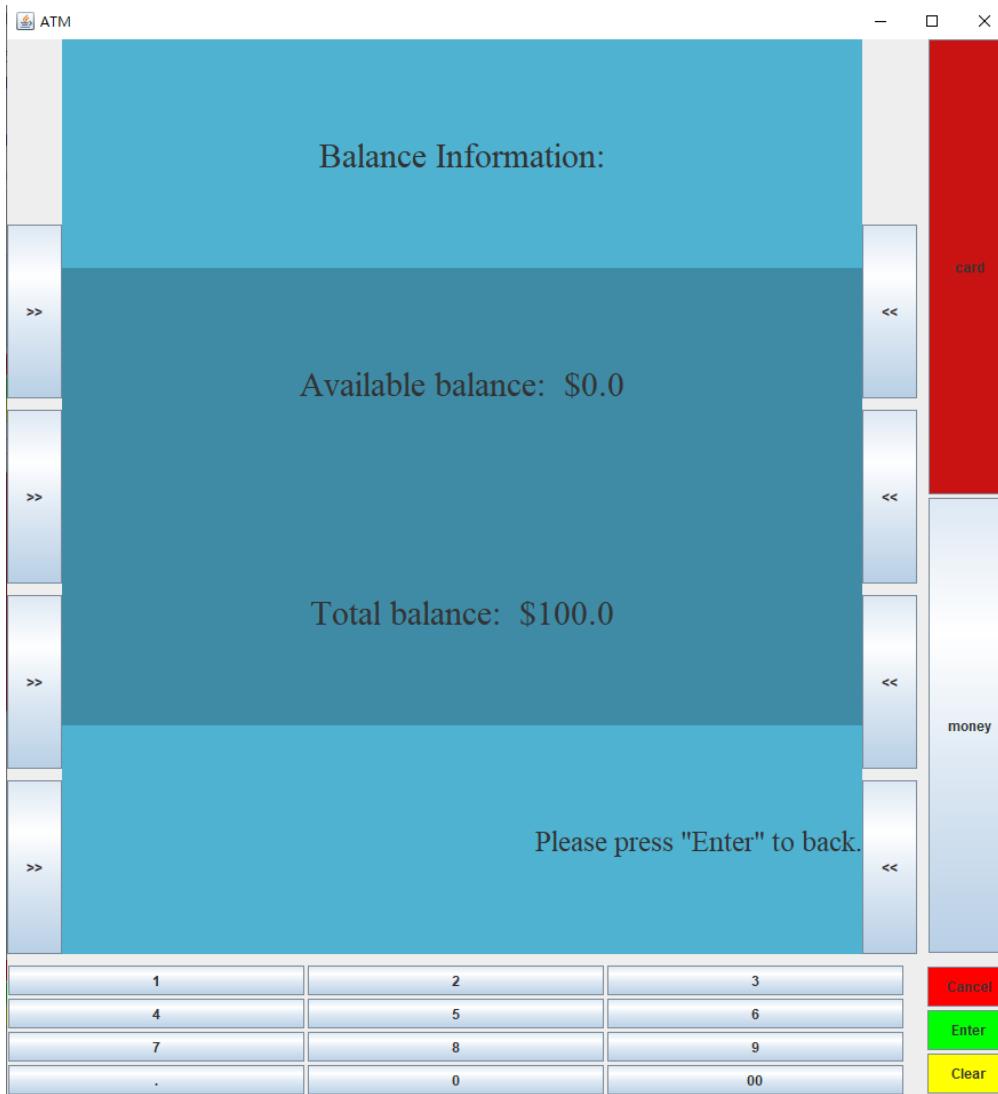
$$\text{Available balance} = \$2300 - \$1000 = \$1300$$

$$\text{Total balance} = \$2500 - \$1000 = \$1500$$

Thus, it is successful to update both balance after withdrawal from 12345's saving account.

Test case 2: account 12345(cheque account) withdrawal \$100





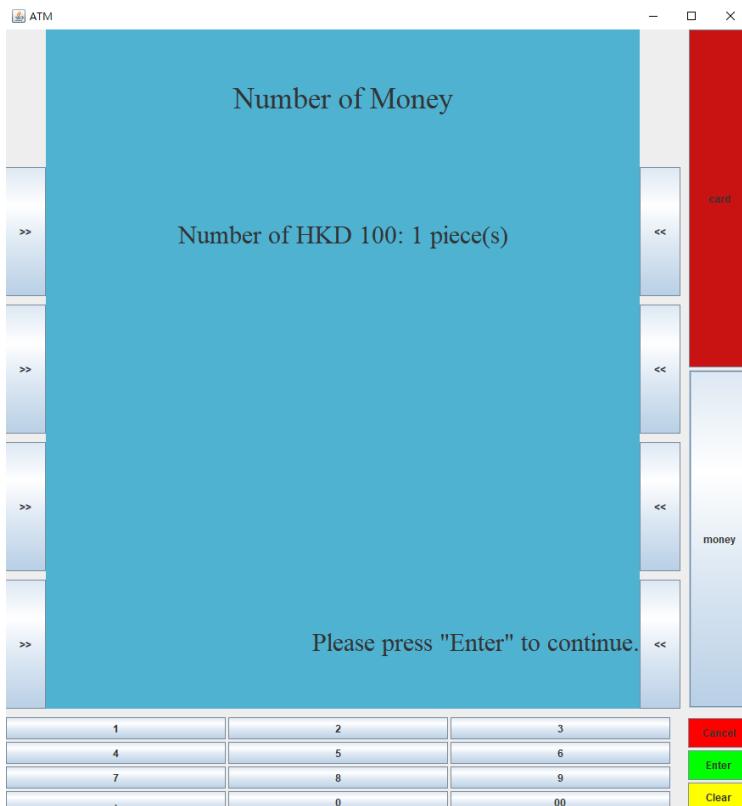
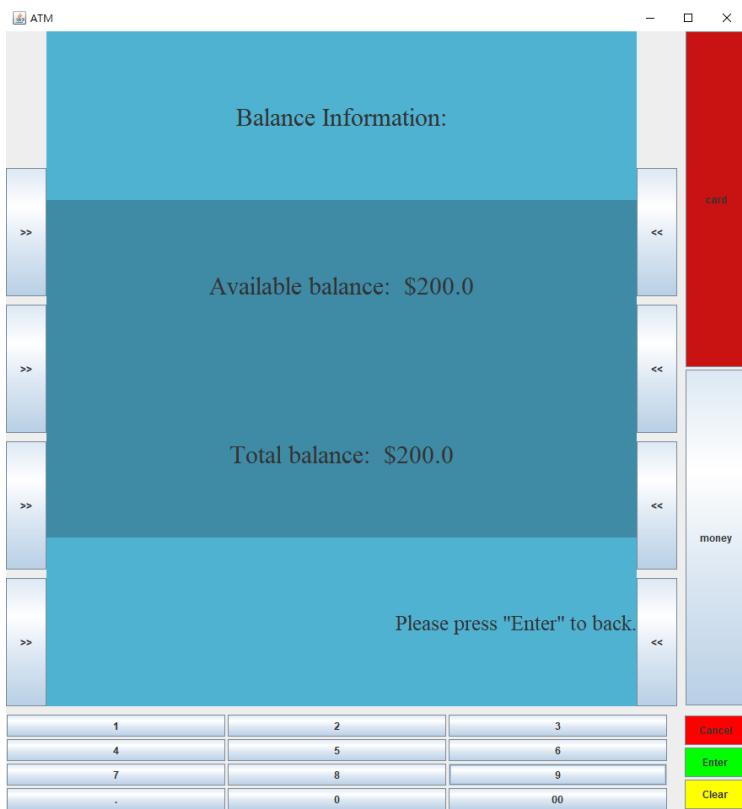
The balance information of 12345's cheque account after withdrawal.

