Overview: As a whole our group did a lot of planning and thinking on how to tackle this project. We figured out specifics of each class and started to think really hard about how each function might work, as well as how to define each user and more. There's a lot that we tackled that might not be directly represented in the code YET which I hope is taken into consideration considering we have to submit the code we tackled for this assignment. We did a lot of this planning so that when we did seriously code, the understanding of the program we are creating as well as how to code it is at least somewhat clear in all of our minds.

Daniel Rong

Problem solving: salting and hashing a password
What I've learned: I learned about how each user gets assigned a salt that is stored in the database as plaintext. This way, multiple users can have the same password, but the hashed password that is also stored as plaintext will look different.
Code instructions: type "python .\DannyTesting.py"

Lily Le

Problem solving: I will be trying to solve the random password generator that is suggested if the user does not have an adequate password.
What I've learned: I figured out that the random password needs to be created randomly with random numbers, letters, and symbols. After gathering all those random characters, the program should shuffle it to create a unique random password.
Code instructions: There are no code instructions for the password.py code for now. It should just be able to run.

Jordan Bell

Problem solving: I started to think about and work on how a text interface and login would work. I started working with the input function to display a message that requires an input as an answer. For now it will always come out as an incorrect username and password because it is not hooked up with the database we are working on.

What I've learned: I learned how the input function worked within python and how it can be used to create a text interface. I also had to think about and work on how to continuously prompt users if their information is correct.

Code instructions: Go to the banking.py file and type in (python banking.py startinguserdata.csv). Once that is typed in you will be prompted for a username so type in either a correct or incorrect username based on the data in the csv file. The file startinguserdata.csv contains a list of existing accounts with usernames and passwords that can hopefully be used for this login function. The function won't do anything beyond telling you if the login is successful or not. I am still working on getting this function to prompt me with the login

correctly so the database is hardcoded in for now so it will work(and this makes sense anyway because we would have a singular database of users).