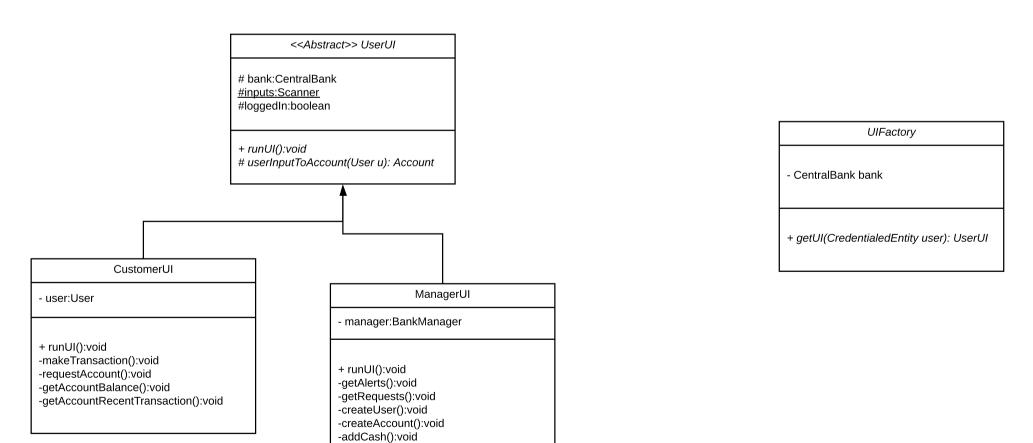
<<Abstract>> Account dateCreated: Calendar # balance:double # recentTransaction: Transaction + getDateCreated():Calendar + getRecentTransaction():Transaction + getBalance():double + transferIn(transaction Transaction):void + parseTransaction(Transaction transaction): boolean + transferOut(double amount): boolean + resetRecentTransaction(): void # checkUndo(accTransaction AccounttTransaction, recentAmount double): boolean updateUndo(accTransaction AccountTransaction, recentAmount double): void writeToOutgoing(transaction Transaction): void <<Abstract>> DebtAccount <<Abstract>> AssetAccount + parseTransaction(transaction Transaction): + resetRecentTransaction(): void CreditCardAccount LineOfCreditAccount SavingsAccount CheckingAccount interestRate:double transferOut(amount double):boolean transferOut(amount amount): boolean + updateInterest(): void # checkUndo(accTransaction # checkUndo(accTransaction AccountTransaction, recentAmount AccountTransaction , recentAmount + transferOut(amount double): boolean + transferOut(amount double): boolean double): boolean Account): boolean



-undoTransaction():void

BankDriver

 bank: CentralBank - currentUser: UserUI

- inputs: Scanner

- logIn():void
- -requestUserAccount(String

+ setManager(BankManager bm): void

- UIGetter: UIFactory

- + main(String[] args):void
- initManager():void
- <u>name): void</u>

+ getTime():Calendar

- + save():void + readDeposits():void
- - + addUser(User user):void
 - + getUsers():HashMap<String, User>

+ getManager(): BankManager

- users: HashMap<String, User>

- manager: BankManager

+ updateInterest():void

+ shutDown():void

- cashMachines: ArrayList<CashMachine>

+ date: Calendar

- addCashMachine(CashMachine cashMachine):void + getCashMachine():ArrayList<CashMachine>

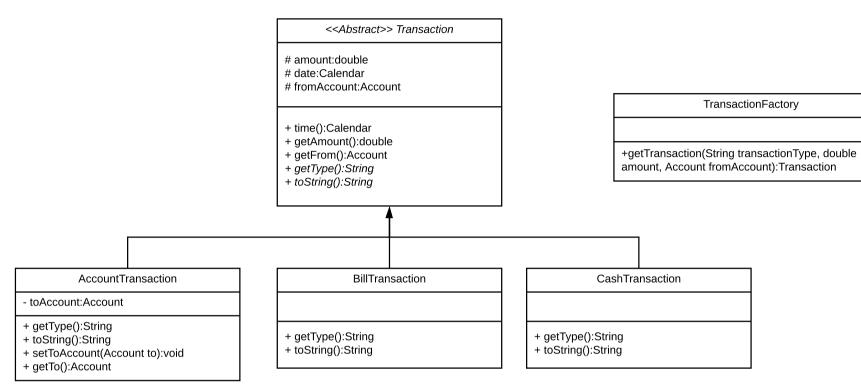
CentralBank

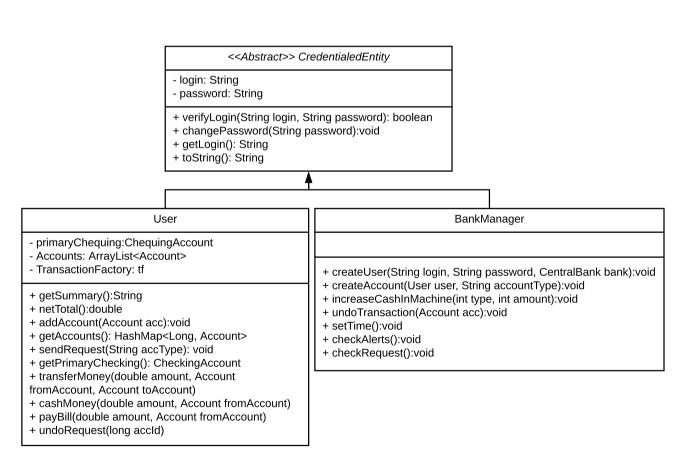
CashMachine

- amountOfCash:int

- typeOfCash:HashMap<Integer, Integer>

- + withdraw(Integer amount):void
- checkAmount():void
- sendAlert():void +addCash(Integer type, Integer amount):void





,	AccountNotFoundException

InputOutput

+ readTextFile(String path, CentralBank centralBank):void

- findAccount(User user, Long accountTime):Account + readUsersFromFile(String filePath): HashMap<String, User>

<u>+ readCashMachinesFromFile(String filePath): ArrayList<CashMachine></u>

+ saveToFile(String filePath, Object obj)