

CS 331 (Software Engineering Lab) - Assignment 7

Part A: Data Access Layer (DAL) Implementation

Step 1: Database Creation and Table Design

In this step, we define the required database schema. Assuming we are using PostgreSQL, we can create tables such as:

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(100),  
    email VARCHAR(100) UNIQUE NOT NULL,  
    password VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE ev_stations (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(100),  
    latitude DECIMAL(9,6),  
    longitude DECIMAL(9,6),  
    capacity INT
```

```
);
```

```
CREATE TABLE potential_locations (  
    id SERIAL PRIMARY KEY,  
    latitude DECIMAL(9,6),  
    longitude DECIMAL(9,6),  
    score FLOAT  
);
```

Step 2: Implementing the Data Access Layer (DAL)

A typical DAL implementation in Python using psycopg2:

```
import psycopg2
```

```
class Database:
```

```
    def __init__(self):  
        self.conn = psycopg2.connect(  
            dbname="your_db",  
            user="your_user",  
            password="your_password",  
            host="localhost",  
            port="5432"
```

```

    )

    self.cursor = self.conn.cursor()

def fetch_all_stations(self):
    self.cursor.execute("SELECT * FROM ev_stations;")
    return self.cursor.fetchall()

def insert_station(self, name, latitude, longitude, capacity):
    self.cursor.execute(
        "INSERT INTO ev_stations (name, latitude, longitude, capacity)
VALUES (%s, %s, %s, %s);",
        (name, latitude, longitude, capacity)
    )
    self.conn.commit()

def close(self):
    self.cursor.close()
    self.conn.close()

```

This class provides methods to interact with the database without exposing raw queries to the application.

Part B: White Box & Black Box Testing

1. White Box Testing

White Box Testing involves checking internal logic, loops, and paths.

Test Case Examples for White Box Testing:

Test Case ID	Function	Test Description	Expected Output
WB-01	fetch_all_stations()	Check if the method retrieves data	Returns list of tuples
WB-02	insert_station()	Check if new station gets inserted	New record appears in DB
WB-03	insert_station()	Check SQL Injection prevention	Query should fail or sanitize input
WB-04	fetch_all_stations()	Handle empty table	Returns an empty list

Example White Box Test Script:

```
import unittest

from DAL import Database # Assuming Database class is saved in
DAL.py

class TestWhiteBox(unittest.TestCase):

    def setUp(self):

        self.db = Database()
```

```
def test_fetch_all_stations(self):
    result = self.db.fetch_all_stations()
    self.assertIsInstance(result, list) # Should return a list

def test_insert_station(self):
    self.db.insert_station("Test Station", 40.7128, -74.0060, 10)
    result = self.db.fetch_all_stations()
    self.assertTrue(any("Test Station" in row for row in result))

def tearDown(self):
    self.db.close()

if __name__ == "__main__":
    unittest.main()
```

2. Black Box Testing

Black Box Testing involves verifying the input/output without knowing the code structure.

Test Case Examples for Black Box Testing:

Test Case ID	Input	Expected Output
BB-01	Valid user credentials	Successful login
BB-02	Invalid user credentials	Error message
BB-03	Empty station list	Returns empty response
BB-04	Adding a new station	Station appears in the list
BB-05	Requesting a non-existent station	Returns "Not Found"

Example Black Box Test Script (API Testing with Flask and Requests):

```
import requests
```

```
import unittest
```

```
BASE_URL = "http://127.0.0.1:5000"
```

```
class TestBlackBox(unittest.TestCase):
```

```
    def test_get_stations(self):
```

```
        response = requests.get(f"{BASE_URL}/stations")
```

```
        self.assertEqual(response.status_code, 200)
```

```
        self.assertIsInstance(response.json(), list)
```

```
def test_add_station(self):  
    new_station = {  
        "name": "New EV Station",  
        "latitude": 37.7749,  
        "longitude": -122.4194,  
        "capacity": 20  
    }  
    response = requests.post(f"{BASE_URL}/stations",  
                             json=new_station)  
    self.assertEqual(response.status_code, 201)  
  
if __name__ == "__main__":  
    unittest.main()
```

Conclusion

- **Part A** covered the creation of a PostgreSQL database and a Python-based DAL implementation.
- **Part B** covered **White Box Testing** (internal logic verification) and **Black Box Testing** (external API testing).