Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

Step 1 ending part:

```
dist:
pre-run-deploy:
Distributing C:\Users\User\Desktop\year2_semester_1_stuff\cos30041\all_labs\week4_lab\ED-JEE-DTO\ED-JEE-DTO-appclient\dist\ED-JEE-DTO-appclient.jar
post-run-deploy:
run-deploy:
Copying 1 file to C:\Users\User\Desktop\year2_semester_1_stuff\cos30041\all_labs\week4_lab\ED-JEE-DTO\ED-JEE-DTO-appclient\dist
Copying 2 files to C:\Users\User\Desktop\year2_semester_1_stuff\cos30041\all_labs\week4_lab\ED-JEE-DTO\ED-JEE-DTO-appclient\dist\ED-JEE-DTO-appclientClient
Warning: C:\Users\User\Desktop\year2_semester_1_stuff\cos30041\all_labs\week4_lab\ED-JEE-DTO\ED-JEE-DTO-appclient\dist\gfdeploy\ED-JEE-DTO-appclient does not exist.
Record with primary key 000001 could not be created in the database table!
Record with primary key 000007 could not be created in the database table!
Type the number to perform the desired action:
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Get records at an Address
5: Exit
```

Step 2 all parts:

```java
    private Myuser myDTO2DAO(MyuserDTO myuserDTO) {
        Myuser myuser = new Myuser();
        myuser.setUserid(myuserDTO.getUserid());
        myuser.setName(myuserDTO.getName());
        myuser.setPassword(myuserDTO.getPassword());
        myuser.setEmail(myuserDTO.getEmail());
        myuser.setPhone(myuserDTO.getPhone());
        myuser.setAddress(myuserDTO.getAddress());
        myuser.setSecqn(myuserDTO.getSecQn());
        myuser.setSecans(myuserDTO.getSecAns());
        return myuser;
    }

    private MyuserDTO myDAO2DTO(Myuser myuser) {
        MyuserDTO smth = new MyuserDTO(myuser.getUserid(), myuser.getName(), myuser.getPassword(), myuser.getEmail(), myuser.getPhone(), myuser.getAddress(), myuser.getSecqn(), myuser.getSecans());
        return smth;
    }

    @Override
    public MyuserDTO getRecord(String userId) {
        Myuser myuser = find(userId);
        if (myuser != null) {
            return new MyuserDTO(myuser.getUserid(), myuser.getName(), myuser.getPassword(), myuser.getEmail(), myuser.getPhone(), myuser.getAddress(), myuser.getSecqn(), myuser.getSecans());
        } else {
            return null;
        }
    }

    @Override
    public boolean updateRecord(MyuserDTO myuserDTO) {
        Myuser myuser = find(myuserDTO.getUserid());
        if (myuser != null) {
            try {

                edit(myDTO2DAO(myuserDTO)); // to add an object to database

                return true;
            } catch (Exception ex) {
                throw ex;
            }

        } else {
```

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

```
        } else {
            return false;
        }
    }

    @Override
    public boolean deleteRecord(String userId) {
        Myuser myuser = find(userId);
        if (myuser != null) {
            try {

                remove(myuser);
                return true;
            } catch (Exception ex) {
                throw ex;
            }

        } else {
            return false;
        }
    }

    @Override
    public ArrayList<MyuserDTO> getRecordsByAddress(String address) {
        javax.persistence.Query query;
        query = em.createNamedQuery("Myuser.findByAddress").setParameter("address", address);
        List<Myuser> daoList = query.getResultList();
        ArrayList<MyuserDTO> dtoList = new ArrayList<>();
        for (Myuser user : daoList) {
            dtoList.add(myDAO2DTO(user));
        }
        return dtoList;
    }

}
```

{Note: here my mistake has been I had written findby here instead of findBy for the address part. Thus the program couldn't understand what I was aksing it to do as the rest of the named queries given in myuser were using findBy and not findby}

Step 3 all code parts:

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

Source History

```java
        Menu(client);

    }

    public static void Menu(MyuserAppClient client) {
        Scanner sc = new Scanner(System.in);
        int option = 0;
        System.out.println("Type the number to perform the desired action:");
        while (option != 5) {
            System.out.println("1: Get a record for a certain UserID");
            System.out.println("2: Update a Record");
            System.out.println("3: Delete a Record");
            System.out.println("4: Get records at an Address");
            System.out.println("5: Exit");
            System.out.print("\nPlease select an option (1-5): ");
            option = sc.nextInt();
            sc.nextLine(); //skip '\n

// fix this part for all cases 1 to 4 so that it can do the task
//take help from your copy and lab notes for that
//.............................
            switch (option) {

                case 1:
                    System.out.println("Enter the UserID:");
                    Scanner newUserID = new Scanner(System.in);
                    String newUserID1 = newUserID.nextLine();



                    MyuserDTO myuser = client.getRecord(newUserID1);

                    if (myuser != null) {

                        System.out.println(myuser.getUserid().toString());
                        System.out.println(myuser.getName().toString());
                        System.out.println(myuser.getPhone().toString());
                        System.out.println(myuser.getPassword().toString());
                        System.out.println(myuser.getAddress().toString());
                        System.out.println(myuser.getEmail().toString());
```

edjee.MyuserAppClient > Menu > while (option != 5) > switch (option) > case 4:

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

```
                    System.out.println(myuser.getSecAns().toString());

                }
                else{
                    System.out.println("UserID is not found");

                }

                break;

                case 2:
                System.out.println("Enter record details to be updated:");
                Scanner newData = new Scanner(System.in);

                System.out.println("UserId:");
                String newUserId2 = newData.nextLine();

                System.out.println("Name:");
                String newName = newData.nextLine();
                System.out.println("Password:");
                String newPassword = newData.nextLine();
                System.out.println("Email:");
                String newEmail = newData.nextLine();
                System.out.println("Phone:");
                String newPhone = newData.nextLine();
                System.out.println("Address:");
                String newAddress = newData.nextLine();
                System.out.println("SECQN:");
                String newSecQn = newData.nextLine();
                System.out.println("SECAns:");
                String newSecAns = newData.nextLine();




                MyuserDTO myuserNew = new MyuserDTO(newUserId2, newName,newPas
sword,newEmail,newPhone,newAddress,newSecQn,newSecAns);


                 boolean outcome2 = client.updateRecord(myuserNew);
```

edjee.MyuserAppClient > Menu > while (option != 5) > switch (option) > case 4:

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

```java
                    boolean outcome2 = client.updateRecord(myuserNew);

                    if (outcome2 == true) {
                        System.out.println("Updated Successfully");
                    } else {
                        System.out.println("Failed to update");
                    }

                    break;
                case 3:
                    System.out.println("Enter the UserID:");
                    Scanner deleteUserID = new Scanner(System.in);
                    String deleteUserID1 = deleteUserID.nextLine();


                    boolean outcome3 = client.deleteRecord(deleteUserID1);

                    if (outcome3 == true) {
                        System.out.println("Deleted Successfully");
                    } else {
                        System.out.println("Failed to delete");
                    }

                    break;

                    case 4:
                    System.out.println("Enter the Address:");
                    Scanner address = new Scanner(System.in);
                    String address1 = address.nextLine();

                    ArrayList<MyuserDTO>  myuserList = client.getRecordsbyAddress(address1);

                    if(myuserList != null){
                    for(int i=0; i < myuserList.size();i++){
                        System.out.println(myuserList.get(i).getName());
                    }
                    }
                    else
                    {
```

edjee.MyuserAppClient > Menu > while (option != 5) > switch (option) > case 4:

Output

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

Source History

```
                    }
                    else
                    {
                        System.out.println("Its null!");
                    }
                    break;


                }
            }//end while

        }


        public void showCreateResult(boolean result, MyuserDTO myuserDTO) {
            if (result) {
                System.out.println("Record with primary key " + myuserDTO.getUserid()
                        + " has been created in the database table.");
            } else {
                System.out.println("Record with primary key " + myuserDTO.getUserid()
                        + " could not be created in the database table!");
            }
        }

        public Boolean createRecord(MyuserDTO myuserDTO) {
            return myuserFacade.createRecord(myuserDTO);
        }

         public MyuserDTO getRecord(String userid) {
            return myuserFacade.getRecord(userid);
        }

        public boolean updateRecord(MyuserDTO myuserDTO) {
            return myuserFacade.updateRecord(myuserDTO);
        }

        public boolean deleteRecord(String userid) {
            return myuserFacade.deleteRecord(userid);
        }

        public ArrayList<MyuserDTO> getRecordsbyAddress(String address) {
```

edjee.MyuserAppClient > Menu > while (option != 5) > switch (option) > case 4:

Output

Java DB Database Process | GlassFish Server 5.1.0 | ED-JEE-DTO-appclient (run)