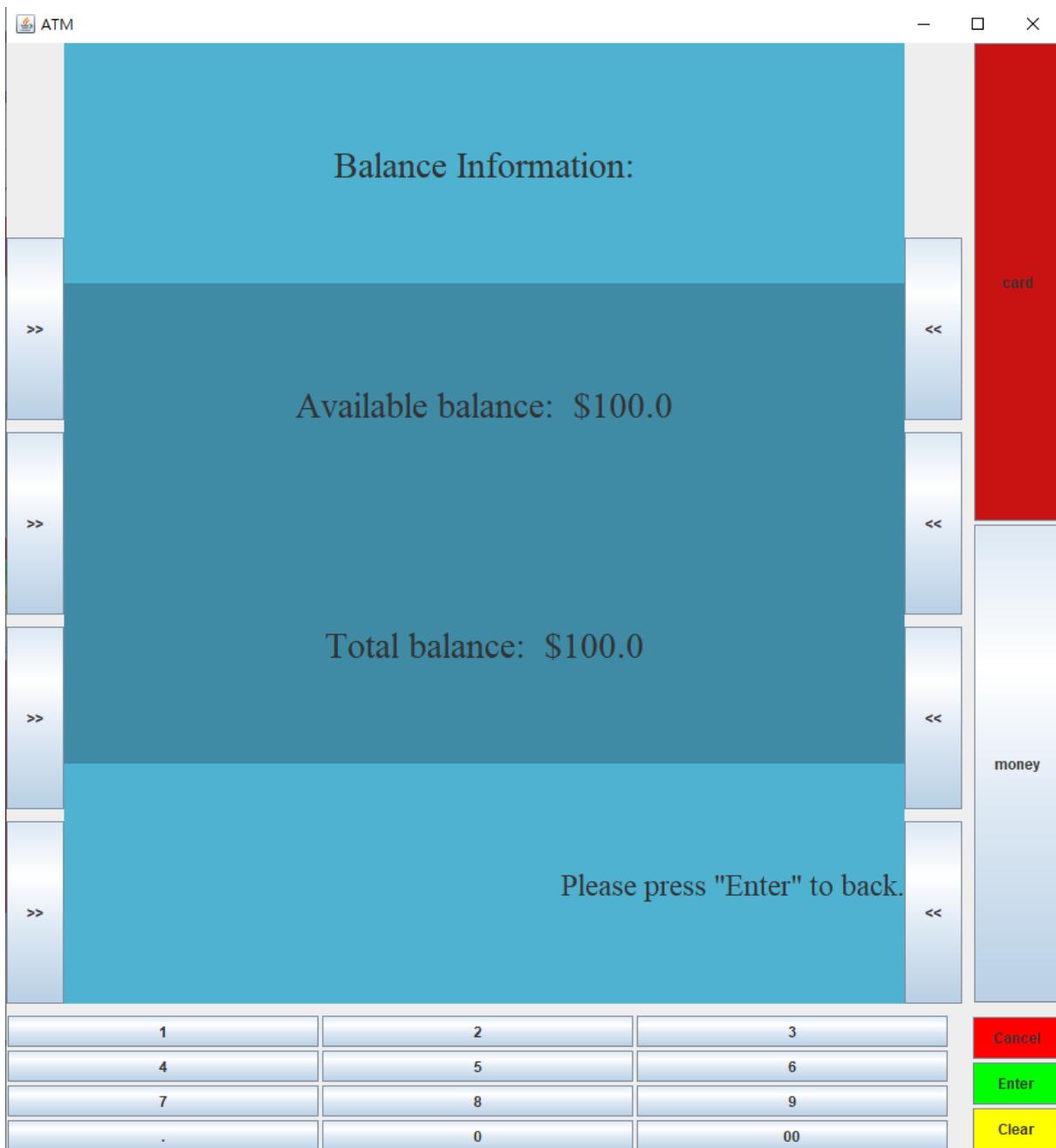
$$\text{Available balance} = \$100 - \$100 = \$0$$

$$\text{Total balance} = \$200 - \$100 = \$100$$

Thus, it is successful to update both balance after withdrawal from 12345's cheque account.

Test case 3: account 98765(saving account) withdrawal \$100





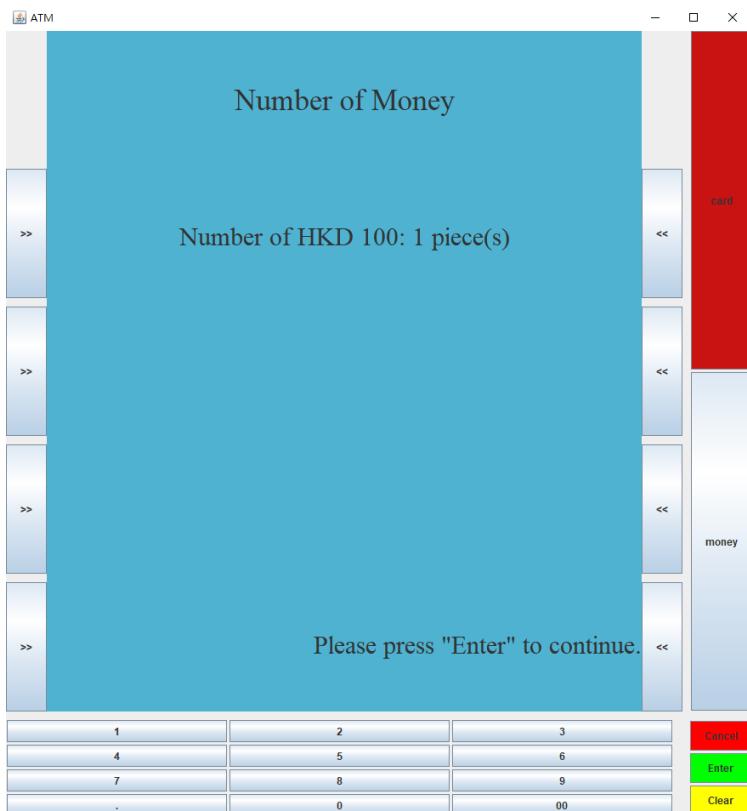
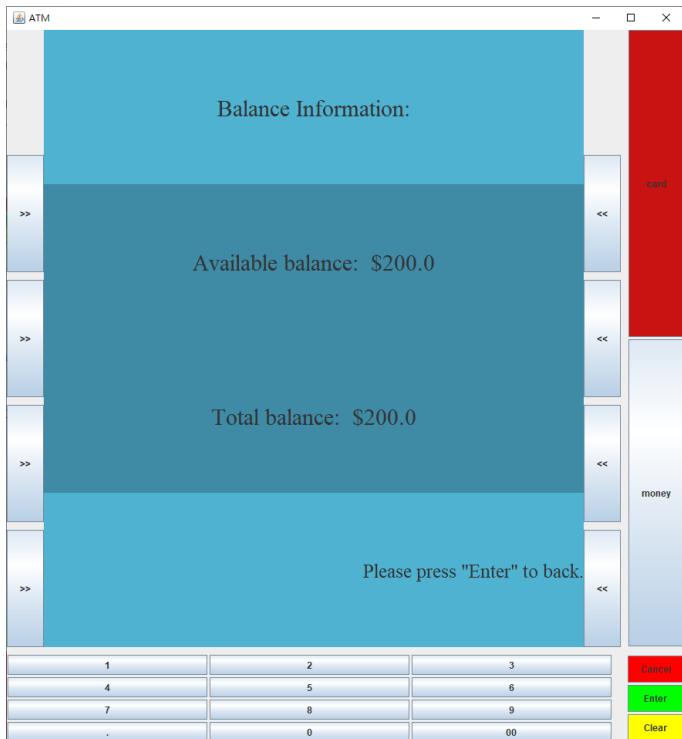
The balance information of 98765's saving account after withdrawal.

