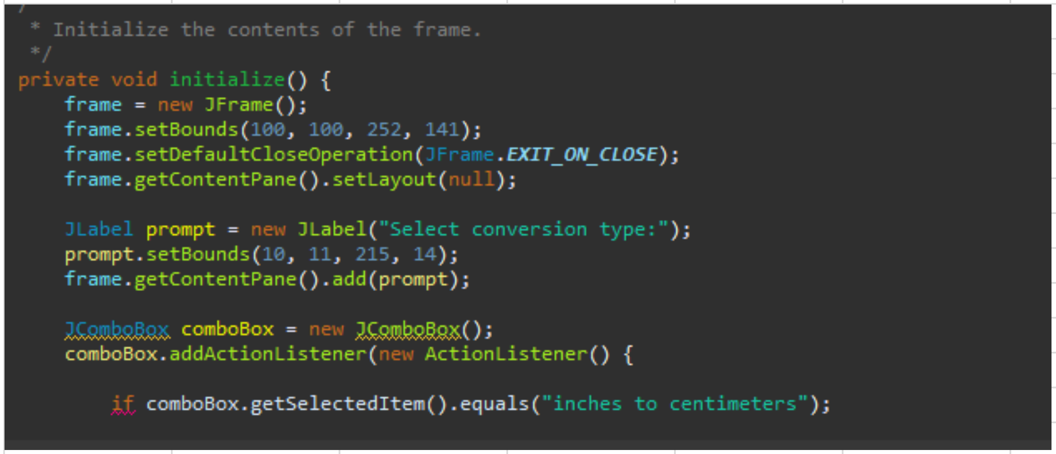
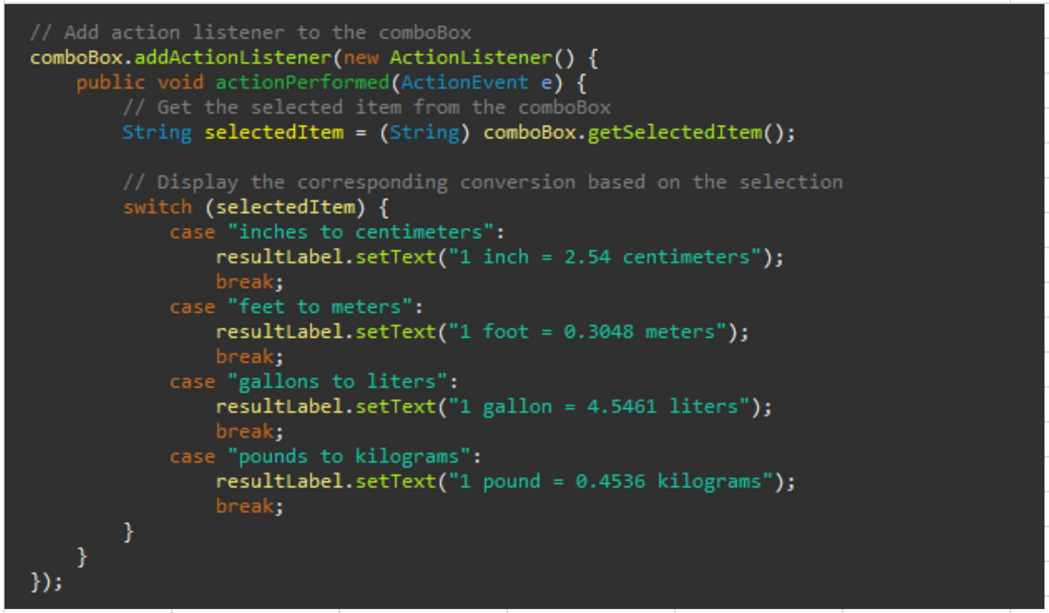
MetricConversion - Errors



```
 * Initialize the contents of the frame.
 */
private void initialize() {
    frame = new JFrame();
    frame.setBounds(100, 100, 252, 141);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel prompt = new JLabel("Select conversion type:");
    prompt.setBounds(10, 11, 215, 14);
    frame.getContentPane().add(prompt);

    JComboBox comboBox = new JComboBox();
    comboBox.addActionListener(new ActionListener() {

        if comboBox.getSelectedItem().equals("inches to centimeters");
```

The first error I made in my code was when I tried to incorporate an if statement to the comboBox without adding an event handler.

