

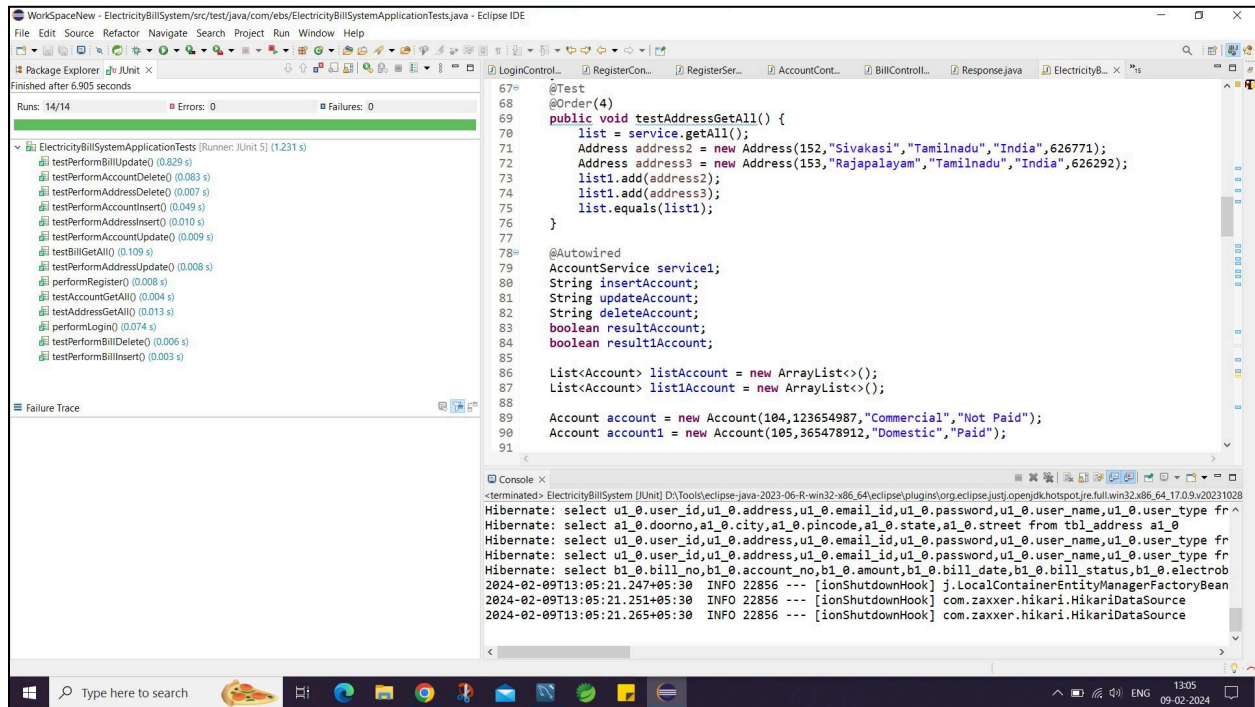
Employee Name : Dharshana

Employee Id : 11953

Electricity Billing System :

JUnit :

Console Output :