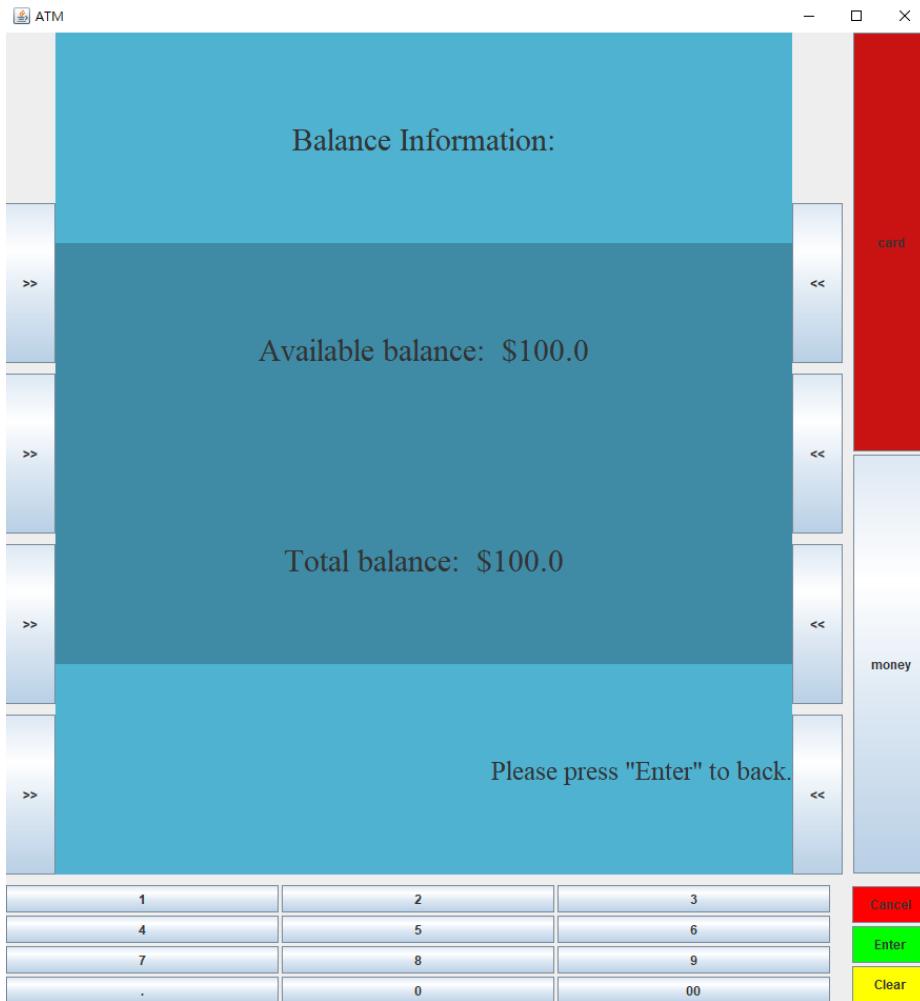
$$\text{Available balance} = \$200 - \$100 = \$100$$

$$\text{Total balance} = \$200 - \$100 = \$100$$

Thus, it is successful to update both balance after withdrawal from 98765's saving account.

Test case 4: account 98765(cheque account) withdrawal \$100





The balance information of 98765's cheque account after withdrawal.

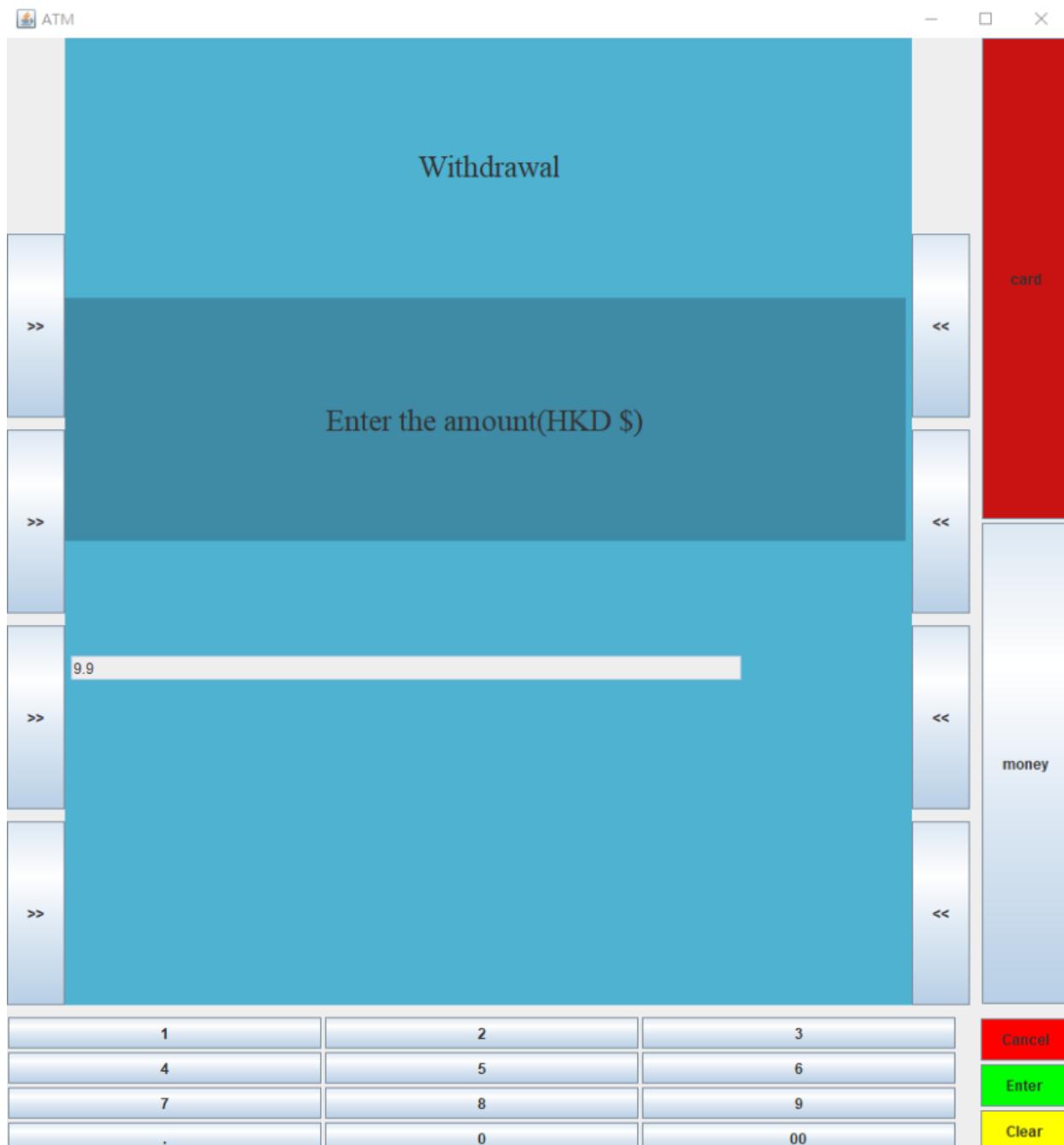
$$\text{Available balance} = \$200 - \$100 = \$100$$

