Differences Between Original and Final Designs

In our finalized class diagram we have a total of ten different classes where in our original class diagram we had seven classes. The first “class” added was the main file which acts as the view and contains the necessary functions to properly initiate the program. The way that the objects would interact with the user was not anticipated during the initial design, so this class had to be added.

The User class and its subclasses are mainly the same. The login functions are now overridden by the individual subclasses so that that functionality is tailored to subclass needs (Customer must make sure the customer is active to allow them to login). A user_type was added to keep track of the user type. Some other smaller functions were also added to make things easier (get_customer, get_all_customers) and added activate_customer to allow admins to reactivate deactivated customers. The attributes were also changed to peewee fields so that they can relate to the database fields.

The account class also has slight differences between the original and finalized class diagrams. An account_type attribute was added to allow the account’s type to be identified. The Savings_Account and Checking_Account both still inherit from Account. The transfer function had the biggest overhaul. The transfer function was broken up into send and receive functions to allow each account to individually edit its own value. A get_account function was also added to Account to retrieve an account based on an account number. This function is also overridden by its subclasses to allow them to check their individual tables. The attributes were also changed to peewee fields so that they can relate to the database fields.

The Log class in the original diagram was renamed to Transaction to better fit the actions it was going to perform. This class creates the different transaction log entries that are created when money is deposited, withdrawn, or transferred. This is a change from the String array based log that was previously defined to better relate the python classes and the database. Now the Transaction class relates to the owner of a transaction and the time of the transaction, rather than referring to the start and end time of a group of transactions. The same functionality that was originally envisioned can be replicated with an array of these new Transaction objects. The new object also contains functions to create new types of transactions based on the different interactions that require a transaction log. The attributes were also changed to peewee fields so that they can relate to the database fields.

The bank system driver driver is the go between (controller) from the command line (view) to the other classes (models). This is what the main function would refer to after it was implemented and the driver would call the necessary classes needed to perform the actions the user wanted to perform. This was not in the original diagram because we did not take into account that we would need a go between driver function to handle functionality between the command line and classes. All of its functions either retrieve information from or make modifications to the database.

The final class that is represented is the database model class. This class is inherited by the rest of the models in this diagram. It contains the information about which database to connect to. It inherits from the model class defined by peewee.