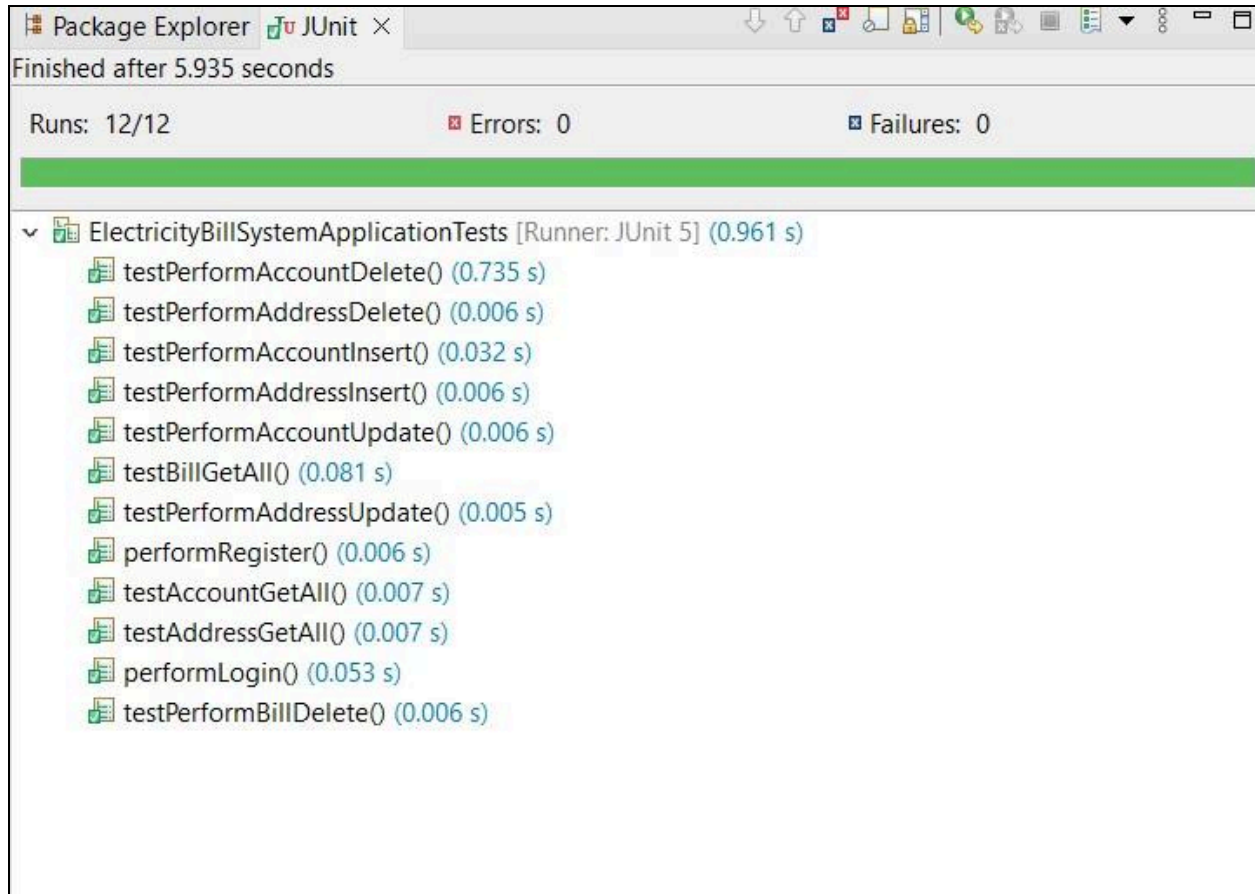


The screenshot displays the Eclipse IDE interface. The top toolbar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Package Explorer on the left shows the project structure, with the 'ElectricityBillingSystemApplicationTests' class selected. The Run console shows the test results for the JUnit 5 (1-231) tests, indicating that all tests passed successfully. The console output includes the following test results:

- testPerformBillUpdate() (0.829 s)
- testPerformAccountDelete() (0.083 s)
- testPerformAddressDelete() (0.007 s)
- testPerformAccountInsert() (0.049 s)
- testPerformAddressInsert() (0.010 s)
- testPerformAccountUpdate() (0.009 s)
- testBillGetAll() (0.109 s)
- testPerformAddressUpdate() (0.008 s)
- performRegister() (0.008 s)
- testAccountGetAll() (0.004 s)
- testAddressGetAll() (0.013 s)
- performLogin() (0.074 s)
- testPerformBillDelete() (0.006 s)
- testPerformBillInsert() (0.003 s)

The Failure Trace section is empty, indicating no test failures. The Console window shows the following output:

```
<terminated> ElectricityBillingSystem [JUnit] D:\Tools\ eclipse -java-2023-06-R-win32-x86_64\ eclipse\ plugins\ org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64.17.0.9.v20231028
Hibernate: select u1_0.user_id,u1_0.address,u1_0.email_id,u1_0.password,u1_0.user_name,u1_0.user_type fr
Hibernate: select a1_0.doorno,a1_0.city,a1_0.pincod,a1_0.state,a1_0.street from tbl_address a1_0
Hibernate: select u1_0.user_id,u1_0.address,u1_0.email_id,u1_0.password,u1_0.user_name,u1_0.user_type fr
Hibernate: select u1_0.user_id,u1_0.address,u1_0.email_id,u1_0.password,u1_0.user_name,u1_0.user_type fr
Hibernate: select b1_0.bill_no,b1_0.account_no,b1_0.amount,b1_0.bill_date,b1_0.bill_status,b1_0.electroba
2024-02-09T13:05:21.247+05:30 INFO 22856 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean
2024-02-09T13:05:21.251+05:30 INFO 22856 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource
2024-02-09T13:05:21.265+05:30 INFO 22856 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource
```