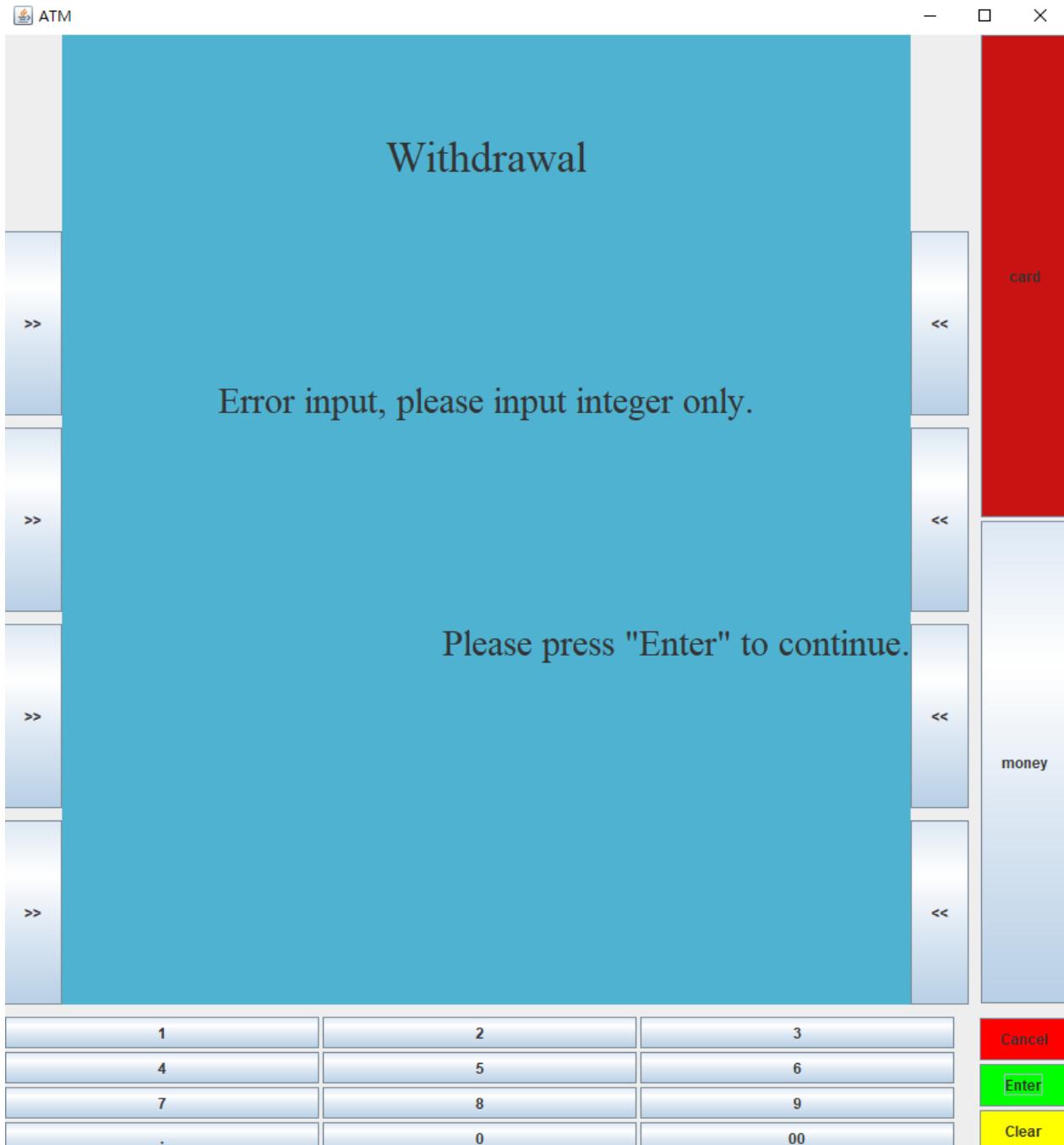
$$\text{Total balance} = \$200 - \$100 = \$100$$

Thus, it is successful to update both balance after withdrawal from 98765's cheque account.

