

Project Report:
Critical Thinking 2 - Bank Account GUI

Alejandro Ricciardi

Colorado State University Global

CSC372: Programming 2

Professor: Dr. Vanessa Cooper

June 23, 2024

Project Report:

Critical Thinking 2 - Bank Account GUI

This documentation is part of the Critical Thinking 2 Assignment from CSC372: Programming 2 at Colorado State University Global. This Project Report is an overview of the program's functionality and testing scenarios including console output screenshots. The program is coded in Java JDK-21; and is named Critical Thinking 2 (Bank Account GUI). The program is composed of a Main class, a BankAccount class, a CheckingAccount class that extends the BankAccount, and a FrameBankAccount class that extends the JFrame (swing) class and implements ActionListener interface.

The Assignment Direction:

Option #1: Creating a GUI Bank Balance Application

Create a simple Graphical User Interface (GUI) Bank Balance application. The application must obtain the BankAccount balance from a user, and then display that balance within a JPanel when the user selects a button. The program should allow for the user to deposit and withdraw funds from their account using a simple interface. The remaining balance in the account should be displayed before exiting the program. Ensure that your application includes the following components:

- JPanel
- JButton
- ActionListener

Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

Students must use appropriate version control for all programmatic assignments created. GIT repositories should be established and screen captures of repositories submitted with each assignment.

⚠ My notes:

- This program implements a GUI using the swing library to an improved version of my project [Critical Thinking 1 - Bank Account](#)
- I added my own icon to the window frame – logo.png
- **For the source code please see Main.java, BankAccount.java, CheckingAccoun.java, and FrameBankAccount class files.**

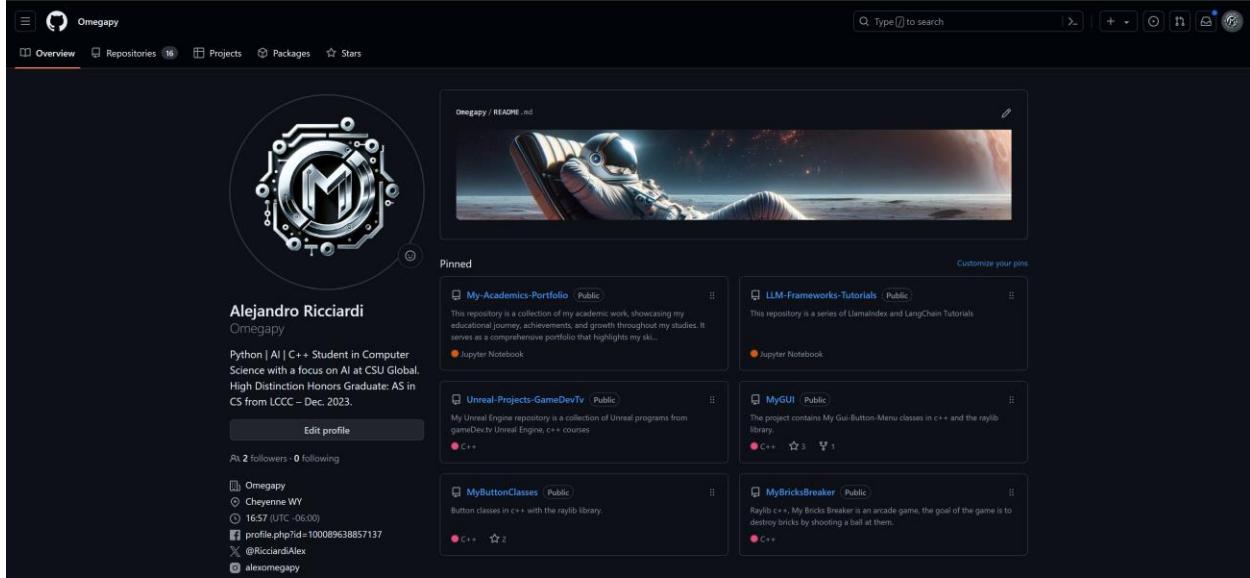
My Program Description:

Bank Account GUI is a simple banking manager system that utilizes the swing library, a graphical user interface (GUI) library.

The program allows users to manage basic bank accounts and checking accounts with various functionalities such as creating accounts, attaching checking accounts, depositing and withdrawing funds, and viewing account balances.

Git Repository

This is a picture of my GitHub page:



I use [GitHub](#) as my Distributed Version Control System (DVCS), the following is a link to my GitHub, [Omegapy](#).

My GitHub repository that is used to store this assignment is named [My-Academics-Portfolio](#) and the link to this specific assignment is:

<https://github.com/Omegapy/My-Academics-Portfolio/tree/main/Programming-2-CSC372/Critical-Thinking-2>

Classes Description:

- **The BankAccount Class:**

The BankAccount class represents a basic bank account with fields for first name, last name, account ID, and balance. It provides methods to deposit and withdraw funds and does not allow cash overdrafts in comparison its extended CheckingAccount class allows with an overdraft fee the checks overdraft.

- **The CheckingAccount Class:**

The CheckingAccount class represents a checking account, it is an extension of the BankAccount class. It includes an interest rate and allows for overdraft withdrawals but it applies an overdraft fee.

- **FrameBankAccount:**

FrameBankAccount class utilizes the swing library to provide a GUI for managing bank accounts. It allows users to use various operations to manage bank accounts. Operations such as adding new bank accounts, attaching checking accounts, depositing, and withdrawing funds, and viewing account balances.

- **The Main Class:**

The Main class initializes and runs the Bank Account GUI application.

