```java
package il.ac.hit.CashFlowManagement.test;

import il.ac.hit.CashFlowManagement.exception.JDBCDataBaseException;
import il.ac.hit.CashFlowManagement.model.JDBCDataBase;
import il.ac.hit.CashFlowManagement.model.UserLogic;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import java.sql.SQLException;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;

public class UserLogicTest {
    UserLogic userLogic = new UserLogic();
    String userName = "testUserName", password = "testPassword", country = "testCountry", gander = "testGander";

    @Before
    public void setUp() {
        userLogic.addUser(userName, password, country, gander);
    }

    @After
    public void tearDown() throws JDBCDataBaseException, SQLException {
        while(userLogic.checkIfExist("userName") || userLogic.checkIfExist(userName)) {
            JDBCDataBase.getInstance().getStatement().executeUpdate("DELETE FROM Registered_Users WHERE UserName = 'testUserName'");
            JDBCDataBase.getInstance().getStatement().executeUpdate("DELETE FROM Registered_Users WHERE UserName = 'userName'");
        }
    }

    @Test
    public void checkIfUserExist() throws JDBCDataBaseException, SQLException {

        assertEquals(false, userLogic.checkIfExist("userName"));
        assertTrue(userLogic.checkIfExist(userName));
    }

    @Test
    public void addUser() {
        assertEquals(false, userLogic.checkIfExist("userName"));
        userLogic.addUser("userName", password, country, gander);
        assertTrue(userLogic.checkIfExist("userName"));
    }

    @Test
    public void checkUserPassword() {
        assertEquals(true, userLogic.checkUserPassword(userName, 
password));
        assertEquals(false, userLogic.checkUserPassword(userName, 
password + "1"));
    }
}

```

```java
package il.ac.hit.CashFlowManagement.test;

import il.ac.hit.CashFlowManagement.exception.JDBCDataBaseException;
import il.ac.hit.CashFlowManagement.model.JDBCDataBase;
import org.apache.log4j.BasicConfigurator;
import org.junit.Test;

import static junit.framework.TestCase.assertNotNull;

public class JDBCDataBaseTest {
    public JDBCDataBaseTest(){
        BasicConfigurator.configure();
    }

    @Test
    public void getInstance() {
        try {
            assertNotNull(JDBCDataBase.getInstance());
        } catch (JDBCDataBaseException e) {
            e.printStackTrace();
        }
    }

    @Test
    public void getStatement() {
        try {
            assertNotNull(JDBCDataBase.getInstance().getStatement());
        } catch (JDBCDataBaseException e) {
            e.printStackTrace();
        }
    }

    @Test
    public void getConnection() {
        try {
            assertNotNull(JDBCDataBase.getInstance().getConnection
());
        } catch (JDBCDataBaseException e) {
            e.printStackTrace();
        }
    }

}

```

```java
package il.ac.hit.CashFlowManagement.test;

import il.ac.hit.CashFlowManagement.model.Expense;
import il.ac.hit.CashFlowManagement.model.ExpensesLogic;
import il.ac.hit.CashFlowManagement.model.JDBCDataBaseLogic;
import il.ac.hit.CashFlowManagement.view.LoginForm;
import il.ac.hit.CashFlowManagement.viewmodel.IViewModel;
import il.ac.hit.CashFlowManagement.viewmodel.ManagementViewModel;
import org.apache.log4j.BasicConfigurator;
import org.junit.After;
import org.junit.Test;

import java.sql.ResultSet;

import static org.junit.Assert.assertEquals;

public class ExpensesLogicTest {
    private JDBCDataBaseLogic dataBaseLogic = JDBCDataBaseLogic.
getInstance();
    private String expenseDate = "01/01/2020", expenseClassification
 = "testClassification", expenseDetails = "testDetails";
    private double expenseCost = 5.55;
    private IViewModel viewModel;
    private String nameOfTable;
    private ExpensesLogic expensesLogic;
    private Expense expense = new Expense(expenseDate,
expenseClassification, expenseDetails, expenseCost);

    public ExpensesLogicTest(){
        BasicConfigurator.configure();
        LoginForm.username = "testUserName";
        viewModel = ManagementViewModel.getInstance();
        nameOfTable = LoginForm.username.toUpperCase() + "_Expenses".
