Sequence diagram with lifelines: **UI**, **Controller**, **Manager**, **Service**, **Data access**, **Data**

Create

Single Creation

- Controller → Controller: singleOrBulk()
- Controller → Controller: return instance
- UI → Controller: createUser(creationMethod)

Single Creation

- Manager → Service: getSystemIDAndPassword()
- Service → Manager: return instance
- Manager → Service: checkSystemID(ID)
- Service → Data access: new DS.get()
- Data access → Service: return instance
- Service → Data access: DS.get(ID)
- Data access → Data: get query that contains ID
- Data → Data access: return query
- Data access → Service: return query
- Service → Manager: return bool
- Manager → UI: if ID exist → notify that system ID already exists

NOTE: In creation, you're not verifying anything, so there's no need to verify if the inputted password matches requirements

- Manager → Service: new storeUser()
- Service → Manager: return instance
- Manager → Service: storeUser(system ID, password)
- Service → Data access: new DS.store()
- Data access → Service: return instance
- Service → Data access: DS.store(email, name, password)
- Data access → Data: store userInfo
- Data access → Service: return success/fail
- Service → Manager: return success/fail
- Manager → Controller: return success/fail

Bulk Creation

Bulk Creation

- Manager → Service: getAmountOfAccounts()
- Service → Manager: return instance
- Manager → Service: checkDatabaseCapacity()
- Service → Data access: new DS.getCapacity()
- Data access → Service: return instance
- Service → Manager: return bool
- Manager → UI: if (capacity is full) OR (max capacity < current capacity + amount) → Error message

Loop

- Controller → Manager: discard ID
- Manager → Service: generateSystemIDAndPassword(amount)
- Service → Data access: DS.get(ID)
- Data access → Data: get query that contains ID
- Data → Data access: return query
- Data access → Service: return bool
- Service → Manager: return bool
- Manager → UI: if ID exist → notify that system ID already exists
- Manager → Service: storeUser(system ID, password)
- Service → Data access: new DS.store()
- Data access → Service: return instance

NOTE: Should get a CSV of all of the emails so that we can create the accounts, but that's not my job

- Service → Data access: DS.store(email, name, password)
- Data access → Data: store userInfo
- Data access → Service: return success/fail
- Service → Manager: return success/fail

Update

- Controller → Controller: UpdateUser()
- Controller → Controller: return instance
- UI → Controller: UpdateUser(UserID)
- Controller: Authorization Check
- Controller → UI: if failed → Error message
- Manager → Service: UpdateInfo()
- Service → Manager: return instance
- Manager → Service: UpdateInfo(Name, Phone...)
- Service → Data access: new DS.update()
- Data access → Service: return instance
- Service → Data access: DS.update(Name, Phone...)
- Data access → Data: update User info
- Data access → Service: return success/fail
- Manager → Service: SetDisable()
- Service → Manager: return instance
- Manager → Service: SetDisable(True, False)
- Service → Manager: return success/fail
- Controller → Manager: End update

Bulk enable / disable

- Controller: Authorization Check
- Controller → UI: if not admin → Error message
- Controller → Manager: List of UserID

Loop

- Manager → Service: SetDisable()
- Service → Manager: return instance
- Manager → Service: SetDisable(True, False)
- Service → Manager: return success/fail

Delete

- Controller → Controller: DeleteUser()
- Controller → Controller: return instance
- UI → Controller: DeleteUser(UserID)
- Controller: Authorization Check
- Controller → UI: if failed → Error message
- Manager → Service: DeleteUser()
- Service → Manager: return instance
- Manager → Service: DeleteUser(UserID)
- Service → Data access: new DS.delete()
- Data access → Service: return instance
- Service → Data access: DS.delete(UserID)
- Data access → Data: Delete User
- Data access → Service: return success/fail
- Service → Manager: return success/fail
- Controller → Manager: end delete

Bulk delete

- Controller: Authorization Check
- Controller → UI: if not admin → Error message
- Controller → Manager: List of UserID

Loop

- Manager → Service: DeleteUser()
- Service → Manager: return instance
- Manager → Service: DeleteUser(UserID)
- Service → Data access: new DS.delete()
- Data access → Service: return instance
- Service → Data access: DS.delete(UserID)
- Data access → Data: Delete User
- Data access → Service: return success/fail