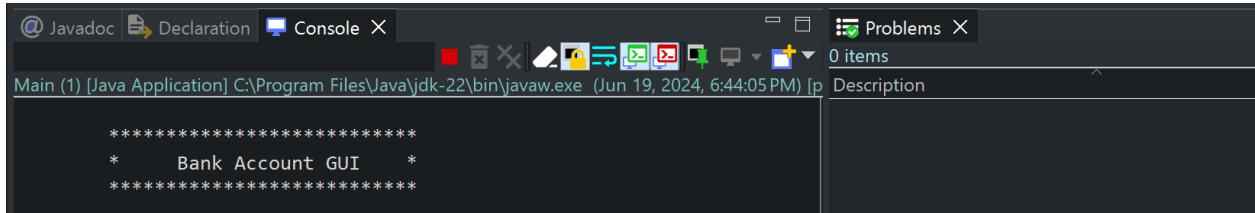
It also populates the system with some fake checking accounts and launches the GUI.

Screenshots

Program Functionality and Testing Scenarios:

Figure 1

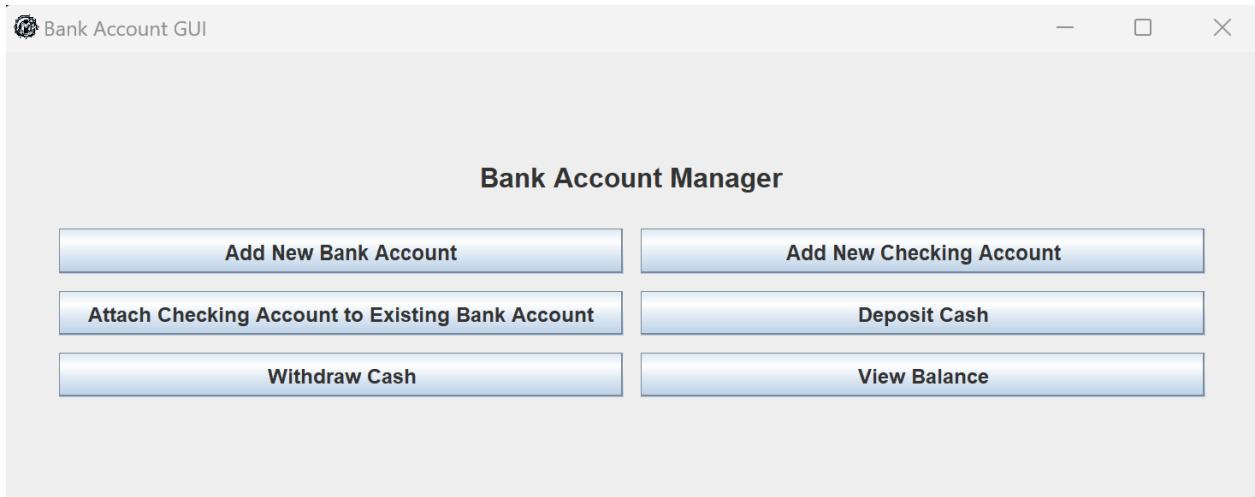
Welcome Screen Console



Note: Eclipse Console output from the program and Problem window showing no items.

