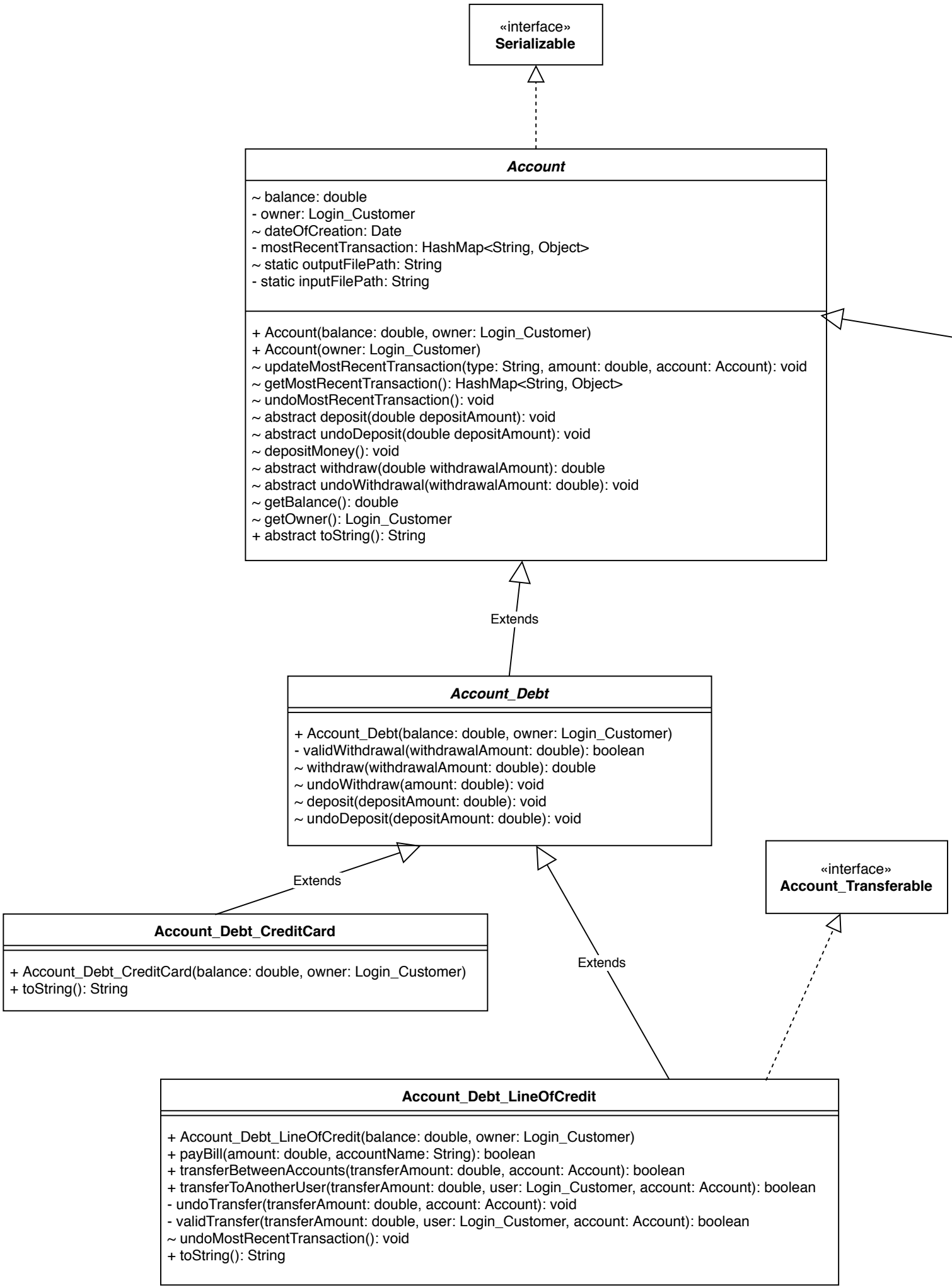
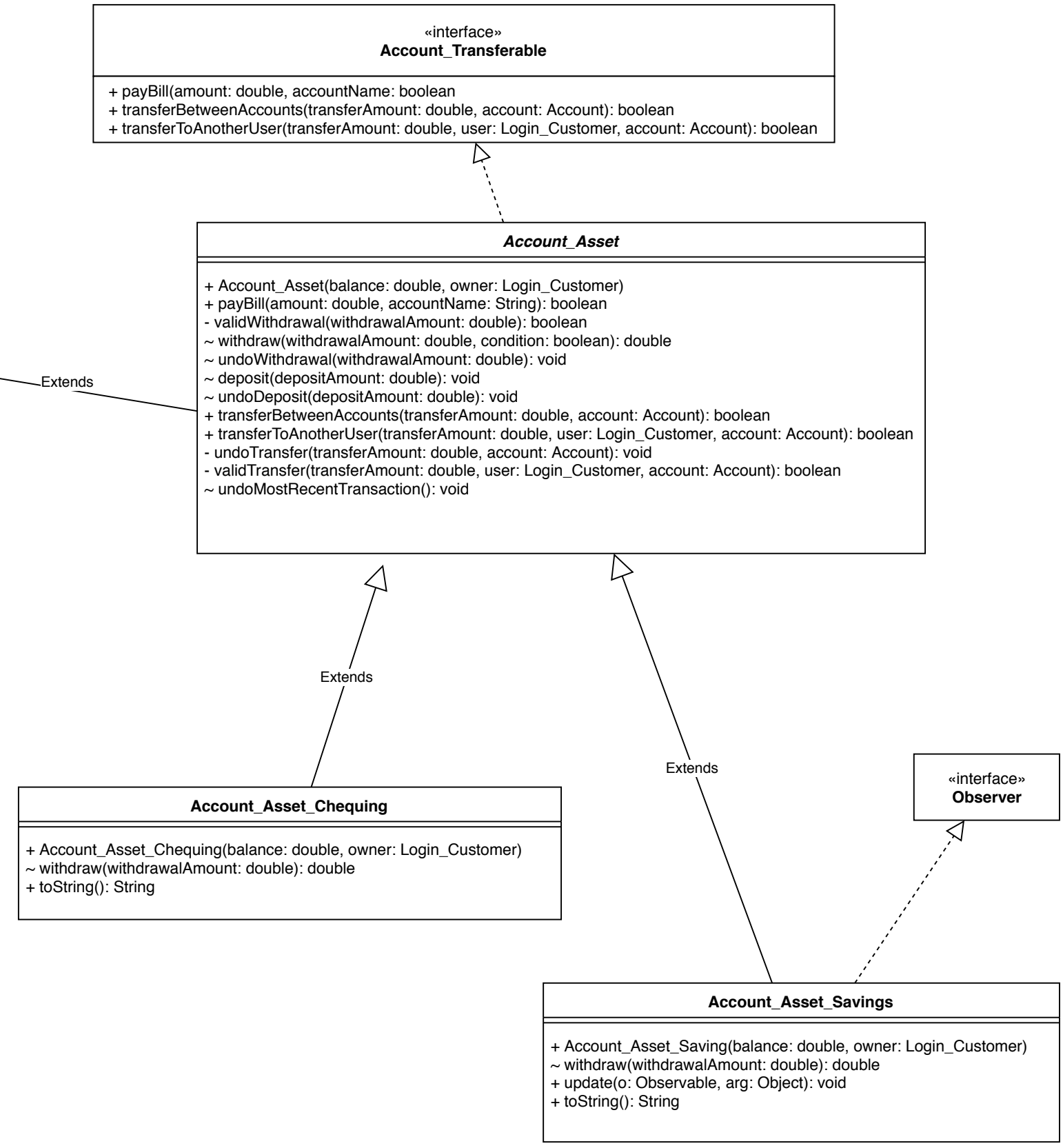
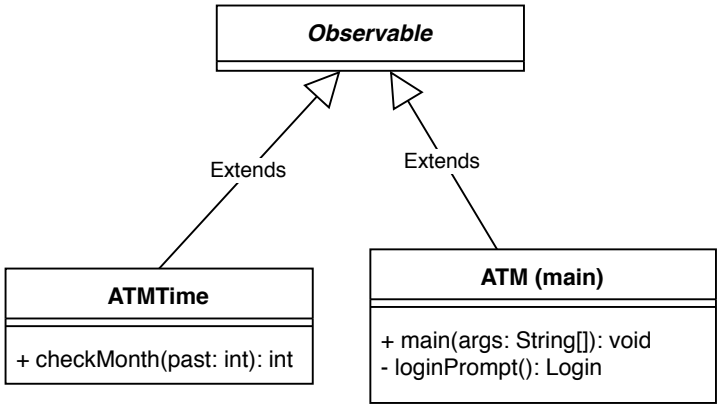

