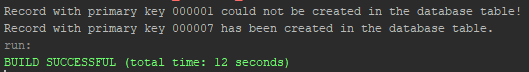
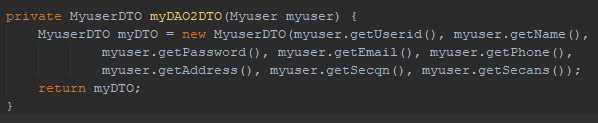
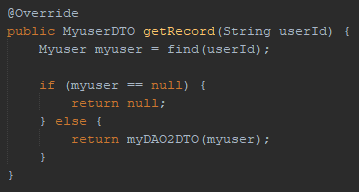
# 4.1P

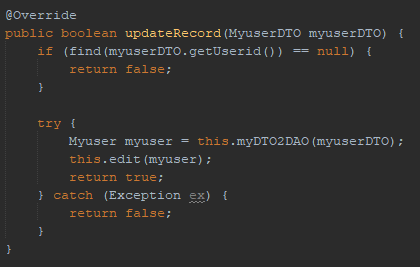
## Task 1.

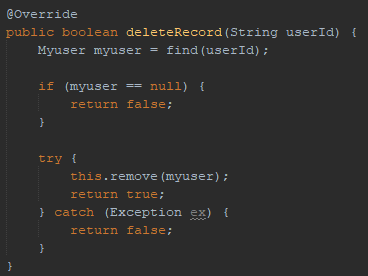
The lab was completed and produced the following output:  


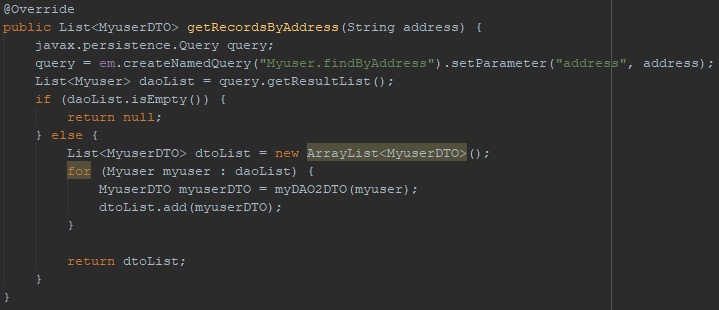
## Task 2.

1. The function ‘myDAO2DTO()’ was created:  


2. The function ‘getRecord()’ was created:  


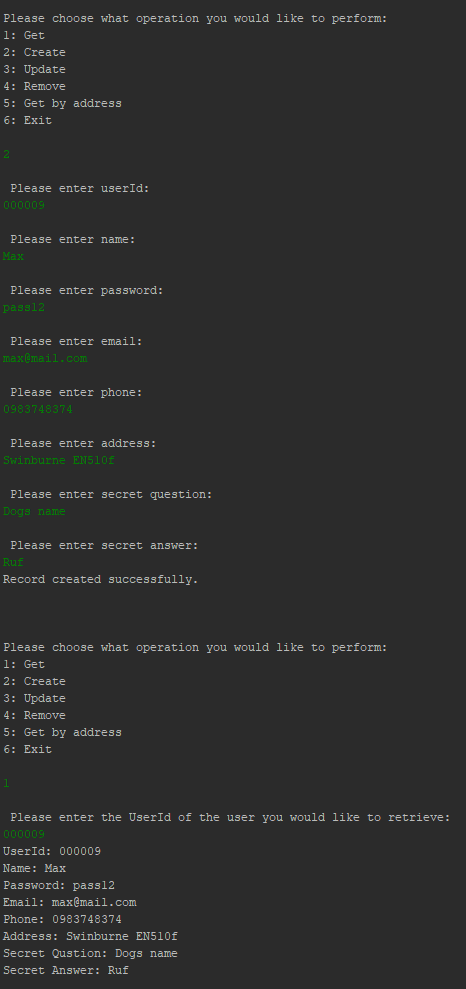
3. The function ‘updateRecord()’ was created:  