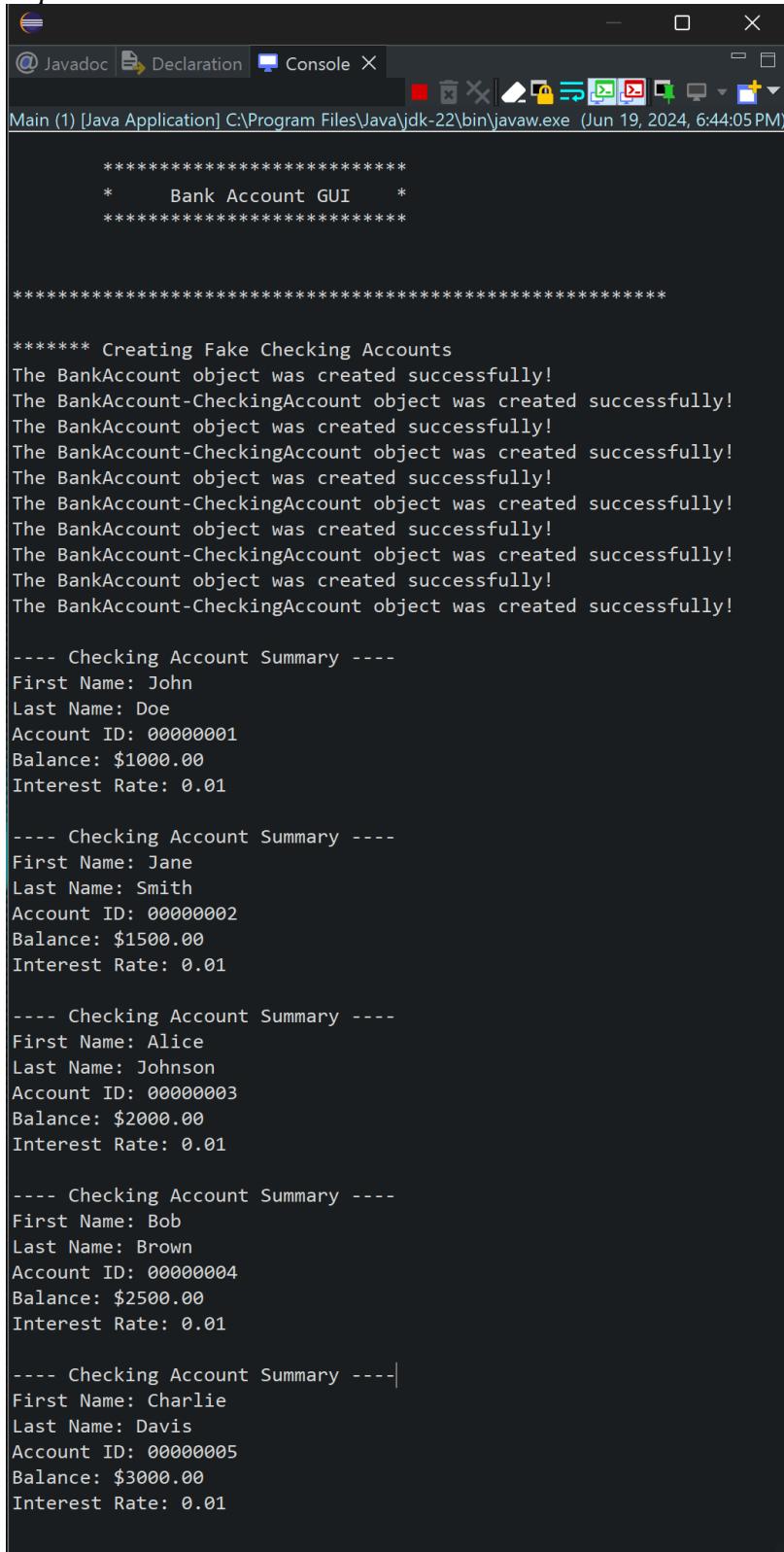
Figure 2

Welcome Screen GUI



More next page

Figure 3
Populate With Fake Data



The screenshot shows a Java application window titled "Main (1) [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (Jun 19, 2024, 6:44:05 PM)". The window has tabs for "Javadoc", "Declaration", and "Console". The console tab is active, displaying the following text:

```
*****
*      Bank Account GUI      *
*****  
  
***** Creating Fake Checking Accounts  
The BankAccount object was created successfully!  
The BankAccount-CheckingAccount object was created successfully!  
The BankAccount object was created successfully!  
The BankAccount-CheckingAccount object was created successfully!  
The BankAccount object was created successfully!  
The BankAccount-CheckingAccount object was created successfully!  
The BankAccount object was created successfully!  
The BankAccount-CheckingAccount object was created successfully!  
The BankAccount object was created successfully!  
The BankAccount-CheckingAccount object was created successfully!  
  
---- Checking Account Summary ----  
First Name: John  
Last Name: Doe  
Account ID: 00000001  
Balance: $1000.00  
Interest Rate: 0.01  
  
---- Checking Account Summary ----  
First Name: Jane  
Last Name: Smith  
Account ID: 00000002  
Balance: $1500.00  
Interest Rate: 0.01  
  
---- Checking Account Summary ----  
First Name: Alice  
Last Name: Johnson  
Account ID: 00000003  
Balance: $2000.00  
Interest Rate: 0.01  
  
---- Checking Account Summary ----  
First Name: Bob  
Last Name: Brown  
Account ID: 00000004  
Balance: $2500.00  
Interest Rate: 0.01  
  
---- Checking Account Summary ----|  
First Name: Charlie  
Last Name: Davis  
Account ID: 00000005  
Balance: $3000.00  
Interest Rate: 0.01
```