«interface»
Serializable
Account
~ balance: double
- owner: Login_Customer
~ dateOfCreation: Date
- mostRecentTransaction: HashMap<String, Object>
~ static outputFilePath: String
- static inputFilePath: String
+ Account(balance: double, owner: Login_Customer)
+ Account(owner: Login_Customer)
~ updateMostRecentTransaction(type: String, amount: double, account: Account): void
~ getMostRecentTransaction(): HashMap<String, Object>
~ undoMostRecentTransaction(): void
~ abstract deposit(double depositAmount): void
~ abstract undoDeposit(double depositAmount): void
~ depositMoney(): void
~ abstract withdraw(double withdrawalAmount): double
~ abstract undoWithdrawal(withdrawalAmount: double): void
~ getBalance(): double
~ getOwner(): Login_Customer
+ abstract toString(): String
Extends
Account_Debt
+ Account_Debt(balance: double, owner: Login_Customer)
- validWithdrawal(withdrawalAmount: double): boolean
~ withdraw(withdrawalAmount: double): double
~ undoWithdraw(amount: double): void
~ deposit(depositAmount: double): void
~ undoDeposit(depositAmount: double): void
Extends
«interface»
Account_Transferable
Account_Debt_CreditCard
+ Account_Debt_CreditCard(balance: double, owner: Login_Customer)
+ toString(): String
Extends
Account_Debt_LineOfCredit
+ Account_Debt_LineOfCredit(balance: double, owner: Login_Customer)
+ payBill(amount: double, accountName: String): boolean
+ transferBetweenAccounts(transferAmount: double, account: Account): boolean
+ transferToAnotherUser(transferAmount: double, user: Login_Customer, account: Account): boolean
- undoTransfer(transferAmount: double, account: Account): void
- validTransfer(transferAmount: double, user: Login_Customer, account: Account): boolean
~ undoMostRecentTransaction(): void
+ toString(): String


