Include a document (in pdf format) describing your results:

○ Any changes that you made to the proposed design and how they continue to support the requirements

The Merkle Tree that was specified in the design was modified so that the hash of the account balances is included as leaf node. This was done to meet the design document requirement, that the computation of hash should include the account balance and the transactions.

Each block is created once all the info required is available, and all contents written to it are final. This implementation is similar to bitcoin, as no machine is actually writing to the blockchain as the transactions come in. Instead, they compete and only one winner writes all the pending transactions to the blockchain, and receives all the associated transaction fees.

Another test named TestInteractive was created, that allows interactive inputs of the commands. With this test, users can type in commands line by line for the ledger to process. This allows for testing specific commands line by line on the same active ledger, unlike the file input method where once all the transactions are processed the program exits and the ledger is erased.

○ Did the design document help with the implementation?

The design document was very helpful with the implementation. It specified all the required classes and methods, and how to organize them together. The document makes it so that anyone that follows it would have created the same classes and functions, with the same expected outputs.

○ How could the design have been better, more straightforward, or made the implementation easier?

The design could be improved by writing the ledger to file, so that it has persistence and can be read in by another machine to resume transactions.