toUpperCase();
        expensesLogic = new ExpensesLogic();
    }

    @After
    public void tearDown() throws Exception {
        dataBaseLogic.removeTable(nameOfTable);
    }

    @Test
    public void addExpense() throws Exception {
        expensesLogic.addExpense(expense);
        ResultSet rs = expensesLogic.getAllUserExpenses();
        rs.next();
        String date = rs.getString("Date");
        String cost = rs.getString("Cost");
        String classification = rs.getString("Classification");
        String details = rs.getString("Details");
        double costDouble = Double.parseDouble(cost);

        assertEquals(expenseDate, date);
        assertEquals((long)expenseCost, (long)costDouble);
        assertEquals(expenseClassification, classification);
        assertEquals(expenseDetails, details);
    }

    @Test
    public void getAllUserExpenses() throws Exception {
        int expensesCounter = 0;
        expensesLogic.addExpense(expense);
        ResultSet rs = expensesLogic.getAllUserExpenses();
        while (rs.next()){
            expensesCounter++;
        }
        assertEquals(1, expensesCounter);
    }
}

```

```java
package il.ac.hit.CashFlowManagement.test;


import il.ac.hit.CashFlowManagement.exception.JDBCDataBaseLogicException;
import il.ac.hit.CashFlowManagement.model.JDBCDataBaseLogic;
import org.apache.log4j.BasicConfigurator;

import static org.hamcrest.CoreMatchers.is;
import static org.junit.Assert.*;


public class JDBCDataBaseLogicTest {
    private JDBCDataBaseLogic dataBaseLogic = JDBCDataBaseLogic.getInstance();
    private final String iNameOfTable = "Test";

    public JDBCDataBaseLogicTest(){
        BasicConfigurator.configure();
    }

    @org.junit.Before
    public void setUp() throws Exception {
        dataBaseLogic.createTableIfNotExist(iNameOfTable,"Password", "varchar(300)");
    }


    @org.junit.After
    public void tearDown() throws Exception {
        dataBaseLogic.removeTable(iNameOfTable);
    }

    @org.junit.Test
    public void createTableIfNotExist() throws Exception {
        String tableName = "CreateTableTest";

        assertEquals(true,dataBaseLogic.createTableIfNotExist(tableName,"Password", "varchar(300)"));
        assertEquals(false,dataBaseLogic.createTableIfNotExist(tableName,"Password", "varchar(300)"));
        dataBaseLogic.removeTable(tableName);
        try{
            dataBaseLogic.createTableIfNotExist(iNameOfTable,"Password");
            fail("Expected exception has not been thrown");
        } catch (Exception e) {
            assertThat(e.getMessage(), is("Please enter the parameters as follows: Param1Name ,Param1Type, Param2Name ,Param2Type .... , ParamName ,ParamType"));
        }
    }

    @org.junit.Test
    public void insertValue() throws Exception {
        String params = "'" + "134" + "'";

        dataBaseLogic.insertValue(iNameOfTable,params);
        try {
            dataBaseLogic.insertValue(iNameOfTable, null);
            // If the exception is not thrown, the test will fail
            fail("Expected exception has not been thrown");
        } catch (Exception e) {
        assertThat(e.getMessage(), is("Argument for @NotNull parameter 'iParameters' of il/ac/hit/CashFlowManagement/model/JDBCDataBaseLogic.insertValue must not be null"));
        }
    }

    @org.junit.Test
    public void query() throws JDBCDataBaseLogicException {
        assertNotNull(dataBaseLogic.query("select * from " + iNameOfTable));

        try {
            assertNotNull(dataBaseLogic.query("select * from " + iNameOfTable + "1"));
            fail("Expected exception has not been thrown");
        } catch (Exception e) {
            assertThat(e.getMessage(), is("query execution failed Table/View 'TEST1' does not exist."));
        }

        try {
            dataBaseLogic.query("create table TEST (Password varchar(300))");