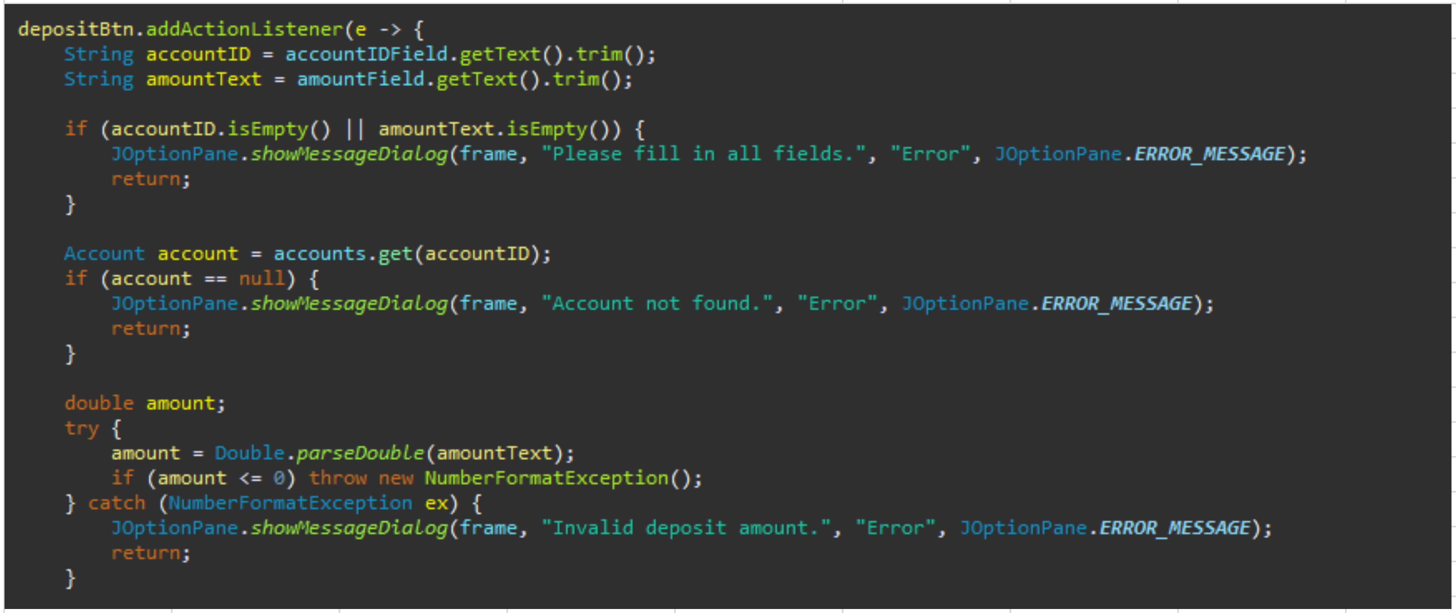
I resolved this error by adding an event handler with a switch statement to display the intended output:



```
// Add action listener to the comboBox
comboBox.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Get the selected item from the comboBox
        String selectedItem = (String) comboBox.getSelectedItem();

        // Display the corresponding conversion based on the selection
        switch (selectedItem) {
            case "inches to centimeters":
                resultLabel.setText("1 inch = 2.54 centimeters");
                break;
            case "feet to meters":
                resultLabel.setText("1 foot = 0.3048 meters");
                break;
            case "gallons to liters":
                resultLabel.setText("1 gallon = 4.5461 liters");
                break;
            case "pounds to kilograms":
                resultLabel.setText("1 pound = 0.4536 kilograms");
                break;
        }
    }
});
```