«interface»
Account_Transferable
+ payBill(amount: double, accountName: boolean
+ transferBetweenAccounts(transferAmount: double, account: Account): boolean
+ transferToAnotherUser(transferAmount: double, user: Login_Customer, account: Account): boolean
Account_Asset
+ Account_Asset(balance: double, owner: Login_Customer)
+ payBill(amount: double, accountName: String): boolean
- validWithdrawal(withdrawalAmount: double): boolean
~ withdraw(withdrawalAmount: double, condition: boolean): double
~ undoWithdrawal(withdrawalAmount: double): void
~ deposit(depositAmount: double): void
~ undoDeposit(depositAmount: double): void
+ transferBetweenAccounts(transferAmount: double, account: Account): boolean
+ transferToAnotherUser(transferAmount: double, user: Login_Customer, account: Account): boolean
- undoTransfer(transferAmount: double, account: Account): void
- validTransfer(transferAmount: double, user: Login_Customer, account: Account): boolean
~ undoMostRecentTransaction(): void
Extends
Extends
Extends
Account_Asset_Chequing
+ Account_Asset_Chequing(balance: double, owner: Login_Customer)
~ withdraw(withdrawalAmount: double): double
+ toString(): String
«interface»
Observer
Account_Asset_Savings
+ Account_Asset_Saving(balance: double, owner: Login_Customer)
~ withdraw(withdrawalAmount: double): double
+ update(o: Observable, arg: Object): void
+ toString(): String



<<Utility>>
Cash

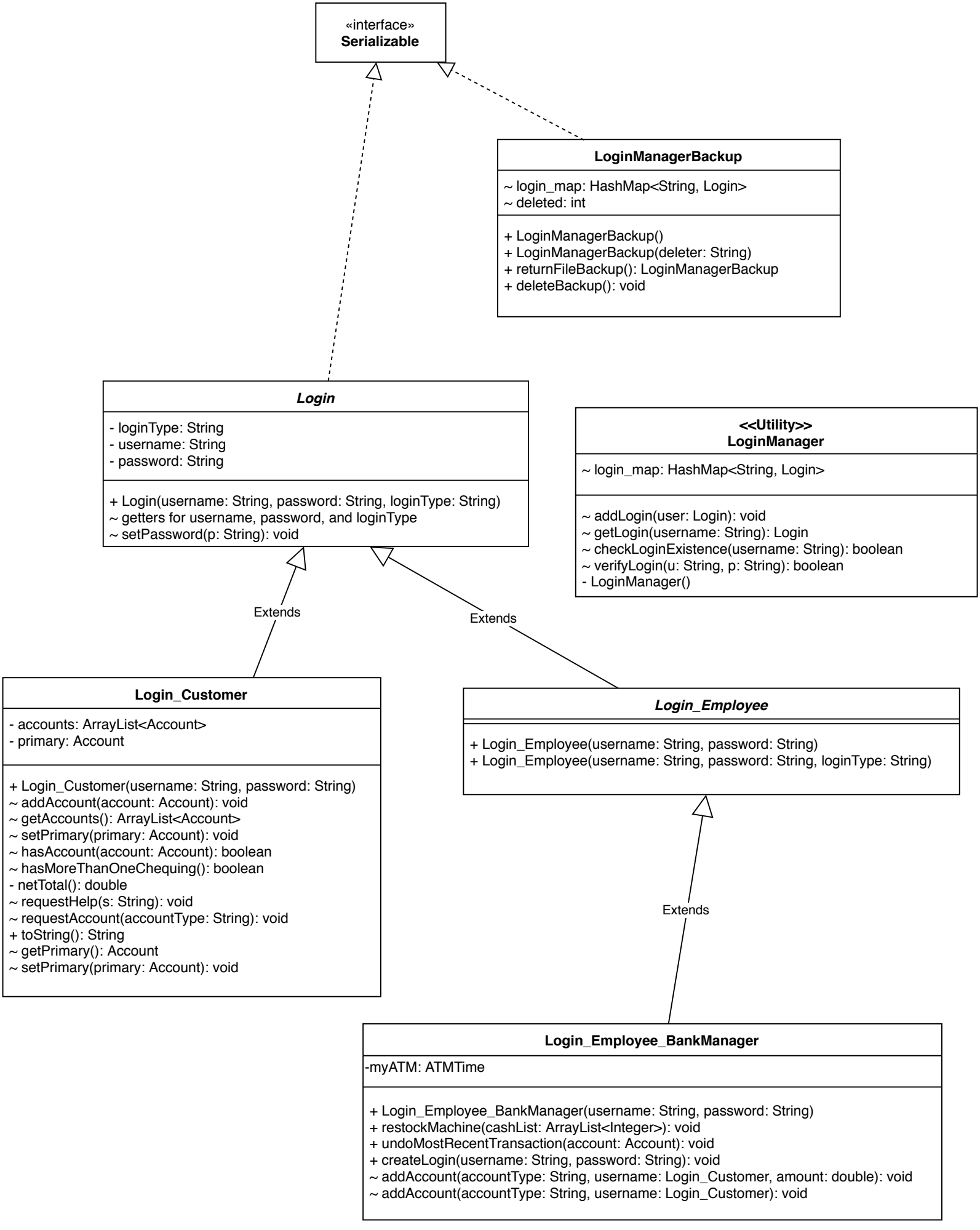
- bills: HashMap<String, Integer>
- outputFilePath: String