Source Code :

```
package com.ebs;
import static org.testng.Assert.assertEquals;
import java.util.ArrayList;
import java.util.List;
import org.junit.jupiter.api.Order;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import com.ebs.bean.Account;
import com.ebs.bean.Address;
import com.ebs.bean.Bill;
import com.ebs.bean.Response;
import com.ebs.bean.User;
import com.ebs.service.AccountService;
```

```
import com.ebs.service.AddressService;
import com.ebs.service.BillService;
import com.ebs.service.LoginService;
import com.ebs.service.RegisterService;
@SpringBootTest
class ElectricityBillSystemApplicationTests {

    @Autowired
    AddressService service;
    String insert;
    String update;
    String delete;
    String result;
    String result1;

    List<Address> list = new ArrayList<>();
    List<Address> list1 = new ArrayList<>();

    Address address = new Address(111,"Viswai","Tamilnadu","India",626189);
    Address address1 = new Address(222,"Sivakasi","Tamilnadu","India",626119);

    Address address2;
    Address address3;

    @Test
    @Order(1)
    public void testPerformAddressInsert() {
        insert = "Inserted";
        result = service.insertAddress(address);
    }
    @Test
    @Order(2)
    public void testPerformAddressUpdate() {
        update = "Updated";
        result = service.updateAddress(address1);
    }
    @Test
    @Order(3)
    public void testPerformAddressDelete() {
        delete = "Deleted";
    }
}
```

```

        result = service.deleteAddress(124);
    }
    @Test
    @Order(4)
    public void testAddressGetAll() {
        list = service.getAll();
        Address address2 = new
Address(152,"Sivakasi","Tamilnadu","India",626771);
        Address address3 = new
Address(153,"Rajapalayam","Tamilnadu","India",626292);
        list1.add(address2);
        list1.add(address3);
        list.equals(list1);
    }

    @Autowired
    AccountService service1;
    String insertAccount;
    String updateAccount;
    String deleteAccount;
    boolean resultAccount;
    boolean result1Account;

    List<Account> listAccount = new ArrayList<>();
    List<Account> list1Account = new ArrayList<>();

    Account account = new Account(104,123654987,"Commercial","Not Paid");
    Account account1 = new Account(105,365478912,"Domestic","Paid");

    Account account2;
    Account account3;

    @Test
    @Order(5)
    public void testPerformAccountInsert() {
        insertAccount = "Inserted";
        resultAccount = service1.insertAccount(account);
        assertEquals(true, resultAccount);
    }

    @Test
    @Order(6)

```

```

public void testPerformAccountUpdate() {
    updateAccount = "Updated";
    resultAccount = service1.insertAccount(account1);
    assertEquals(true, resultAccount);
}

@Test
@Order(7)
public void testPerformAccountDelete() {
    deleteAccount = "Deleted";
    resultAccount = service1.deleteAccount(3);
    assertEquals(true, resultAccount);
}

@Test
@Order(8)
public void testAccountGetAll() {
    listAccount = service1.getAll();
    Account account2 = new Account (104,123654987,"Commercial","Not
Paid");
    Account account3 = new Account(105,365478912,"Domestic","Paid");

    list1Account.add(account2);
    list1Account.add(account3);
    listAccount.equals(list1Account);
}

@Autowired
BillService service2;
String insertBill;
String updateBill;
String deleteBill;
String resultBill;
boolean result1Bill;

List<Bill> listBill = new ArrayList<>();
List<Bill> list1Bill = new ArrayList<>();

Bill bill = new Bill(03,1003,50000,"05-02-24","Pending",365478912);
Bill bill1 = new Bill(05,1003,10000,"05-02-24","Paid",123654795);

Bill bill2;

```

Bill bill3;

@Test

@Order(9)

```
public void testPerformBillInsert() {  
    insertBill = "Inserted";  
    resultBill = service2.insertBill(bill);  
    assertEquals(true, resultBill);  
    //return insert;  
}
```

@Test

@Order(10)

```
public void testPerformBillUpdate() {  
    updateBill = "Updated";  
    resultBill =service2.updateBill(bill1);  
    assertEquals(true, resultBill);  
    //return update;  
}
```

@Test

@Order(11)

```
public void testPerformBillDelete() {  
    deleteBill = "Deleted";  
    resultBill = service2.deleteBill(3);  
}
```

@Test

@Order(12)

```
public void testBillGetAll() {  
    listBill = service2.getAll();  
    Bill bill = new Bill(03,1003,50000,"05-02-24","Pending",365478912);  
    Bill bill1 = new Bill(03,1003,50000,"05-02-24","Pending",123654795);  
  
    list1Bill.add(bill2);  
    list1Bill.add(bill3);  
    listBill.equals(list1Bill);  
}
```

@Autowired

LoginService loginservice;

```

    @Autowired
    RegisterService registerservice;

    boolean res;
    boolean val;
    User log = new User (4,"jeni", "jeni@gmail.com","2/353
South","User","jeni@123");

    @Test
    @Order(13)
    void performLogin() {
        Response res1 = new Response("User Success");
        Response res = loginservice.login(log);
        res1.equals(res);

    }

    @Test
    @Order(14)
    void performRegister() {
        val=true;
        res=registerservice.Register(log);
        assertEquals(val, res);
    }

}

```