## Invalid

Test case 1: Withdraw \$9.9

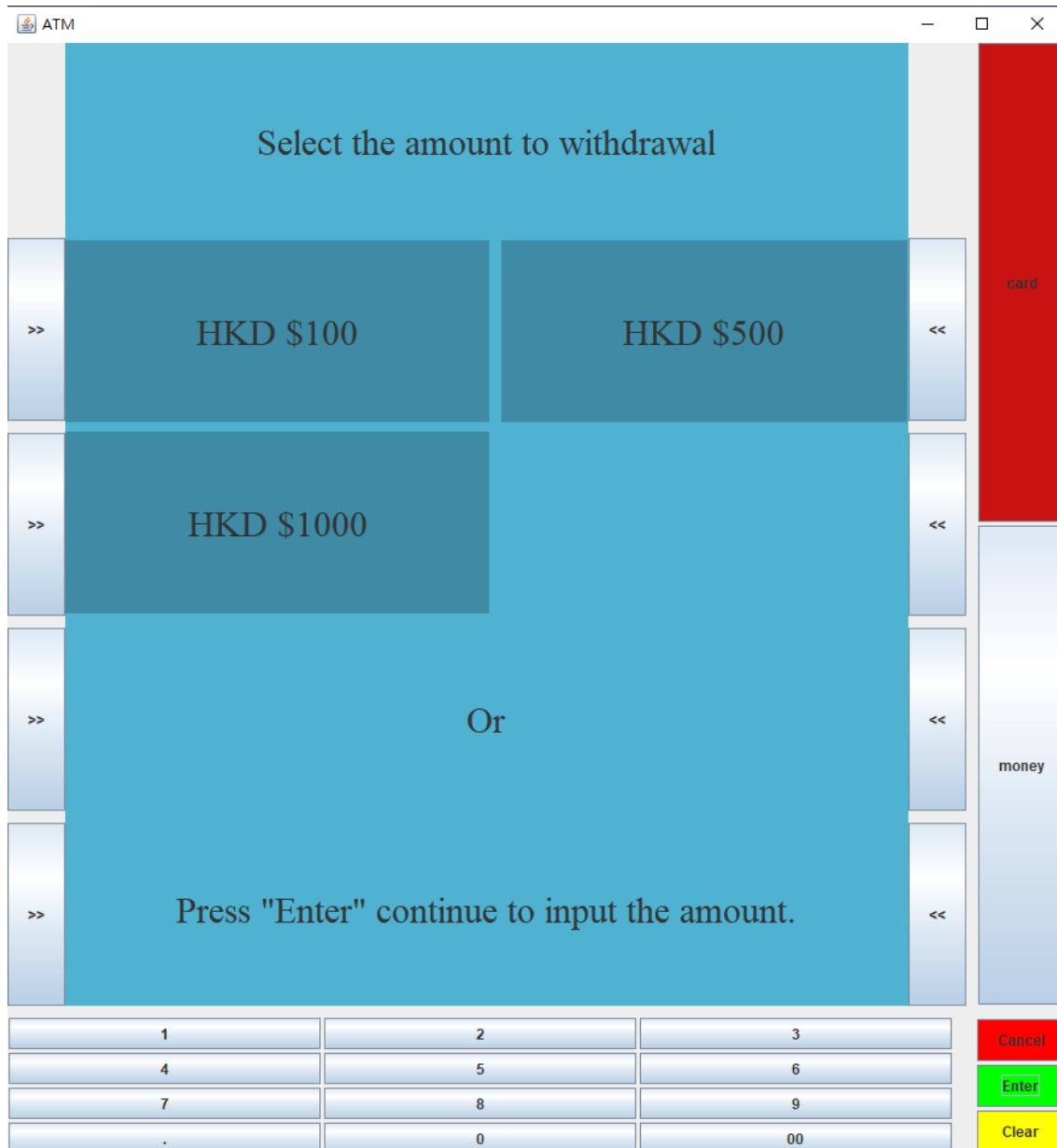


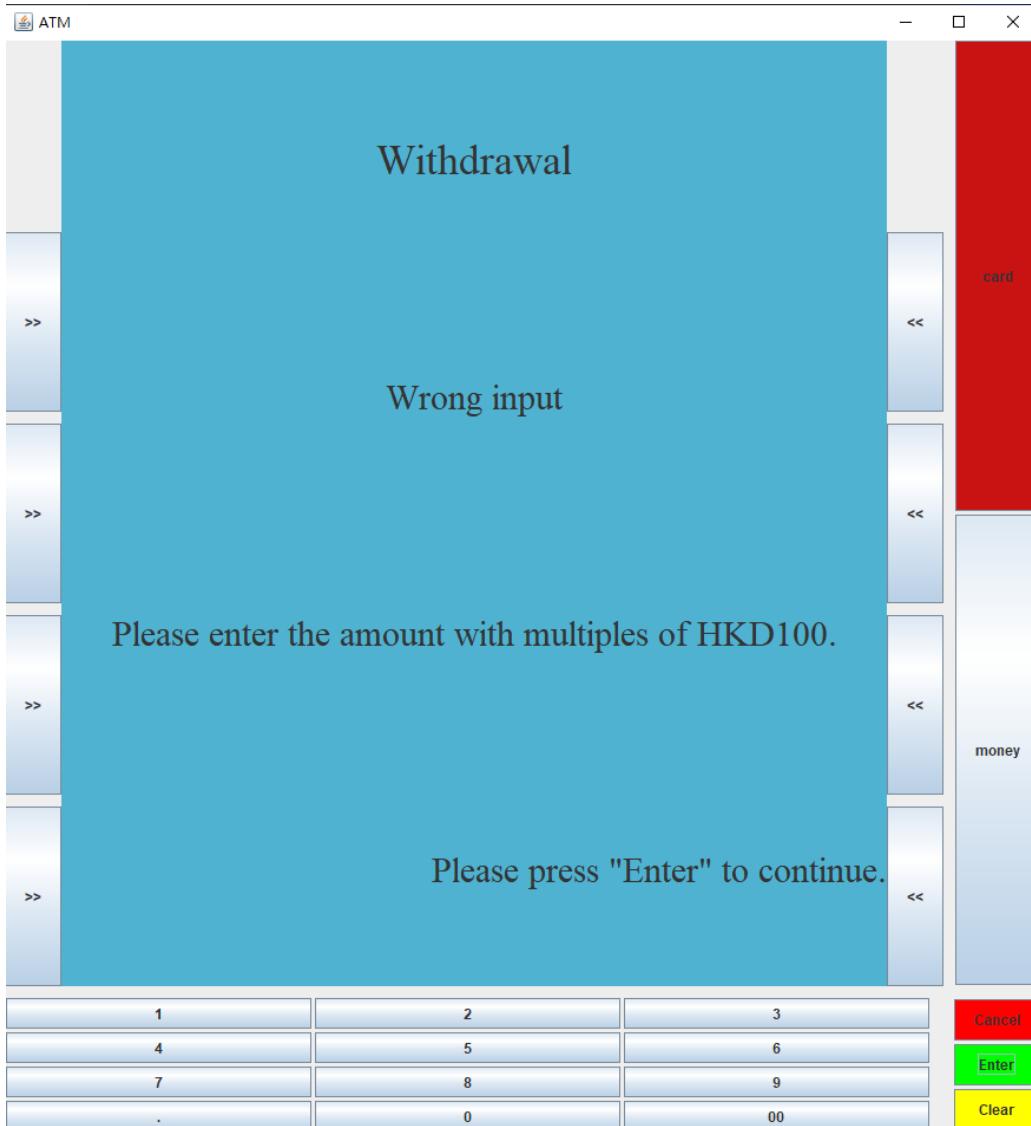


It will show the error message when input amount is not integer.

Test case 2: Withdraw valid amount <= available balance

Example for \$10

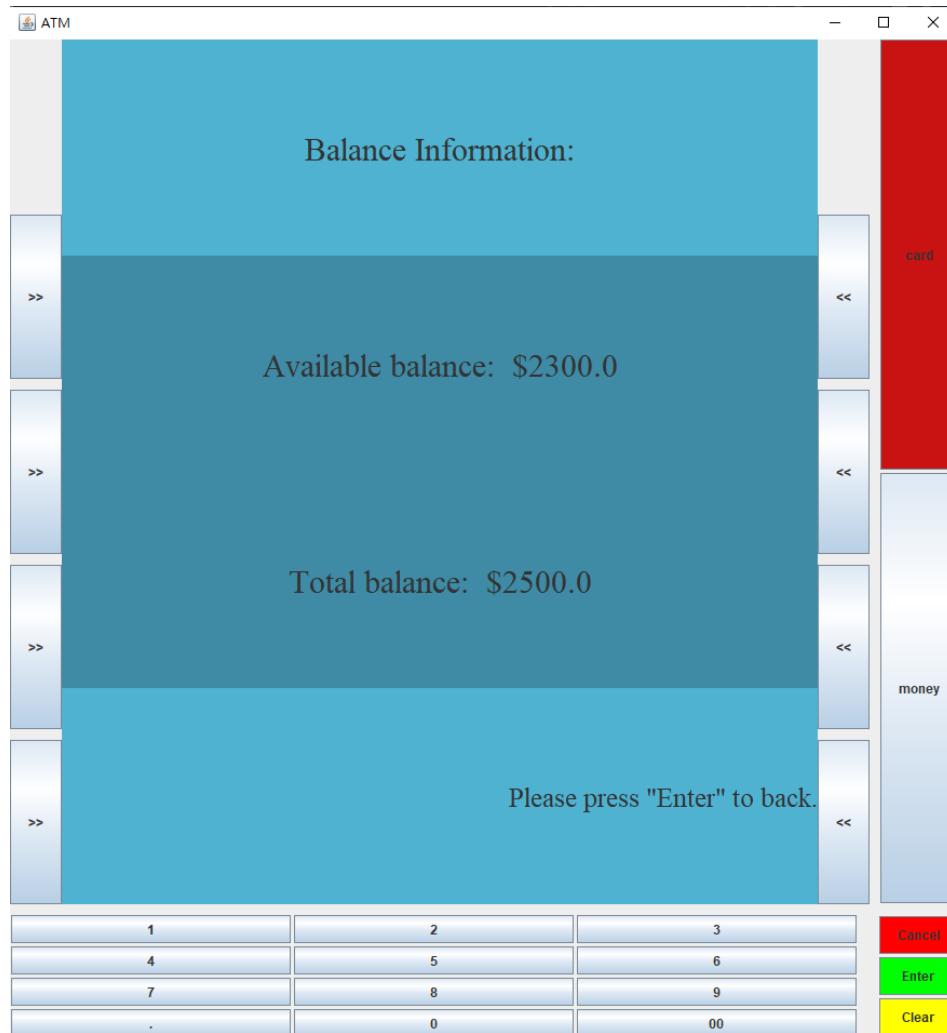




It will show the error message when input amount is not multiples of HKD\$100.

