assignment_3

The page contains a UML sequence diagram and a UML class diagram.

Sequence diagram:

- `enterAccount()` → : Controller
- : Controller → x :ManagerAccount : `<<create>>`
- : Controller → a :AimAccount : `<<create>>`
- : Controller → :AccountManager : `val := CheckAccount(x)`
- :AccountManager → :AccountStorage : `sx := getPermissionByAccount(x)`
- :AccountManager → :AccountManager : `eq:=checkPermission(Administrator,sx)`

Alt

val == true
- : Controller → :AccountManager : `delActList := deleteAccount(a)`
  - **Loop** [for all stored account]
    - :AccountManager → :AccountStorage : `sa := getNext()`
    - :AccountManager → :AccountManager : `eq:=compare(a,sa)`
    - :AccountManager → :AccountStorage : `if(eq) delete(sa)`
- : Controller → : Logger : `logTransaction(x,a,delActList,val)`
- `<<destroy>>` a :AimAccount
- `<<destroy>>` x :ManagerAccount

  Alt
  - [delActList.size() >0]
    - `signal:"Success"`
  - [delActList.size() == 0]
    - `prompt:"try again"`

else
- : Controller → : Controller : `Reject the Delete operation`
- : Controller → : Logger : `logTransaction(x,a,val)`

Class diagram:

AccountManager
- + deleteAccount(a:Account):list
- - compare(a,sa):boolean
- + CheckAccount(x):boolean
- - checkPermission(administrator, sx): boolean

AccountStorage
- + getNext():Account
- + delete(sa:Account)
- + getPermissionByAccount(x:ManagerAccount): permissionType

validAccounts — relationship between AccountManager and AccountStorage

Logger
- + logTransaction(a:Account,x:ManagerAccount,delActList: list)
- + logTransaction(x:ManagerAccount,a:Account,val: boolean)

Controller
- + enterAccount(a:Account)

checker — 1 (Controller to AccountManager)
logger — 1 (Controller to Logger)

Account
- - id_: string
- - code: string
- - timestamp: long
- - permissionType: int64

1...* / 1 (Account to AccountStorage)