- isAmountBelowTwenty(): boolean
- checkDenom(): void
- sendAlert(): void
+ cashDeposit(cashList: ArrayList<Integer>): void
- verifyCashWithdrawal(amount: double): ArrayList<Integer>
+ isThereEnoughBills(amount: double): boolean
+ cashWithdrawal(amount: double): void
~ undoCashWithdrawal(amount: double): void

Options

- loginUser: Login
- options: LinkedHashMap<String, Thread>
- helped: boolean

+ Options(loginUser: Login)
- createOptions(): void
- displayOptions(): void
- selectOptions(): void
- createLoginPrompt(): void
- selectAccountTypePrompt(): void
- createAccountPrompt(): void
- restockPrompt(): void
- logoutPrompt(): void
- clearDataPrompt(): void
- selectAccountPrompt(customer: Login_Customer): Account
- selectAccountPrompt(customer: Login_Customer, exclusion: String): Account
- undoPrompt(): void
- setPasswordPrompt(): void
- setPrimaryPrompt(): void
- requestAccountPrompt(): void
- withdrawalPrompt(): void
- depositPrompt(): void
- payBillPrompt(): void
- transferBetweenAccountsPrompt(): void
- transferToAnotherUserPrompt(): void


«interface»
Serializable
LoginManagerBackup
~ login_map: HashMap<String, Login>
~ deleted: int
+ LoginManagerBackup()
+ LoginManagerBackup(deleter: String)
+ returnFileBackup(): LoginManagerBackup
+ deleteBackup(): void
Login
- loginType: String
- username: String
- password: String
+ Login(username: String, password: String, loginType: String)
~ getters for username, password, and loginType
~ setPassword(p: String): void
<<Utility>>
LoginManager
~ login_map: HashMap<String, Login>
~ addLogin(user: Login): void
~ getLogin(username: String): Login
~ checkLoginExistence(username: String): boolean
~ verifyLogin(u: String, p: String): boolean
- LoginManager()
Extends
Extends
Login_Customer
- accounts: ArrayList<Account>
- primary: Account
+ Login_Customer(username: String, password: String)
~ addAccount(account: Account): void
~ getAccounts(): ArrayList<Account>
~ setPrimary(primary: Account): void
~ hasAccount(account: Account): boolean
~ hasMoreThanOneChequing(): boolean
- netTotal(): double
~ requestHelp(s: String): void
~ requestAccount(accountType: String): void
+ toString(): String
~ getPrimary(): Account
~ setPrimary(primary: Account): void
Login_Employee
+ Login_Employee(username: String, password: String)
+ Login_Employee(username: String, password: String, loginType: String)
Extends
Login_Employee_BankManager
-myATM: ATMTime
+ Login_Employee_BankManager(username: String, password: String)
+ restockMachine(cashList: ArrayList<Integer>): void
+ undoMostRecentTransaction(account: Account): void
+ createLogin(username: String, password: String): void
~ addAccount(accountType: String, username: Login_Customer, amount: double): void
~ addAccount(accountType: String, username: Login_Customer): void