Test case 3: Withdrawal amount > available balance

For example: input \$2400





- □ ×

Withdrawal

>>

Enter the amount(HKD \$)

>>

2400

>>

>>

card

<<

<<

money

<<

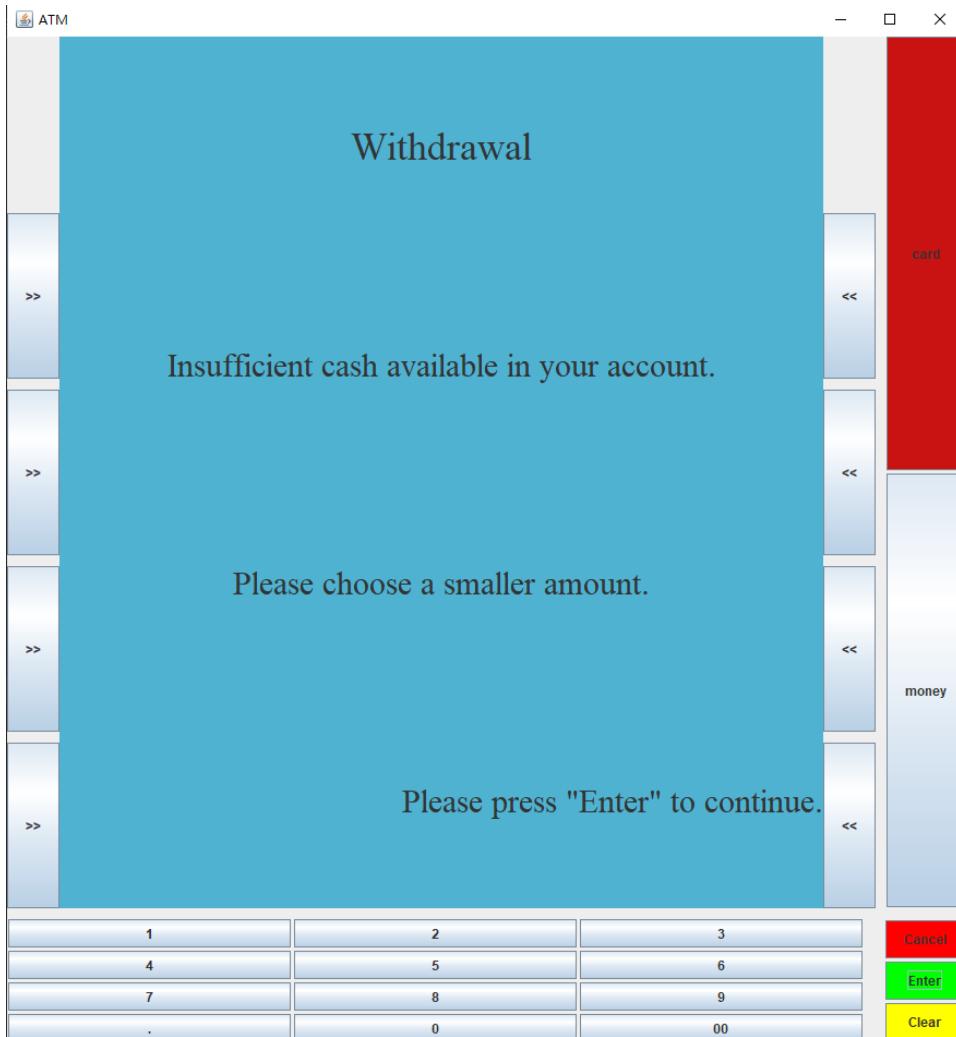
<<

1	2	3
4	5	6
7	8	9
.	0	00

Cancel

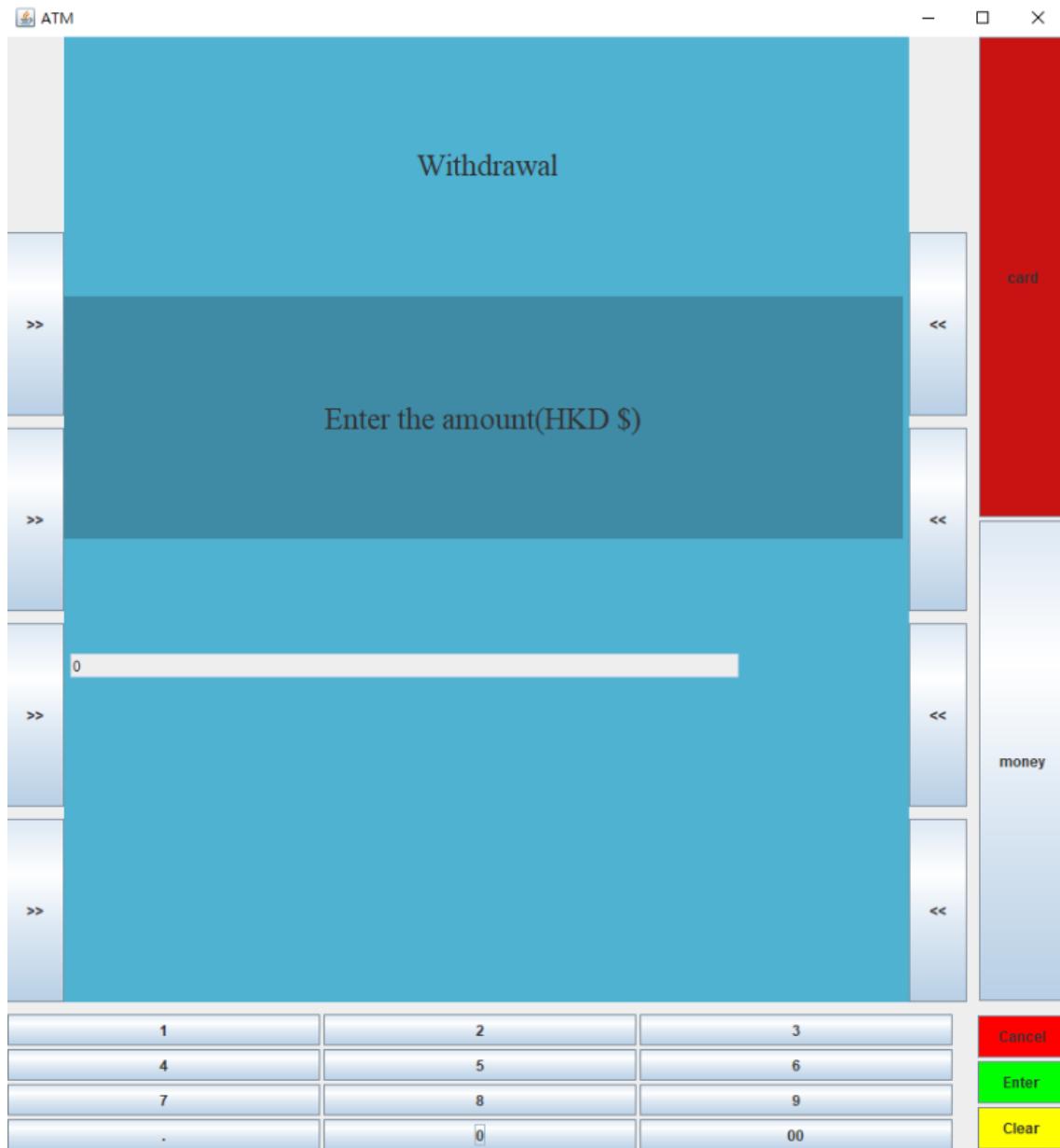
Enter

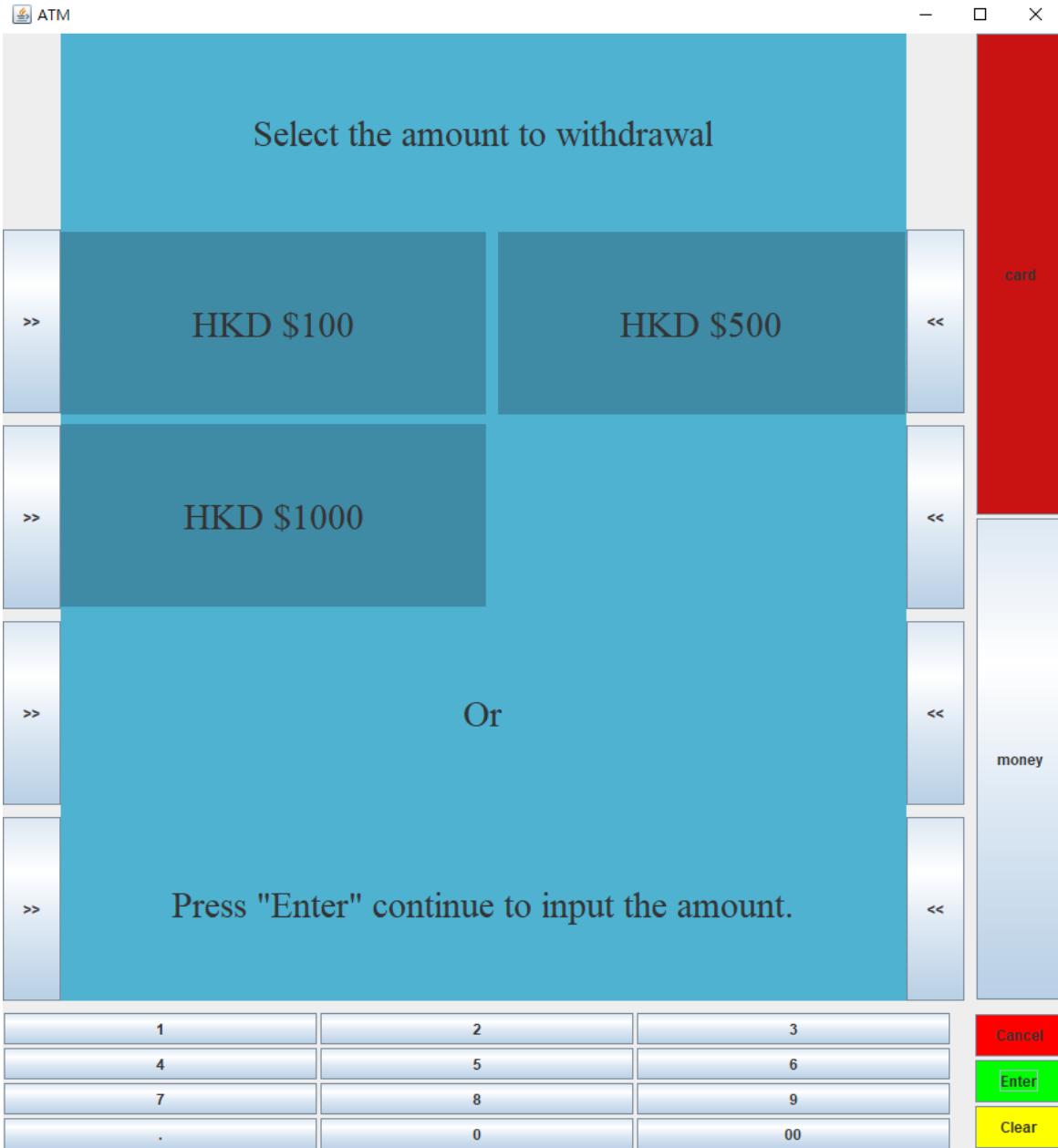
Clear



If user inputs withdrawal amount is larger than available balance. It will occur the error message of noticing user input smaller amount.