Figure 4
Add New Bank Account

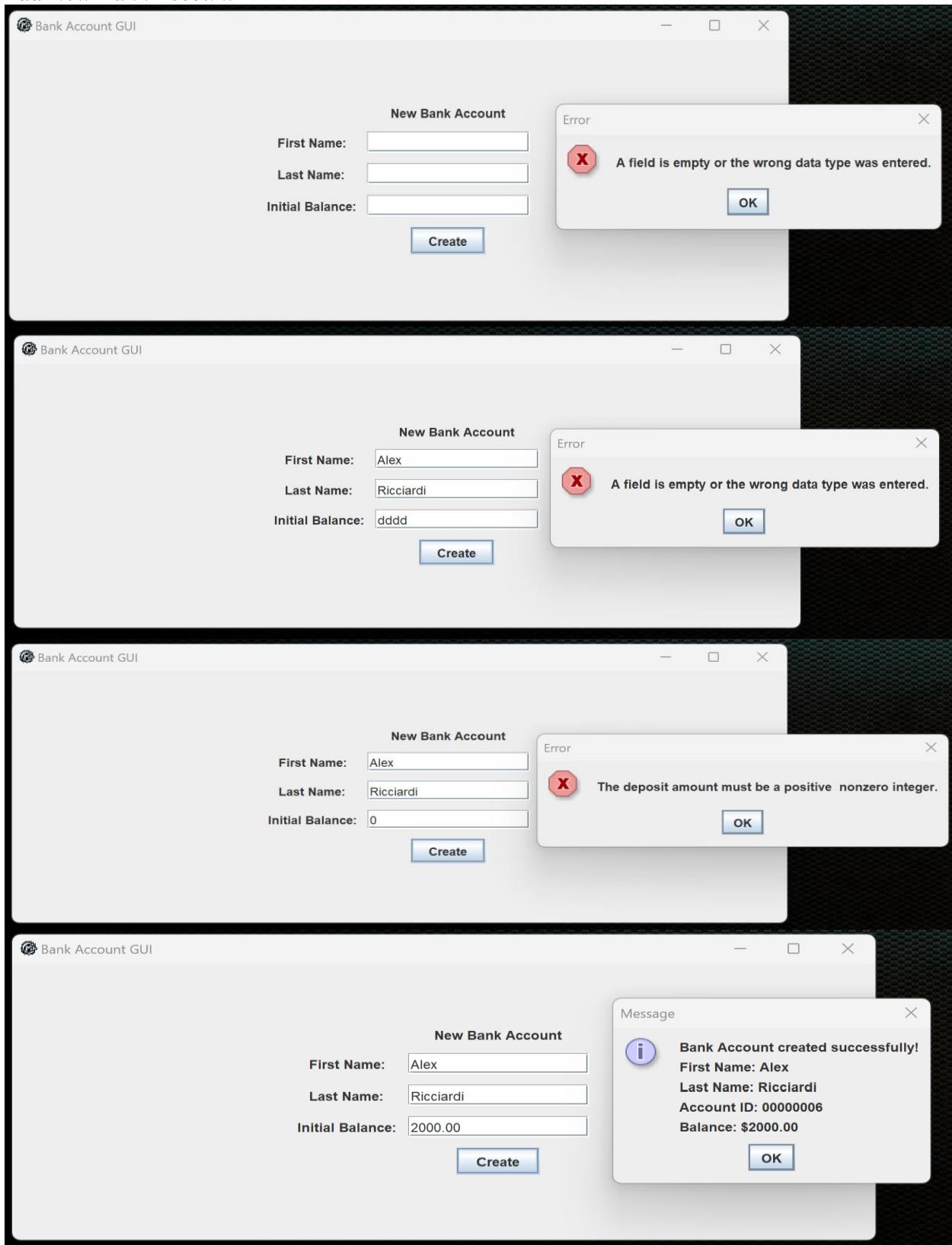


Figure 5
Add New Checking Account

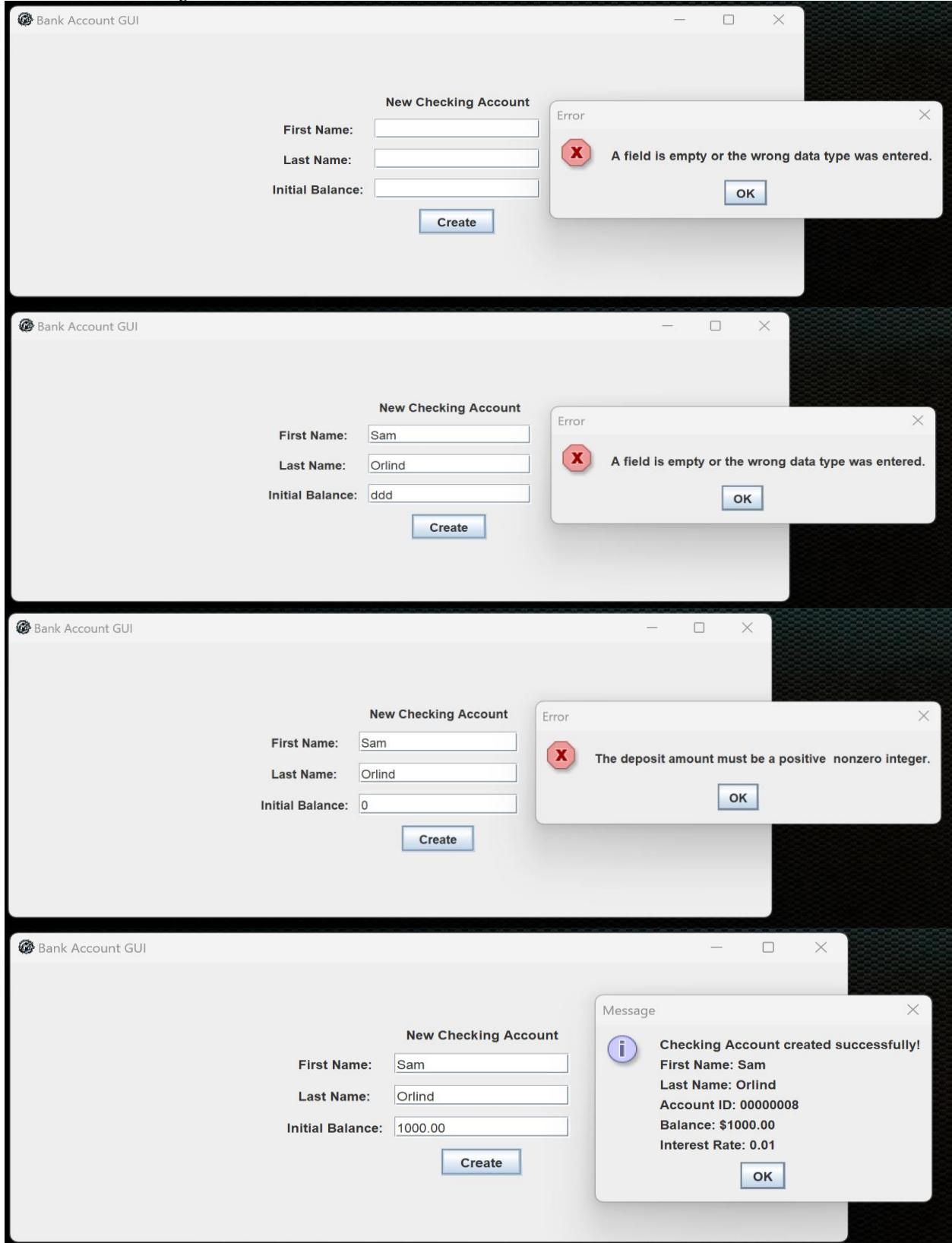