LocalBank - Errors

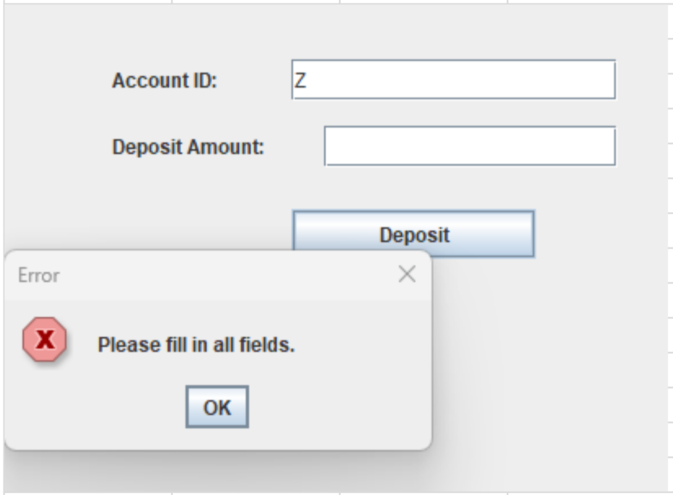
The first error I encountered was while I was testing the deposit feature of the GUI.



```
depositBtn.addActionListener(e -> {
    String accountID = accountIDField.getText().trim();
    String amountText = amountField.getText().trim();

    if (accountID.isEmpty() || amountText.isEmpty()) {
        JOptionPane.showMessageDialog(frame, "Please fill in all fields.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    Account account = accounts.get(accountID);
    if (account == null) {
        JOptionPane.showMessageDialog(frame, "Account not found.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    double amount;
    try {
        amount = Double.parseDouble(amountText);
        if (amount <= 0) throw new NumberFormatException();
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(frame, "Invalid deposit amount.", "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }
```

The issue was that no matter what was inputted, I was prompted with an error message:

Case 1:

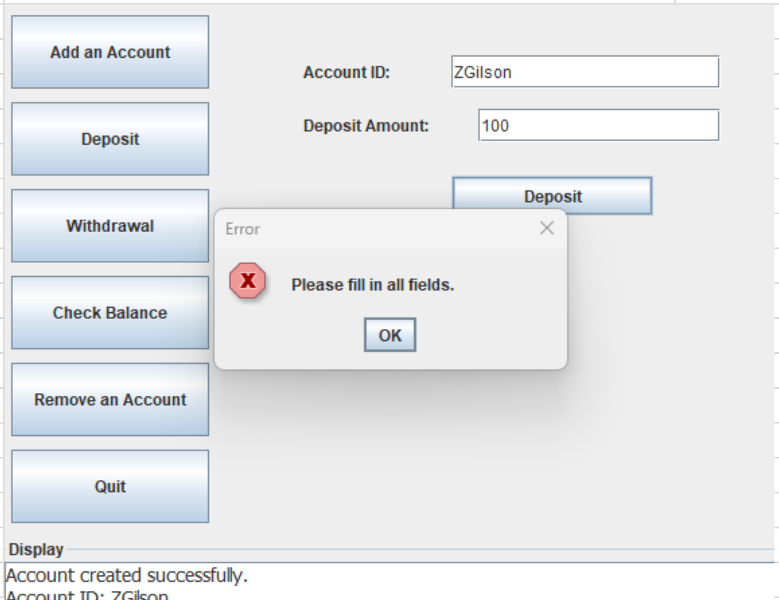


Account ID: Z

Deposit Amount:

Please fill in all fields.

Case 2: (expanded to show proof of account creation)



Account ID: ZGilson

Deposit Amount: 100

Please fill in all fields.

