

---

# **CMPSC122-hw4-bank Documentation**

**Eric Roeum**

**Mar 13, 2018**



## **CONTENTS:**



## CMPSC122-HW4-BANK

### 1.1 sample package

#### 1.1.1 Submodules

#### 1.1.2 sample.assistant module

**class** `sample.assistant.Assistant` (*userID*, *customers*, *bankTellers*)

Bases: `bankTeller.BankTeller`

Represents a user with mid-level priveledges Users having these priveledges are usually bank tellers

**createBankTeller** (*userID*, *customers*)

Creates a Bank Teller under this account

##### Parameters

- **userID** (*string*) – the userID of the assistant
- **customers** (*list*) – all customers that assitant can access
- **bankTellers** (*list*) – all bank tellers that assistant can access

**viewBankTellers** ()

Returns the bank tellers under this account

#### 1.1.3 sample.bankTeller module

**class** `sample.bankTeller.BankTeller` (*userID*, *customers*)

Bases: `customer.Customer`

Represents a user with mid-level priveledges Users having these priveledges are usually bank tellers

**acceptRequest** (*customer*, *reqNum*)

Accepts a request for a customer

##### Parameters

- **customer** (*Customer*) – Customer with the request
- **reqNum** (*int*) – Request number that wishes to be resolved

**viewCustomers** ()

Returns the customers under this account

**viewRequests** (*customer*)

Retrieves the requests held within the customer

**Parameters** **customer** (*Customer*) – Customer with requests

### 1.1.4 sample.core module

`sample.core.main (arg)`

Runs the core of the program Integrates all other python scripts/modules Use the core.py to run the Bank python script

**Parameters** **arg** (*list*) – list of arguments in terminal

### 1.1.5 sample.customer module

**class** `sample.customer.Customer (value, userID_owner, userID_users=[])`

Bases: `object`

Represents a user with low-level priveledges Users only having these priveledges are usually customers

**addHistory** (*event*)

Adds an event to the history

**Parameters** **event** (*string*) – Event that has take place (only accepted requests)

**getHistory** ()

Retrieves history of this account

**getOwner** ()

Retrieves owner of the bank

**getRequests** ()

Retrieves request of account

**getSubaccounts** ()

Retrives accounts owned by this account

**getUsers** ()

Retrieves priveledged users of the bank

**getValue** ()

Retrieves value held within bank

**removeRequest** (*reqNum*)

Removes request from request list

**Parameters** **reqNum** (*int*) – Index of request

**requestDelta** (*delta*)

Requests the withdrawl or deposit into customer's account

**Parameters** **delta** (*float*) – Amount to request funds

### 1.1.6 sample.manager module

**class** `sample.manager.Manager (userID, customers, bankTellers, assistants)`

Bases: `assistant.Assistant`

Represents a user with high-level priveledges Users having these priveledges are usually managers

**createAssistant** (*userID, customers, bankTellers*)

Creates an assistant object

**Parameters**

- **userID** (*string*) – the userID of the assistant
- **customers** (*list*) – all customers that assistant can access
- **bankTellers** (*list*) – all bank tellers that assistant can access

**viewAssistants()**

Returns the assistants under this class

**1.1.7 sample.passwordAuthentication module**

**class** sample.passwordAuthentication.Password(*approvedUsers*)

Bases: object

**addAuthenticatedUser** (*userid, password*)

Adds authenticated user using UUID and SHA512 password hashing

**Parameters**

- **userid** (*string*) – User ID that user has chosen
- **password** (*string*) – Password to attempt to authenticate userid

**authenticate\_password** (*userid, password*)

Authenticates Passwords using UUID and SHA512 password hashing

**Parameters**

- **userid** (*string*) – User ID that user has chosen
- **password** (*string*) – Password to attempt to authenticate userid

**hashUsername** (*userID*)

Hashes provided username for hashed user id

**Parameters** **userid** – User ID that user has chosen

**readEncrypted** (*filename, dest='./*)

Imports all usernames and passwords from a txt file File must be formatted using \_writeEncrypted function  
NOTE: All previous passwords will be deleted and replaced

**Parameters**

- **filename** (*string*) – filename of txt file
- **dest** (*string*) – location to write txt file

**writeEncrypted** (*filename, dest='./*)

Writes all usernames and passwords to a txt file

**Parameters**

- **filename** (*string*) – filename of txt file
- **dest** (*string*) – location to write txt file

**1.1.8 sample.terminalFunctions module**

sample.terminalFunctions.**acceptRequest** (*user, customer, reqNum*)

Accepts a customer's request

**Parameters**

- **user** (*class*) – The current user
- **customer** (*customer*) – Customer that is being accepted
- **reqNum** (*int*) – Request number corresponding to the request

`sample.terminalFunctions.balance` (*person*)

Shows balance in the account

`sample.terminalFunctions.clear` (*platf*)

`sample.terminalFunctions.deposit` (*person*, *value\_to\_add*)

Deposits value into the account

**Parameters** *value\_to\_add* (*float*) – Value to be added in funds

`sample.terminalFunctions.exit` ()

Exits the program This is a “fake” function that only confirms exit Clears screen after exit

`sample.terminalFunctions.help` ()

Help menu for giving users all commands with short description

`sample.terminalFunctions.ls` (*accountType*, *account*, *customers*)

Lists all accounts accessible

**Parameters**

- **accountType** (*string*) – type of account
- **account** (*Customer*) – The account that is being inspected

`sample.terminalFunctions.make` (*genesis*, *accountType*, *userID*, *users*)

Create an account

**Parameters**

- **genesis** (*Customer*) – Original account
- **accountType** (*string*) – type of account
- **userID** (*string*) – user id of the account

`sample.terminalFunctions.requests` (*user*, *customer*)

Lists all requests of the customer

**Parameters**

- **user** (*class*) – The current user
- **customer** (*customer*) – Customer that will be viewed

`sample.terminalFunctions.whoami` (*person*)

Return the owner’s ID

`sample.terminalFunctions.withdrawal` (*person*, *value\_to\_deduct*)

Withdrawals value from the account

**Parameters** *value\_to\_deduct* (*float*) – Value to be withdrawn in funds

### 1.1.9 sample.terminalInterface module

`sample.terminalInterface.displayInterface` ()

Terminal-Based banking application

**Parameters** *customers* (*dict*) – dictionary of all created customers



**Params users** Class with all developed user-password authentication

### 1.1.10 Module contents

## 1.2 tests package

### 1.2.1 Module contents



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### S

sample, ??  
sample.assistant, ??  
sample.bankTeller, ??  
sample.core, ??  
sample.customer, ??  
sample.manager, ??  
sample.passwordAuthentication, ??  
sample.terminalFunctions, ??  
sample.terminalInterface, ??

### t

tests, ??