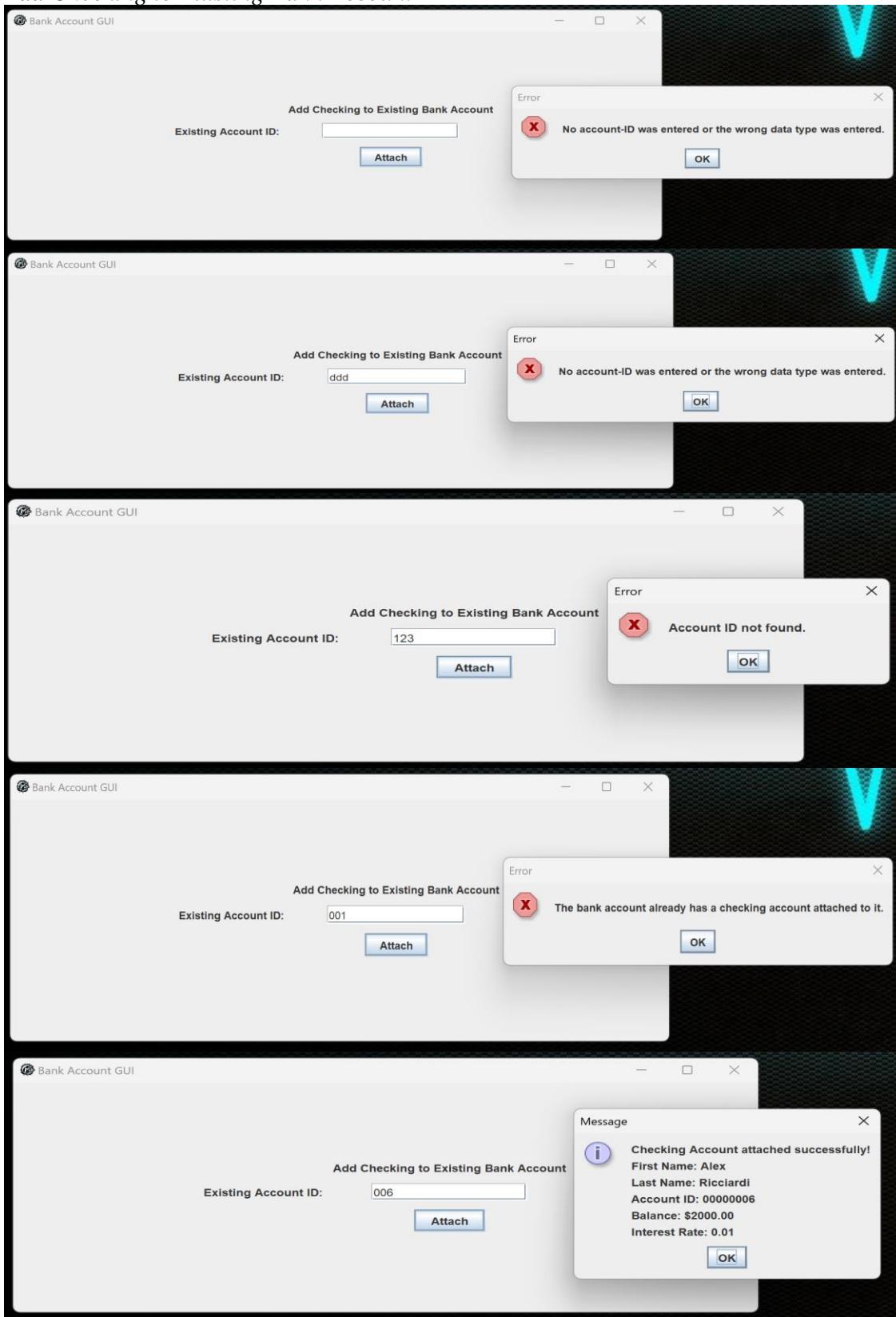
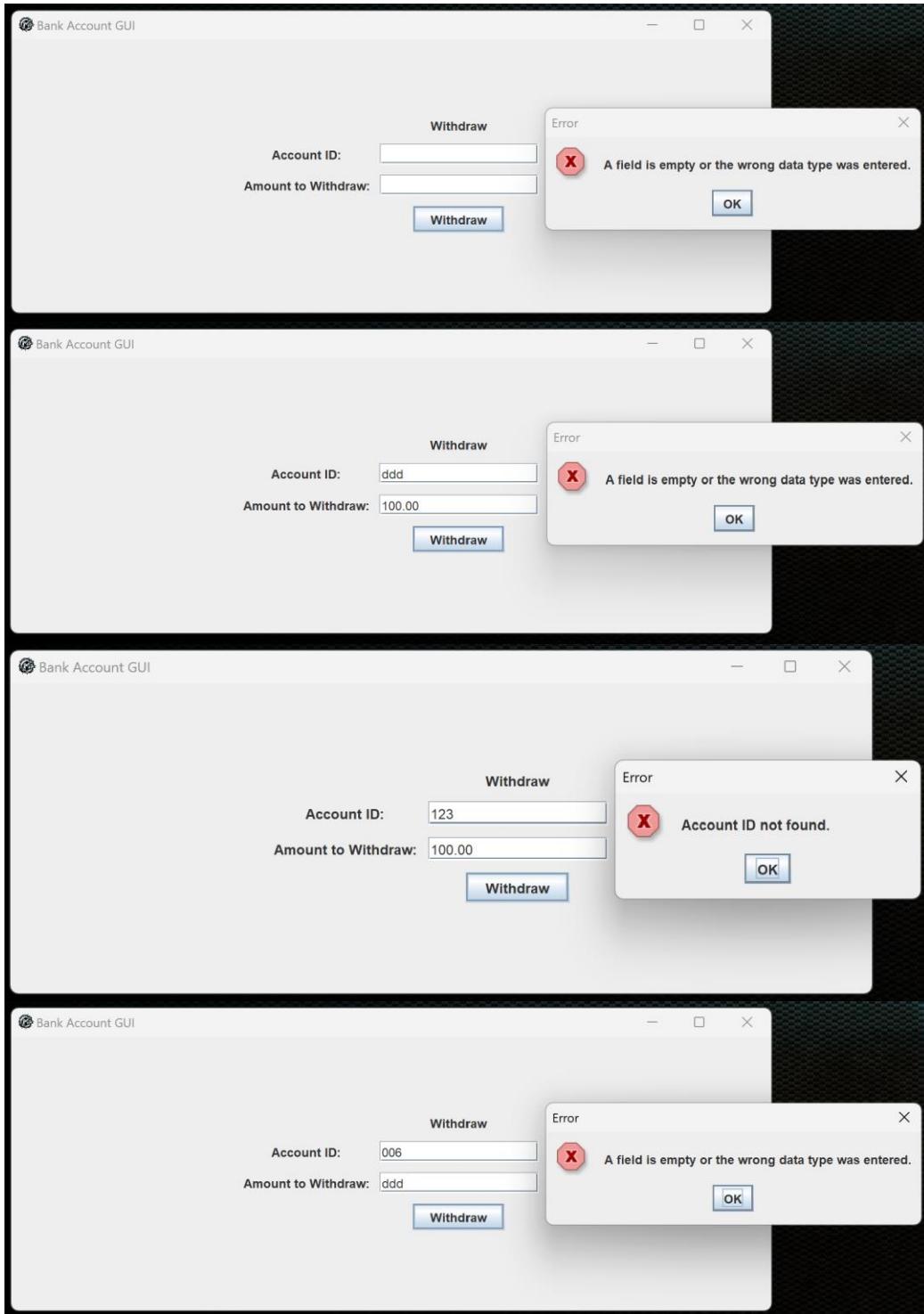
Figure 6*Add Checking to Existing Bank Account*