Account created successfully.
Account ID: ZGilson

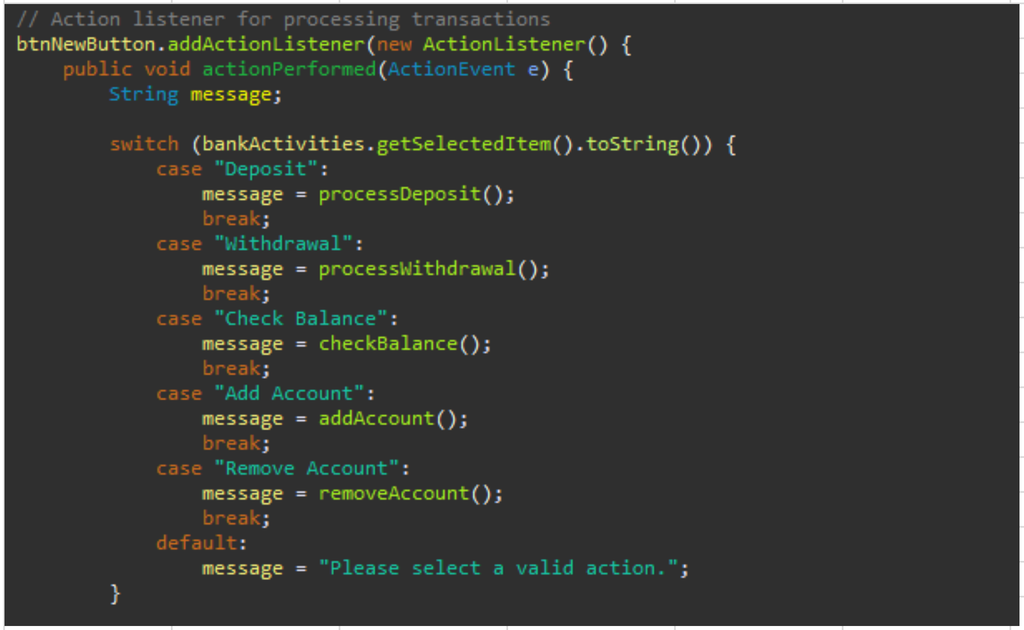
Account ID: ZGilson

In this case, my source of error was that `accountIDField` and `amountField` are declared as class-level variables:

private JTextField firstNameField, lastNameField, accountIDField, amountField;

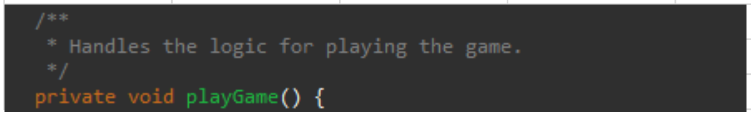
However, these fields are reused across multiple panels, such as the Add Account, Deposit, Withdraw, etc. This reuse led to unintended behavior, in this case,
actions in one panel being clear or modifying the values of accountIDField and amountField, causing them to appear empty when performing operations like deposit or withdrawal.

To resolve this error, I first separated my code into three classes: Account, Bank, and Customer.
Then, I adjusted the functionality of the backend to work seamlessly with the other classes.
This is what the modified action listener looks like:

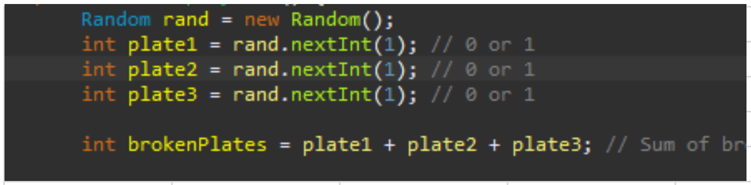


```java
// Action listener for processing transactions
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String message;

        switch (bankActivities.getSelectedItem().toString()) {
            case "Deposit":
                message = processDeposit();
                break;
            case "Withdrawal":
                message = processWithdrawal();
                break;
            case "Check Balance":
                message = checkBalance();
                break;
            case "Add Account":
                message = addAccount();
                break;
            case "Remove Account":
                message = removeAccount();
                break;
            default:
                message = "Please select a valid action.";
        }
```

BreakAPlate - Errors



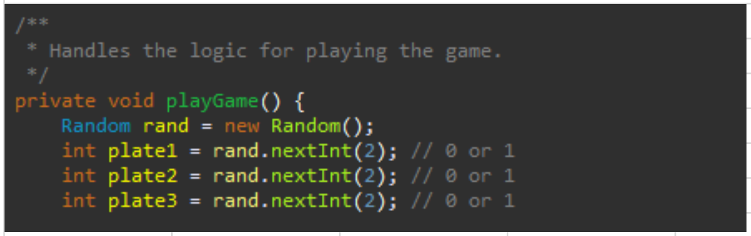
```java
/**
 * Handles the logic for playing the game.
 */
private void playGame() {
```



```
    Random rand = new Random();
    int plate1 = rand.nextInt(1); // 0 or 1
    int plate2 = rand.nextInt(1); // 0 or 1
    int plate3 = rand.nextInt(1); // 0 or 1

    int brokenPlates = plate1 + plate2 + plate3; // Sum of br
```

The one error I made in my process was having my value within the `rand.nextInt(x);` equal to one, when it was supposed to be two.

I mistakenly believed that the above code would choose a number between zero and one, however this is incorrect. The correct code is shown below:



```
/**
 * Handles the logic for playing the game.
 */
private void playGame() {
    Random rand = new Random();
    int plate1 = rand.nextInt(2); // 0 or 1
    int plate2 = rand.nextInt(2); // 0 or 1
    int plate3 = rand.nextInt(2); // 0 or 1
```

I made no other major logic errors throughout my process.