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

Source History

```java
    public void showCreateResult(boolean result, MyuserDTO myuserDTO) {
        if (result) {
            System.out.println("Record with primary key " + myuserDTO.getUserid()
                    + " has been created in the database table.");
        } else {
            System.out.println("Record with primary key " + myuserDTO.getUserid()
                    + " could not be created in the database table!");
        }
    }

    public Boolean createRecord(MyuserDTO myuserDTO) {
        return myuserFacade.createRecord(myuserDTO);
    }

     public MyuserDTO getRecord(String userid) {
        return myuserFacade.getRecord(userid);
    }

    public boolean updateRecord(MyuserDTO myuserDTO) {
        return myuserFacade.updateRecord(myuserDTO);
    }

    public boolean deleteRecord(String userid) {
        return myuserFacade.deleteRecord(userid);
    }

    public ArrayList<MyuserDTO> getRecordsbyAddress(String address) {
        return myuserFacade.getRecordsByAddress(address);
    }

}
```

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

```java
    public void showCreateResult(boolean result, MyuserDTO myuserDTO) {
        if (result) {
            System.out.println("Record with primary key " + myuserDTO.getUserid()
                    + " has been created in the database table.");
        } else {
            System.out.println("Record with primary key " + myuserDTO.getUserid()
                    + " could not be created in the database table!");
        }
    }

    public boolean createRecord(MyuserDTO myuserDTO) {
        return mydb.createRecord(myuserDTO);
    }

    //can also directly call it with client in the main code, but doing it like this to follow the format given in createRecord for good practise:
    public MyuserDTO getRecord(String userid) {
        return mydb.getRecord(userid);
    }

    public boolean updateRecord(MyuserDTO myuserDTO) {
        return mydb.updateRecord(myuserDTO);
    }

    public boolean deleteRecord(String userid) {
        return mydb.deleteRecord(userid);
    }
}
```

Step 3 outputs:

```
ed.jpa.MyuserApp > Menu >
Output
Java DB Database Process | ED-JPA (run) | SQL 24 execution | SQL 25 execution | SQL 26 execution

Type the number to perform the desired action:
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Exit

Please select an option (1-4): 1
Enter the UserID:
0000069
UserID is not found
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Exit

Please select an option (1-4): 1
Enter the UserID:
000006
000006
David Lee
0123456789
654321
Swinburne EN510g
dlee@swin.edu.au
What is my name?
David
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Exit

Please select an option (1-4): 2
Enter record details to be updated:
UserId:
00004
Name:
blah
Password:
d
Email:
d
Phone:
d
```

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

Output

Java DB Database Process | ED-JPA (run) | SQL 24 execution | SQL 25 execution | SQL 26 execution

```
Please select an option (1-4): 2
Enter record details to be updated:
UserId:
00004
Name:
blah
Password:
d
Email:
d
Phone:
d
Address:
d
SECQN:
d
SECAns:
d
Failed to update
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Exit

Please select an option (1-4): 2
Enter record details to be updated:
UserId:
000006
Name:
s
Password:
s
Email:
s
Phone:
s
Address:
s
SECQN:
s
SECAns:
s
```

Output

Java DB Database Process | ED-JPA (run) | SQL 24 execution | SQL 25 execution | SQL 26 execution

```
UserId:
000006
Name:
s
Password:
s
Email:
s
Phone:
s
Address:
s
SECQN:
s
SECAns:
s
Updated Successfully
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Exit

Please select an option (1-4): 3
Enter the UserID:
000000006
Failed to delete
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Exit

Please select an option (1-4): 3
Enter the UserID:
000001
Deleted Successfully
1: Get a record for a certain UserID
2: Update a Record
3: Delete a Record
4: Exit

Please select an option (1-4): 4
BUILD SUCCESSFUL (total time: 1 minute 43 seconds)
```

Name: S M Ragib Rezwan
ID: 103172423
Tutor: Wei Lai

4.1) Here myuser is the DAO object and thus is the one directly accessing the database when the object is created. On the other hand MyuserFacade is just a bean type class. So myuser is responsible for the ORM work

4.2) basically this means that for each session bean, there is always a certain number of bean instances available for use in the pool. So, when a client sends a request, the pool can quickly be used to fulfill the request, and when it's over, the instance can be returned back to pool for later reuse. So the same bean instance can be used to fulfill the request of lots of clients later on, which in turn make it scalable.