Figure 7**Withdraw***Continue next page*

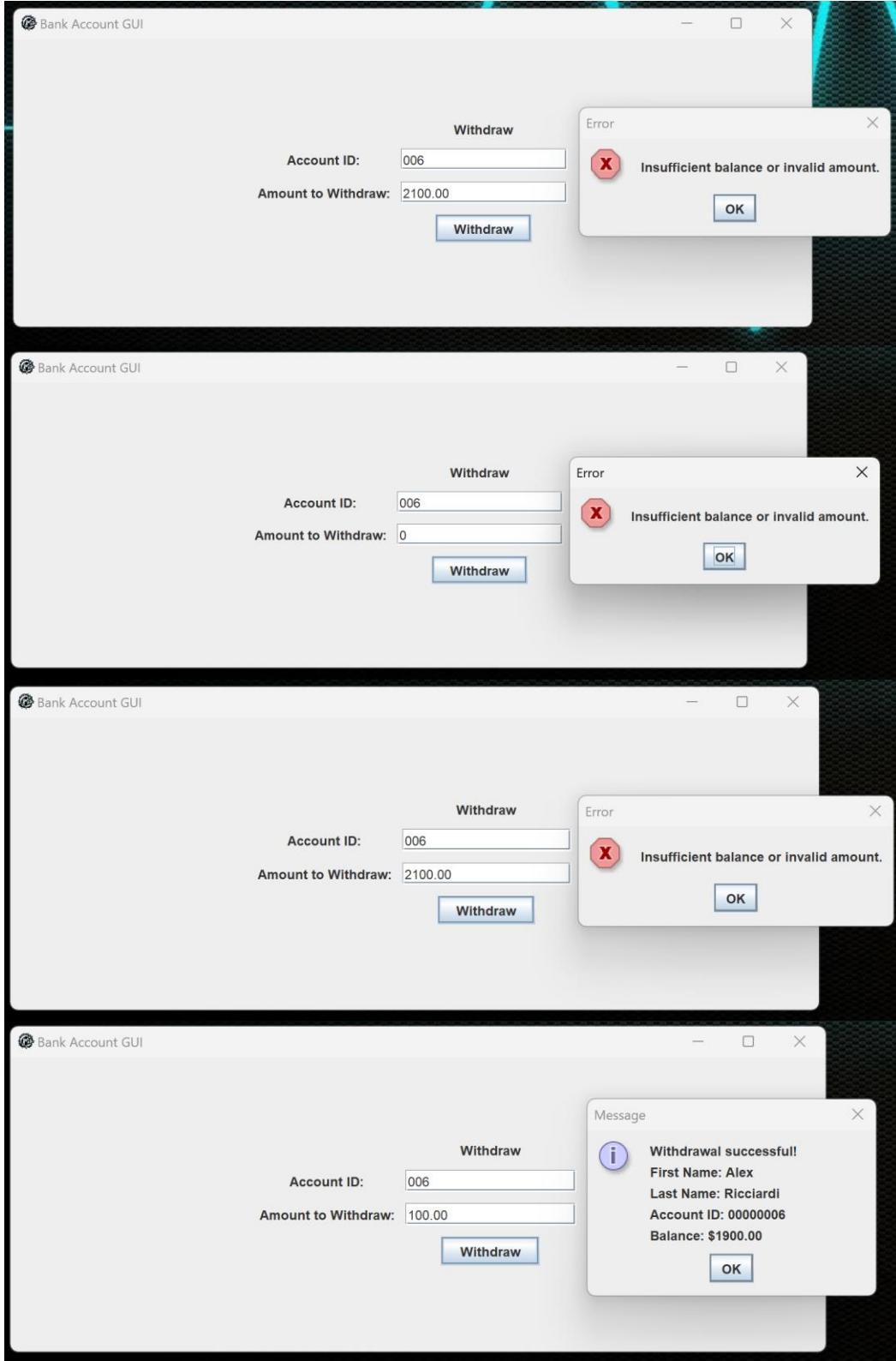
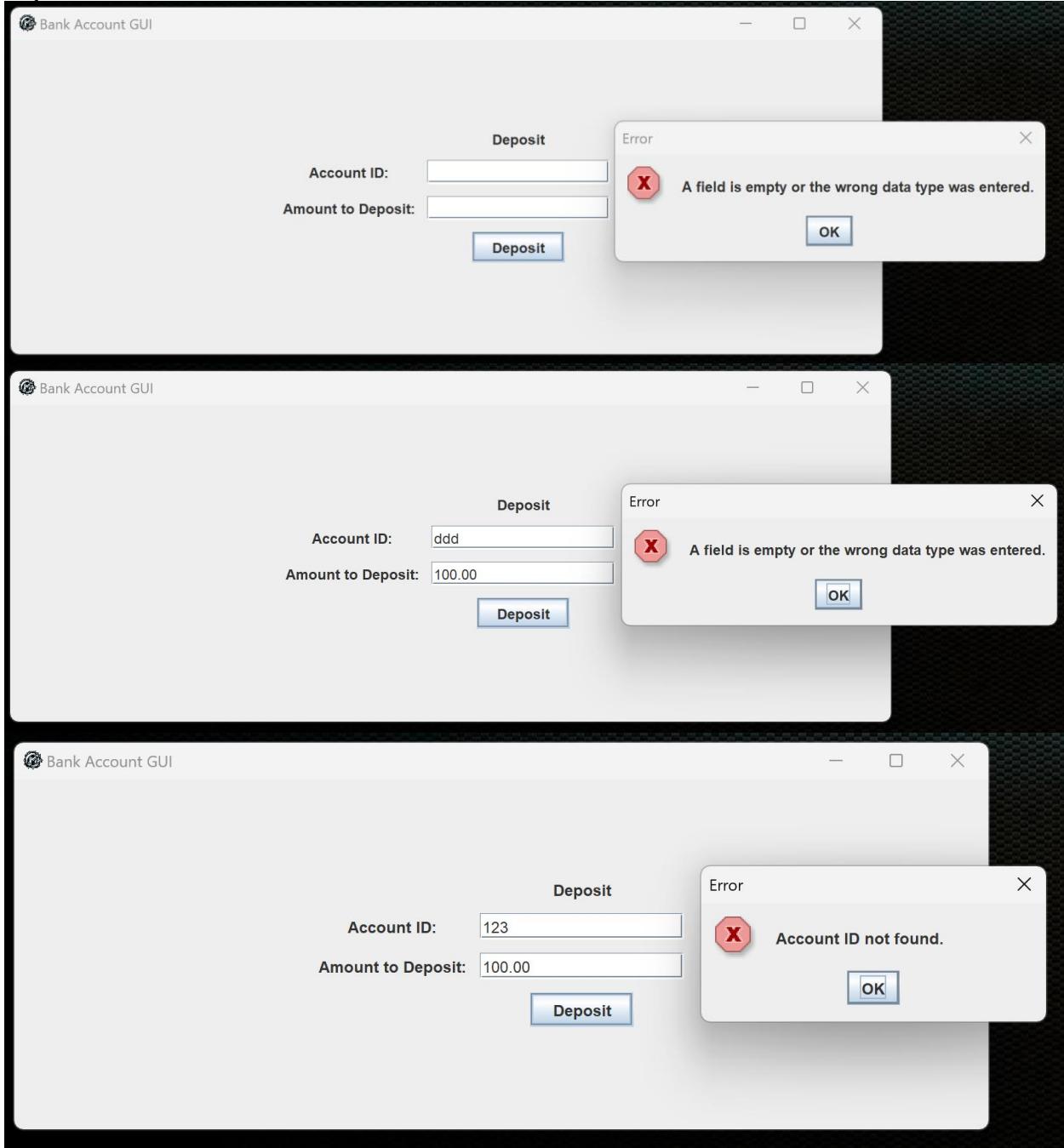