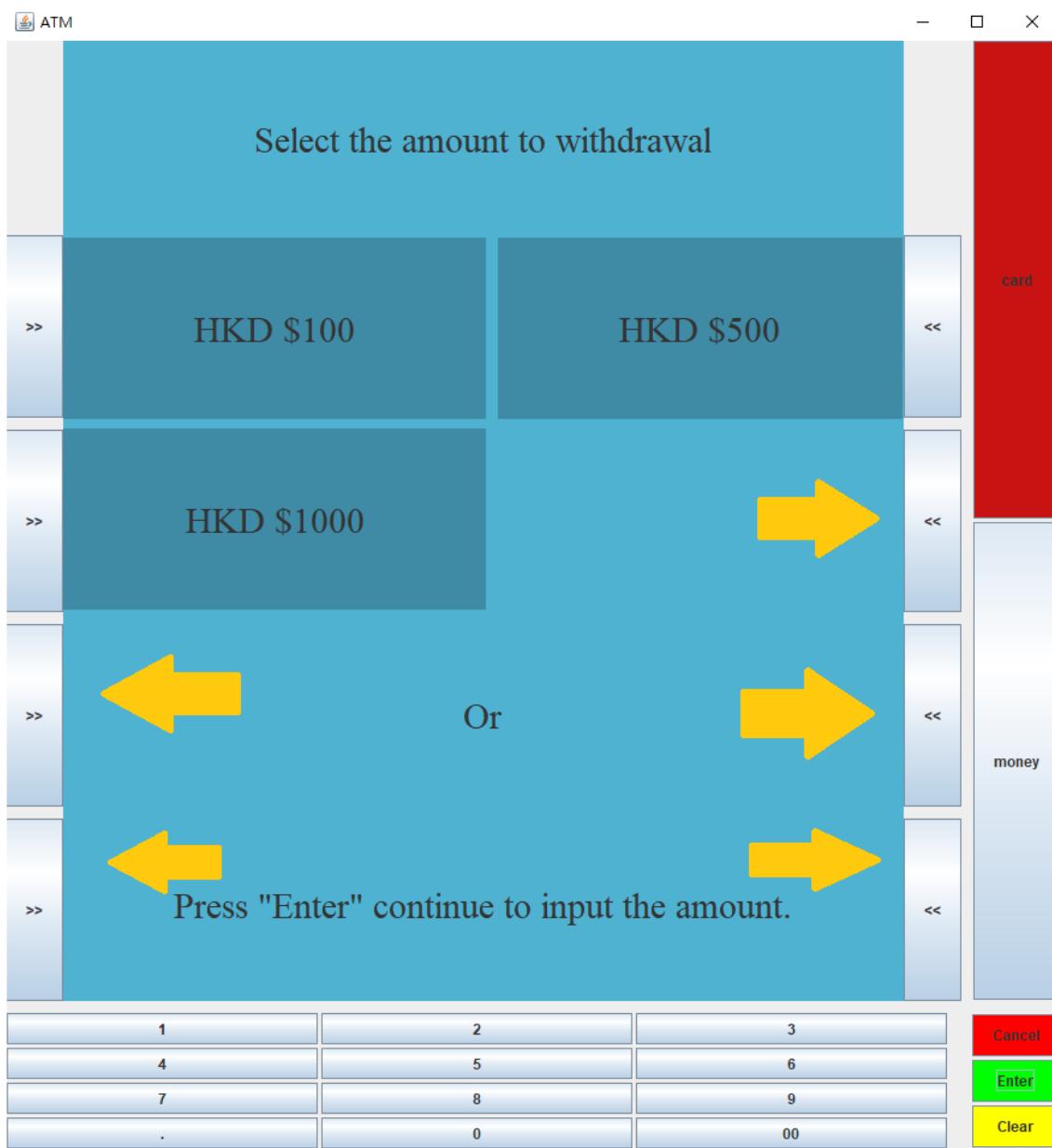
Test case 4: Withdrawal \$0

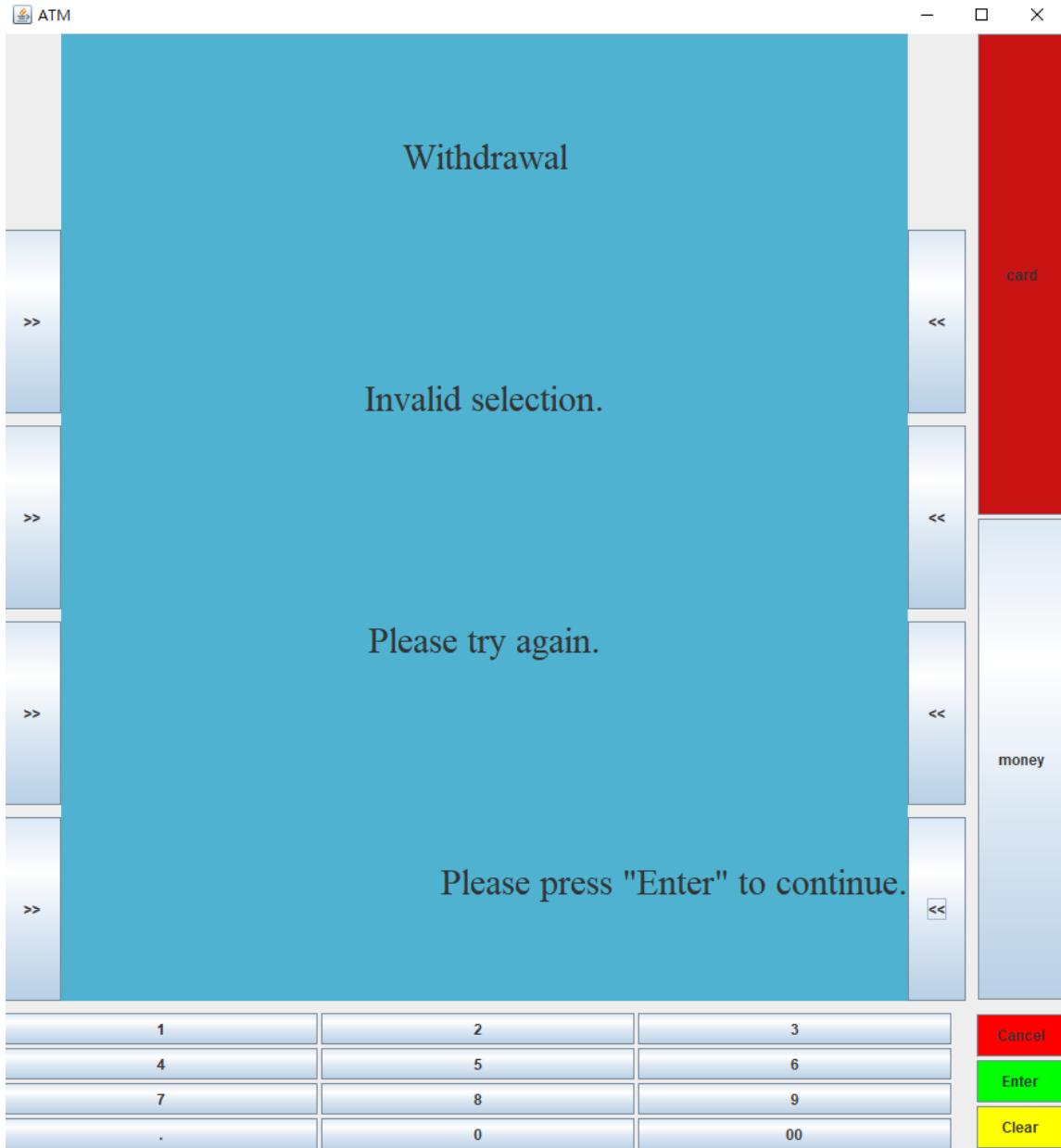




If user input \$0 for withdrawal amount, it will automatically return to previous “Select Amount” page.

Test case 5: Press wrong button on side buttons

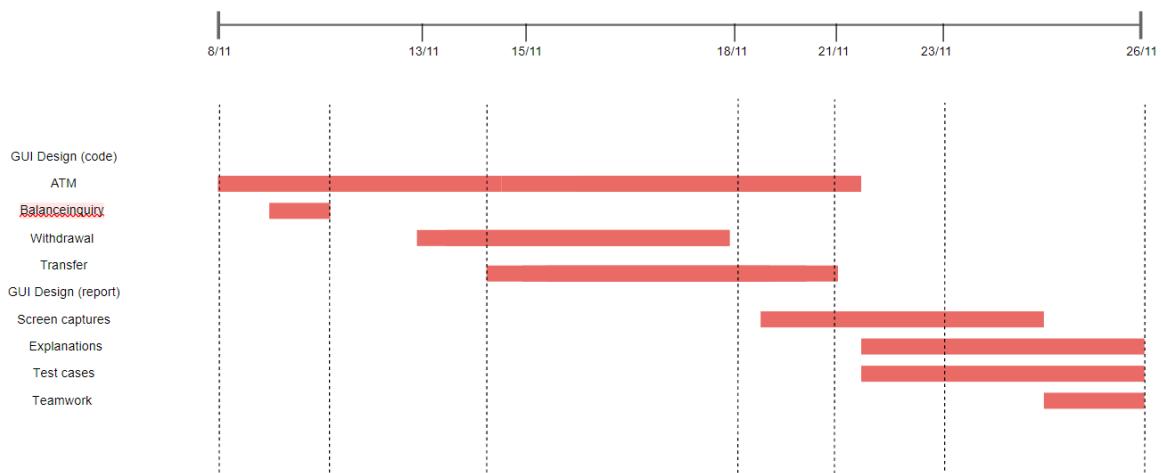




If user press wrong button, it will show the invalid selection message.

# Appendix

Timeline of the work done.



Group learning experience:

After this Java GUI project, we rise the ability of coding GUI. This project required us to design a ATM GUI system. Since this project is exactly related to the last project we did. Before this project, we didn't know the code we made will affect this project. So, we have to study our code we did last month. It is pretty complicated since there are lots of message we write for different situation, which led us to make a lots of GUI method for each situation especially the transfer function since it is the most difficult part from last project. It also becomes our most difficult challenge we face.

Transfer function is the most difficult part we faced. We wrote so many messages in last project, which make us to do around 10 GUI method in this function. It used so much time in this project. Also, the GUI lessons is not finished yet when we were doing this project, so we discovered different kinds of coding that has not been taught in lesson. After doing the GUI, the button is another challenge we faced. Since it is an ATM system so it totally have 23 buttons we have to make. The most complicated button is the "Cancel" button since it set to be press in anywhere with this system, after cancelling some same message must show to user and let them know what is happening. This used most of the time in this project. This also take different reference from lesson and internet to deal with the challenge we faced.

Moreover, the text field is also a complicated in this project. In withdrawal function, we have to let user to input the amount what they want. According to the last project, we did a great job so we can use more time to deal with the text field button. The text field button we used can show the user input amount in next step. We design a String to save the input text in every GUI since there are some amounts and accounts required user to input, so we have to rename many strings and text fields to fulfill the requirement. The text field and string can show the user input and let user know that and confirm it or not. Also, we think a lot of GUI when we design the interface. We must think carefully about the flow of the ATM system, so we really film some videos in real ATM to help us do the project. To pursue the exactly the real ATM system in real life. After all, we can finish the whole project and solve the problems of designing and coding GUI for the last project we made.

One of the external factors is time, there are so many assignments between these weeks. And this project required us have a great teamwork and time to finish. Also, we just learnt the GUI for few lessons. We cannot apply it within these days, so we must pay more time to finish this project. Luckily, we can form the group in last project, and we have a good communication on this project but we still don't have much time on this project.

Lastly, we must think twice before we do the project. If the next projects have relationship with each other. We must think a cautious method to finish the coding. Otherwise, we have to face the complicated coding again when we finish next project.