            fail("Expected exception has not been thrown");
        } catch (Exception e) {
            assertThat(e.getMessage(), is("query execution failed Table/View 'TEST' already exists in Schema 'APP'."));
        }
    }

    @org.junit.Test
    public void removeTable() throws JDBCDataBaseLogicException {
        String tableName = "RemoveTestTable";

        dataBaseLogic.createTableIfNotExist(tableName,"Password", " varchar(300)");
        dataBaseLogic.removeTable(tableName);

        try {
            dataBaseLogic.removeTable(tableName+"1");
            fail("Expected exception has not been thrown");
        } catch (Exception e) {
            assertThat(e.getMessage(), is("failed to remove RemoveTestTable1 from DB"));
        }
    }

    @org.junit.Test
    public void getInstance() throws JDBCDataBaseLogicException {
        assertNotNull(JDBCDataBaseLogic.getInstance());
    }

}
```

```java
package il.ac.hit.CashFlowManagement.test;

import il.ac.hit.CashFlowManagement.exception.FormCastingException;
import il.ac.hit.CashFlowManagement.exception.GetAllUserExpensesException;
import il.ac.hit.CashFlowManagement.exception.JDBCDataBaseException;
import il.ac.hit.CashFlowManagement.model.Expense;
import il.ac.hit.CashFlowManagement.model.ExpensesLogic;
import il.ac.hit.CashFlowManagement.model.JDBCDataBase;
import il.ac.hit.CashFlowManagement.model.UserLogic;
import il.ac.hit.CashFlowManagement.view.LoginForm;
import il.ac.hit.CashFlowManagement.viewmodel.ManagementViewModel;
import org.junit.After;
import org.junit.Before;
import org.junit.Test;

import java.sql.ResultSet;
import java.sql.SQLException;

import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertNotNull;

public class ManagementViewModelTest {
    private ManagementViewModel viewModel;
    private UserLogic userLogic = new UserLogic();
    private String userName = "testUserName", password = "testPassword", country = "testCountry", gander = "testGander", nameOfTable;
    private String expenseDate = "01/01/2020", expenseClassification = "testClassification", expenseDetails = "testDetails";
    private double expenseCost = 5.55;
    private Expense expense = new Expense(expenseDate, expenseClassification, expenseDetails, expenseCost);


    public ManagementViewModelTest(){
        LoginForm.username = "testUserName";
        viewModel = ManagementViewModel.getInstance();
        viewModel.setExpensesLogic(new ExpensesLogic());
        nameOfTable = LoginForm.username.toUpperCase() + "_Expenses".toUpperCase();
    }

    @Before
    public void setUp() {
        userLogic.addUser(userName, password, country, gander);
    }

    @After
    public void tearDown() throws JDBCDataBaseException, SQLException {
        while(userLogic.checkIfExist("userName") || userLogic.checkIfExist(userName)) {
            JDBCDataBase.getInstance().getStatement().executeUpdate("DELETE FROM Registered_Users WHERE UserName = 'testUserName'");
            JDBCDataBase.getInstance().getStatement().executeUpdate("DELETE FROM Registered_Users WHERE UserName = 'userName'");
        }
    }

    @Test
    public void getInstance() {
        assertNotNull(ManagementViewModel.getInstance());
    }

    @Test
    public void verifyUser() {
        assertEquals(true, viewModel.verifyUser(userName, password));
    }

    @Test
    public void register() {
        assertEquals(true, viewModel.register("userName", password, country, gander));
        assertEquals(false, viewModel.register(userName, password, country, gander));
    }

    @Test
    public void getAllExpenses() throws GetAllUserExpensesException, SQLException, FormCastingException {
        ResultSet rs = viewModel.getAllExpenses();
        int counter = 0;

        while (rs.next()){
            counter++;
        }
        assertEquals(0, counter);
    }
}

```