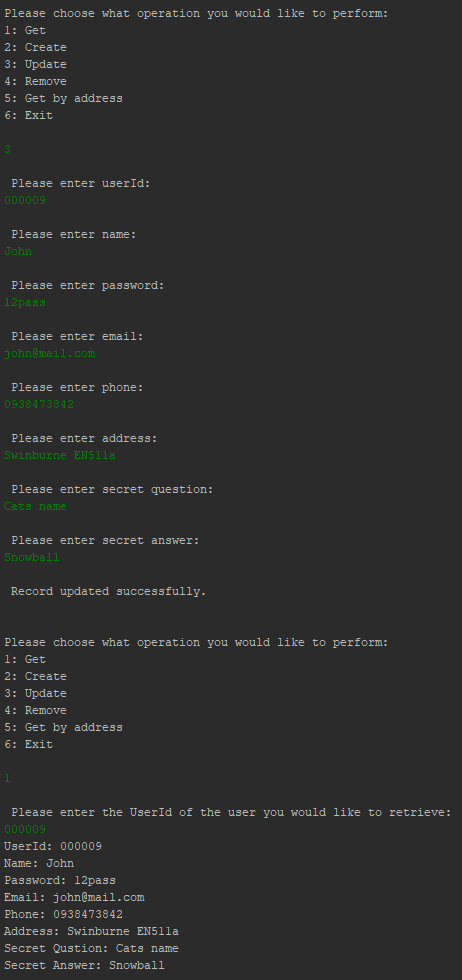
4. The function ‘deleteRecord()’ was created:  


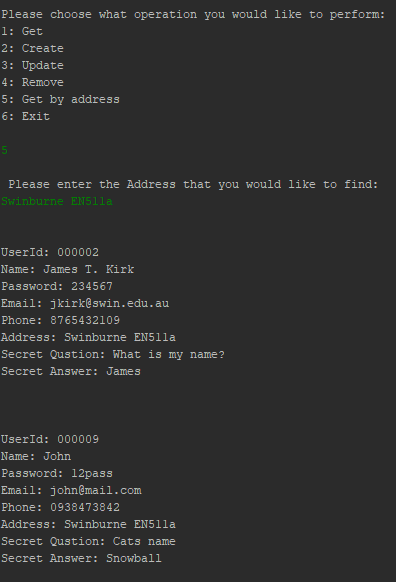
5. The function ‘getRecordsByAddress()’ was created:  


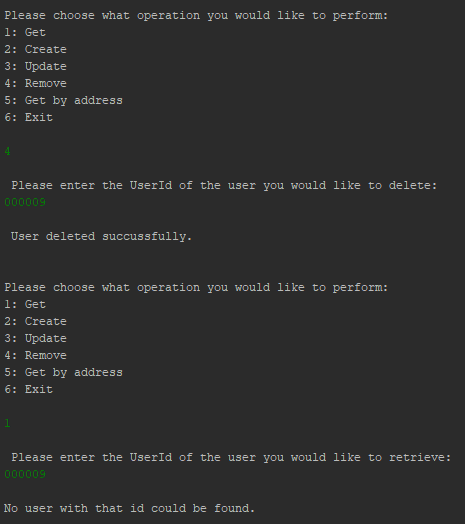
## Task 3.

The same code for the simple menu was taken from previous tasks and modified slightly to work with the new features of this task.

Creating and getting a record:  


Updating a record:  


Getting users by address:  


Deleting a record:  


## Task 4.

4.1  
Myuser is the class that is responsible for the ORM work, as it is the object which holds the data and is used to directly manipulate the database with the values it contains.

4.2  
Stateless session bean pooling is the process of having pre-prepared instances of an object that can each be used for a single transaction. As each transaction uses a different bean, there is no memory of what occurred in each transaction and separate beans can not act on values that were passed with other beans. In this task, ‘MyuserFacade’ was the class that defined the bean and what it could do. Bean instance pooling can help scalability through performance optimization, and also through the split multi-layer approach it requires when dealing with a client, database and business layer.

Code can be found for further review at:   
<https://github.com/CyrusEdgren/Secure_Scalable_Software/tree/master/4.1P>