Credit Card Extension Report

Saul van der Vies & Noël Keijzer

August 31st 2017

Total time spent: (combined)

Category 1: 7 hours 55 minutes
- Writing methods to request a new credit card(Includes altering the database and figuring out how to add the credit cards to the system).
  3 hours 20 minutes
- Writing methods to process credit card payments.
  1 hour 20 minutes
- Changing existing methods to support the new credit cards(closeAccount, getBalance)
  1 hour
- Writing methods to refill credit cards.
  1 hour
- Writing methods to replace/(un)block credit cards.
  45 minutes
- Writing methods to remove credit cards
  30 minutes

Category 2: 5 hours
- Writing test methods for credit card transactions and debugging them
  2 hours
- Writing tests for adding new credit cards to the system and debugging it
  1 hour 30 minutes
- Writing tests for (un)blocking/invalidating and fixing bugs in these chains
  50 minutes
- Writing test methods for refill credit card system and debugging it
  40 minutes

Category 3: negligible

We added two new tables to our database, one table to track the credit card transactions and one table that contained the credit cards. We chose to add the credit card handling to the pin service as the pin service was already responsible for keeping track of pin cards and validating pin requests. The fact that a transaction was made on credit meant that a credit card payment could not follow the normal path through our system(traversing the transactionDispatchService) as the balance would already be subtracted from the credit card. We solved this problem by keeping track of the credit on the card in the credit card table, and adding a table that keeps track of the credit card transaction history(because regular transaction history is tracked through transactionDispatch and transactionReceive). To close an account we added a check to the api that checks if the accountNumber ends with “S” or “C” and if it does the correct handler method is called, in essence the close account system stayed the same but a close credit card branch was added. For the refilling of credit cards we used the already existant system that processed interest and added methods to refill the credit cards after processing the interest of that

month. For the getBalance method we added an extra request to the getBalance chain that requests the balance of the active card for a given accountNumber, and if it is present in the response this will be added to the reply. Unblocking credit cards makes use of the same system as the already existing pin card blocking(incorrect_attempts column in the credit card table). Invalidate card is handled by deactivating a card through an isActive column in the credit card table. Figuring out how to insert credit cards into the system correctly and accounting for all the requirements above took some experimenting which is why implementing the new card functionality took significantly more time than the other methods. This was purely because this was the first method that we implemented. We did not use new libraries.