**Description of the Intention of Each Method and Class**

**Akira Aida - 100526064**

**Kathryn McKay - 100524201**

**Alexander Wheadon - 100514985**

|  |  |
| --- | --- |
| **Method or Class** | **Description** |
| class Account | Stores information about a customer's account's balance, etc. |
| Account() | Constructor |
| std::map<std::string, std::vector<Account\*> > Parse(const char\* fpath) | Convert from file to accounts map. |
| int QueryAccountStatus(Account\* account) | Returns one of AccountStatus: Active, disabled, etc. Where a deleted account state takes precedence. |
| std::string GetErrorMessage(int status) | Returns the message corresponding to an account status. That means error messages except for when status == kActiveAccount, in which case it's just a  statement that the account is active. |
| class Commands | Bank System storing, tracking, and querying the accounts. The commands the user and admin use during runtime. |
| Commands() | Constructor |
| void SetAccounts(std::map<std::string, std::vector<Account\*> >&& accounts) | Gives map created from the Parse method to this class. |
| void login() | Corresponds to 'login' command. |
| void withdrawal() | Corresponds to 'withdrawal' command. |
| void transfer() | Corresponds to 'transfer' command. |
| void paybill() | Corresponds to 'paybill' command. |
| void deposit() | Corresponds to 'deposit' command. |
| void create() | Corresponds to 'create' command. Admin only. |
| void delete\_account() | Corresponds to 'delete' command. Admin only. |
| void disable() | Corresponds to 'disable' command. Admin only. |
| void changeplan() | Corresponds to 'changeplan' command. Admin only. |
| void logout() | Corresponds to 'logout' command. Causes output to the transaction file. |
| void enable() | Corresponds to 'enable' command. Admin only. |
| bool UserExists(std::string name) | Check if user exists |
| bool CheckUnit(double amount) | Checks if 'amount' is valid currency. |
| std::string DetermineSession() | Determines if it's a standard or admin session |
| Account\* GetAccount(std::string name, int account) | Returns nullptr if the name/account pair is not found in system. |
| std::string GetAccountOwner(int account) | Find customer name corresponding to account number. Returns empty string if account was not found |
| void PushTransactionRecord(int code, std::string name, int account\_number, double money, std::string misc); | Pushes transaction record with that info onto stack so that transactions file can later be created on logout. Name, account\_number, money, and misc have default values. |
| std::string FitStringToSpace(std::string string, size\_t size, char fluff, bool align\_right) | Formats string to match certain number of characters, using fluff to fill space. align\_right has a default value of true. |
| std::string PromptForAccountHolder() | Retrieves the account holder's name. |
| bool CheckLogin(bool admins\_only) | Prints an error if the user is not logged in. Prints an error if the user is not admin, if admins\_only is set to true. admins\_only has a default value of “false”..\ |
| double GetTransactionCharge(std::string name, int account\_number) | Retrieves Transaction Charge for a particular account. Behaviour is undefined if the account doesn't exist. |
| Void PrintToTransactionFile(std::deque<std::string>\* transactions, std::string fpath); | Consumes deque; appends to transactions file. fpath defaults to “transactions.txt” |
| AccountStatus | Defines the assessment of an account's validity e.g. is it available for use.  Also provides error messages corresponding to those states. |
| AccountStatus enum | Representation of various ways an account can be invalid; or valid. |
| AccountParser | Contains utility to parse in bank customer information from file. |
| TransactionIO | Functions enabling the use of the transactions file. |