Credit Name: CSE2130 File Structures & Exception Handling
Assignment Name: MySavings

How has your program changed from planning to coding to now? Please explain?

I actually entirely forgot about the fact that this is in CSE2130 File Structures when first writing the program, and wrote it as an entirely self-contained GUI/OOP program.
This was easily remedied by adding the read() and write() methods to bank, and working them into the rest of the program

Another big change to the file management side was making the files look nicer
It was previously just 4 lines each containing a number, but it was ugly and impossible to read unless you knew how the code works
I fixed this by adding the decorative title lines, skipping over them when reading and rewriting them when writing

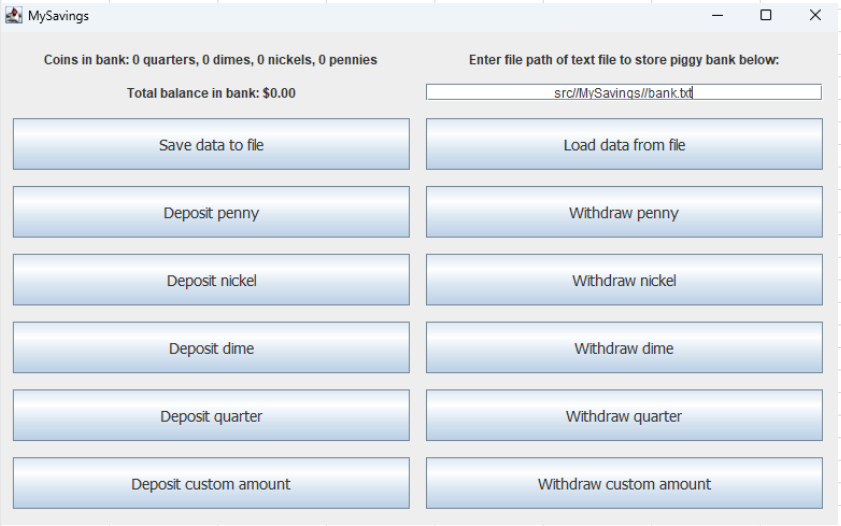
Before

```
9
0
6
4
```

After

```
Pennies
9
Nickels
0
Dimes
6
Quarters
4
```

As mentioned in the error log, I changed the way that one of the main methods (withdraw()) takes and uses data. This was to solve a tricky internal issue that worked with the variables but not logically



An update since I wrote this part:
I changed the top and bottom rows of buttons to pure white instead of shaded blue
I felt it just broke up the UI, making it easier to read and use
All the buttons with special functions stand out from the deposit/withdraw single coin buttons
I feel that it helps somewhat, but I still don't really like the number of buttons

I really hate the absurd number of buttons. The UI feels cluttered and hard to use.
I could easily maintain the same functionality by changing it to just the custom in/out buttons, but I feel that it would violate the piggy bank theme
The entire purpose of a piggy bank is that you put coins in one at a time, and I don't want to throw that away
I don't really have a solution (the only one I can think of is by nesting JOptionPanes or windows to break it up into Save/Load above Deposit/Withdraw with submenus for all the options, but it seems needlessly complicated and not worth the work it would take to rebuild the program)
Just something I'd keep in mind if I was writing a similar program in the future

I didn't enjoy this program as much as the other masteries in this credit, but it wasn't bad or painful- I enjoyed the others quite a bit, and this one I am mostly neutral towards
This program was just a lot more repetitive and tedious. I found the other programs being based around a single complex method of processing the information to be much more interesting than just adding and subtracting numbers

Update:
I realized after finishing the project and reviewing the textbook that the intention was for me to use object serialization to store the bank object
I copied the package and wrote an updated version that uses a serialized object instead of the strings in the file
The read() and write() methods moved to MySavings.java as it was not possible to set the object to the serialized object using an object method, or I failed to find out how

These methods also shortened considerably														
It was quite quick to adapt, and I definitely see the appeal of using serialized objects. The file is no longer nice and readable, but the code itself is much simpler and more efficient.														