

### 3.1P

#### Task 1.

Lab 3 was completed and gave the following output:

```
run:
[EL Info]: 2021-03-24 16:01:35.715--ServerSession(1661406123)--EclipseLink, version: Eclipse
[EL Info]: connection: 2021-03-24 16:01:35.863--ServerSession(1661406123)--file:/D:/Repos/Se
Record with primary key 000001 could not be created in the database table!
Record with primary key 000006 has been created in the database table.
BUILD SUCCESSFUL (total time: 0 seconds)
```

(The first record could not be created as the old table data from the last task already contained an item with the primary key 000001.)

#### Task 2.

The function getRecord() was created:

```
public MyuserDTO getRecord(String userId) {
    try {
        Myuser myuser = findMyuser(userId);

        if (myuser == null) {
            return null;
        } else {
            return myDAO2DTO(myuser);
        }
    } catch (Exception ex) {
        throw ex;
    }
}
```

The function updateRecord() was created:

```
public boolean updateRecord(MyuserDTO myuserDTO) {
    Myuser myuser = this.myDTO2DAO(myuserDTO);
    try {
        if (findMyuser(myuser.getUserid()) == null) {
            // myuser does not exist
            return false;
        }
        // myuser does exist in database
        em.getTransaction().begin();
        em.merge(myuser); // to update an object in the database
        em.getTransaction().commit();
        return true;
    } catch (Exception ex) {
        throw ex;
    }
}
```

The function deleteRecord() was created:

```
public boolean deleteRecord(String userId) {  
    try {  
        Myuser myuser = findMyuser(userId);  
  
        if (myuser == null) {  
            return false;  
        } else {  
            try {  
                em.getTransaction().begin();  
                em.remove(myuser); // to delete an object in the database  
                em.getTransaction().commit();  
                return true;  
            } catch (Exception ex) {  
                throw ex;  
            }  
        }  
    } catch (Exception ex) {  
        throw ex;  
    }  
}
```

### Task 3.

The console menu code from the last pass task was copied over to this task and modified to work with the new CRUD operations.

Get record:

```
Please choose what operation you would like to perform:  
1: Get  
2: Create  
3: Update  
4: Remove  
5: Exit  
  
1  
  
Please enter the UserId of the user you would like to retrieve:  
000005  
UserId: 000005  
Name: Harry Potter  
Password: 567890  
Email: hpotter@swin.edu.au  
Phone: 6543210987  
Address: Swinburne EN514a  
Secret Qustion: What is my last name?  
Secret Answer: Potter
```

Create record:

```
Please choose what operation you would like to perform:
1: Get
2: Create
3: Update
4: Remove
5: Exit

3

Please enter userId:
000009

Please enter name:
Dave

Please enter password:
pas321

Please enter email:
dave@mail.com

Please enter phone:
0938472842

Please enter address:
12 rock st

Please enter secret question:
cat's name

Please enter secret answer:
fluffy
Record created successfully.

Please choose what operation you would like to perform:
1: Get
2: Create
3: Update
4: Remove
5: Exit

1

Please enter the UserId of the user you would like to retrieve:
000009
UserId: 000009
Name: Dave
Password: pas321
Email: dave@mail.com
Phone: 0938472842
Address: 12 rock st
Secret Qustion: cat's name
Secret Answer: fluffy
```

Update record:

```
Please choose what operation you would like to perform:
1: Get
2: Create
3: Update
4: Remove
5: Exit

3

Please enter userId:
000009

Please enter name:
Dave2

Please enter password:
321pas

Please enter email:
Dave2@mailc.om

Please enter phone:
9384028742

Please enter address:
31 pebble rd

Please enter secret question:
Dog's name

Please enter secret answer:
Max

Record updated successfully.

Please choose what operation you would like to perform:
1: Get
2: Create
3: Update
4: Remove
5: Exit

1

Please enter the UserId of the user you would like to retrieve:
000009
UserId: 000009
Name: Dave2
Password: 321pas
Email: Dave2@mailc.om
Phone: 9384028742
Address: 31 pebble rd
Secret Qustion: Dog's name
Secret Answer: Max
```

Delete record:

```
Please choose what operation you would like to perform:
1: Get
2: Create
3: Update
4: Remove
5: Exit

4

Please enter the UserId of the user you would like to delete:
000009

User deleted succussfully.

Please choose what operation you would like to perform:
1: Get
2: Create
3: Update
4: Remove
5: Exit

1

Please enter the UserId of the user you would like to retrieve:
000009

No user with that id could be found.
```

## Task 4.

### 4.1

MyuserDB is responsible for doing all of the ORM work, as it handles the transformation of the data transformation object (MyuserDTO) into the entity class that is used by the database (Myuser).

### 4.2

The Myuser class has a different role than it had in the application from lab 2. It is no longer a DTO, but rather the entity class that is used with the EntityManager API to control records in the database.

### 4.3

The Myuser entity class in this application is in the JPA domain instead of the JDBC domain of the last lab. This is because it is now handling persistent data in a class rather than just using queries to manipulate the database.

Code can be found for further review at:

<https://github.com/CyrusEdgren/Secure-Scalable/tree/master/3.1P>