

This .zip folder contains multiple parts of our project.

Project4UML.png is our final version of our UML class diagram.

Pwpt Presentation_Project4 is the PowerPoint that we showed during our presentation with professor Elham.

In the 'Code' folder, you will find the .java files of our two classes (ATM.java and Account.java). We tried to comment our code as much as possible to show what everything did.

In the 'Runnable' folder, you will find a runnable .jar file of our project (the same one that we showed during our presentation), as well as a couple of .txt files, which are examples of different accounts. Our software creates a .txt file for each user that creates a new account, so it is best to leave the .jar file in a folder so that the text files are created in the same folder as the .jar.

As stated above, when an account is created, it creates a new text file with all of that specific user's information. When signing into an account, that user's file is opened.

Running Instructions:

- ❖ When Running, the software will initialize to a sign-in page, where you can use your username and password to sign in.
 - A successful sign in will open that user's specific file, and dispose of the 'ATM' class/window, and open an 'Account' class/window
 - An unsuccessful sign in will occur if a file with that username cannot be found, or if the password does not match the one in the file. It will cause a red JLabel to appear, explaining that the account could not be accessed
- ❖ If the user does not already have an account to which they can sign-in to, then they can create an account by pressing the button in the lower right hand corner. **Note that while on this panel, the user can select the 'back' button to return to the sign-in page.
 - On the new user panel, the user must enter a first and last name, a desired username, and they must create and confirm a password
 - If any of the above-mentioned fields are left blank, or if the passwords do not match, it will not allow the user to create the account, and will display a message explaining why.
 - If correctly filled out, then the user's account will be created (and his/her corresponding file in the .jar's directory), and the program will dispose of the window, and create a new 'Account' window.
- ❖ On the Account panel, a message will greet the user with their name, and display their standing balance. The user will have the option to withdraw, deposit, change password, and sign-out.
 - Withdraw – a dialogue will open asking for the amount desired, and if a number is entered, it will deduct that amount from the user's balance.
 - Deposit – almost exactly like withdraw, the only difference being that it will add to the user's balance, instead of deducting it.
 - ***Note, even if the user enters an amount such as \$330.123455566, only the digits 330.12 will be displayed, as that is how currency is measured
 - Change Password – A series of 3 dialogue boxes will appear: the first asking for the user's current password.

- If correct, a new dialogue will appear asking for the desired new password. The next dialogue will ask to confirm the new password.
- If any of the following conditions occur, the dialogue boxes will disappear, and the password will not be changed:
 - The user does not enter their old/current password correctly
 - The new password does not match the 'confirm new password'
 - The user presses 'cancel' on any of the dialogue boxes at any time
- ❖ For the most part, the program is fairly intuitive, and easy for a user to understand. The only major issue I am aware of is that a user can create a new account, and if they use a username that is already associated with an existing account, it will overwrite the existing user's account.