What does a good answer to this assignment look like?

There are several parts to a good answer. The following are EXAMPLES ONLY of what a good answer might look like, and are not guaranteed to be worth 100%. You can choose how to organize your own answers and files in any way you think appropriate, as long as it is clear to the markers that you have addressed all the parts asked for in the assignment handout.

- The first part of a good answer is a table of test cases listing the names of the tests, and which feature and requirement they are intended to test, something like this:

Transaction	**Test Name**	**Intention**
login	login1	Test that login transaction is accepted.
	login2	Test that no other transaction is accepted before login.
	login3	Test that we can log in as standard.
	login4	Test that we can log in as admin.
	... and so on ...	

- The second part of a good answer is a printout of the actual test inputs and expected outputs. These could be included in the table also if desired, or printed out separately. Test inputs must be entire sessions. It's important that tests be in separate files.

login1 input file login1.inp

```
login
admin
logout
```

login1 expected bank account transaction file output login1.etf

```
(no real transactions, so event transaction file has only ending line)

00                      00000 00000.00
```

login1 expected terminal output login1.bto

```
welcome to the banking system
enter session type:
login accepted
session terminated
```

... and so on for all other tests ...

- The third part of a good answer is a description of the test file organization and plan to run the tests. This answer is an example only - you may wish to use directories to organize your test input and output files instead of using only one as is described here.

Test Organization
Tests are organized into a single directory organized by file name.
Tests for each transaction are named starting with the transaction name
and numbered sequentially, for example login1, login2, login3 and so on.

Test inputs are named ending in ".inp", for example "login1.inp".
Corresponding expected bank account Transaction File outputs are named with
the same name ending in ".etf", for example "login1.etf". If there is
expected terminal output the corresponding file ends with ".bto", for
example "login1.bto". When tests are run, actual test run outputs for
the bank account Transaction File will be named ending in ".atf"
(e.g. "input1.atf") in the same directory.

... and so on ...

Test Run Plan
Tests will be run using a DOS batch command script that runs the Front End
for each of the test input files in the directory. After each is run, the
script will save the output file and compare it to the expected output file,
reporting if there is a difference.

... and so on ...

- The last part of a good answer is a list of the problems or oversights that you found in the written requirements document (the project desc ription), and what you assumed to deal with them.

Requirements Problems
The requirements do not say what is the maximum number of bank accounts
that a account holder can have. We assume that
the maximum number is (10)

... and so on
...