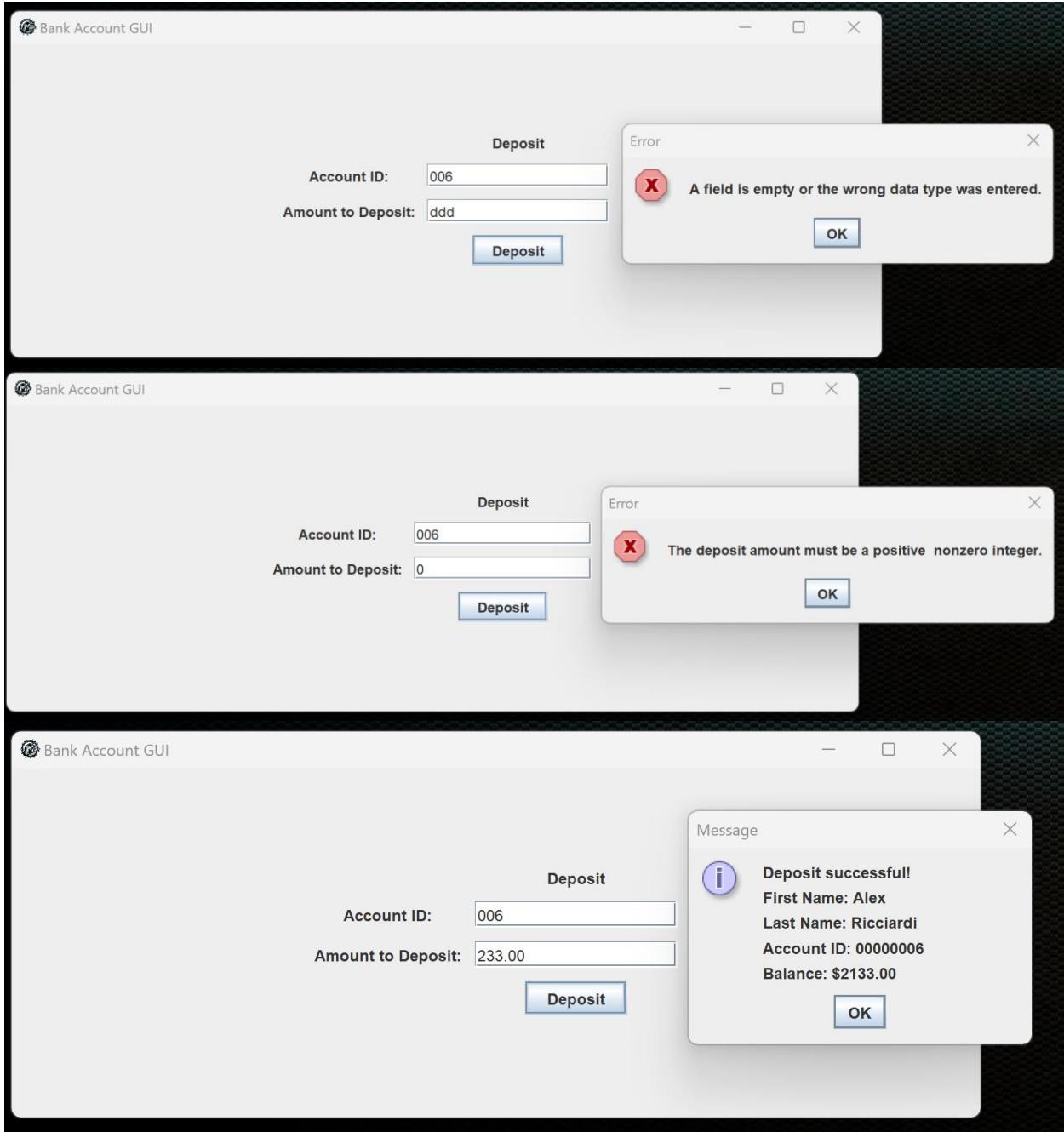
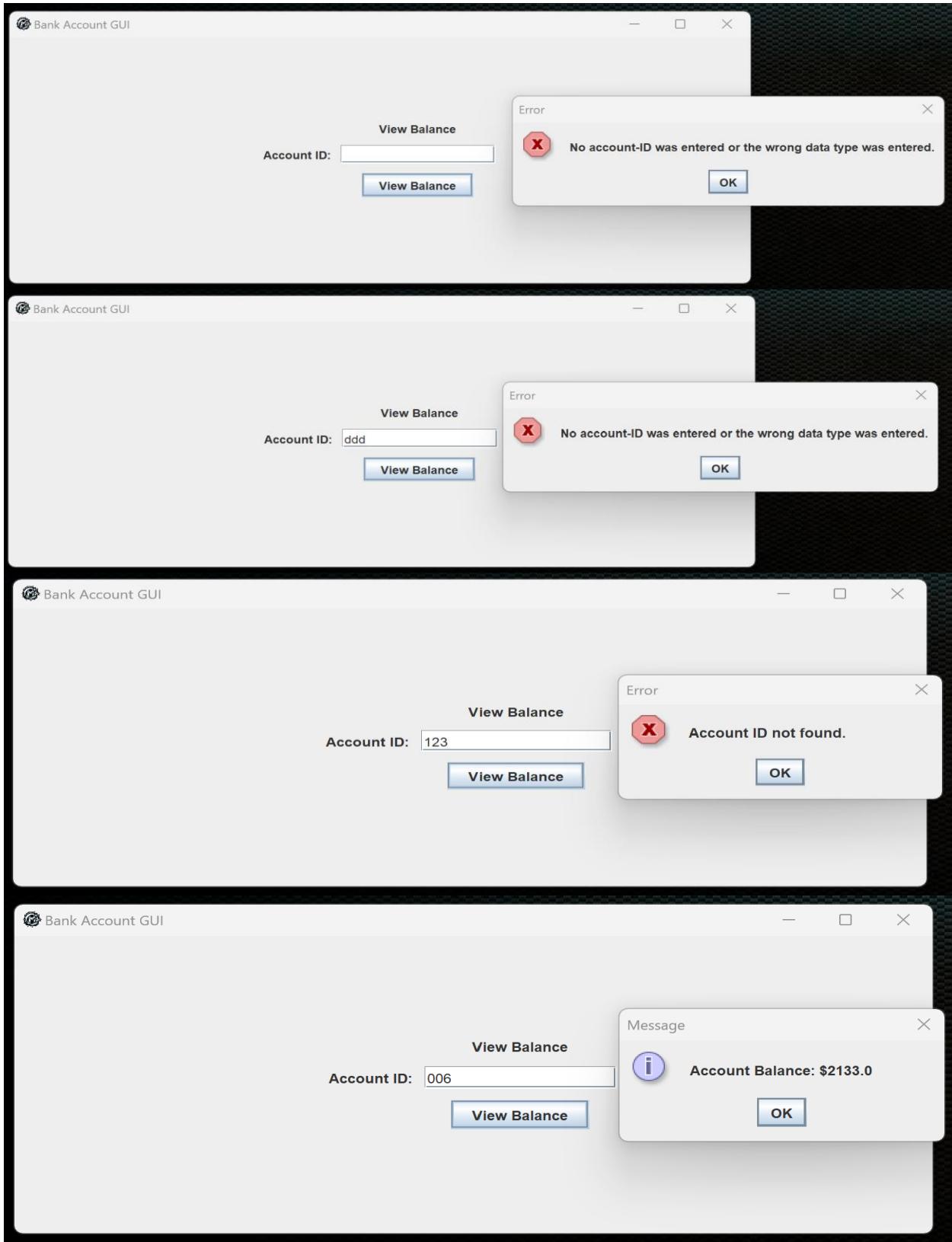


Figure 8*Deposit*



More next page

Figure 9
View balance



As shown in Figure 1 through Figure 9 the functionalities tests run without any issues, displaying the correct outputs as expected.