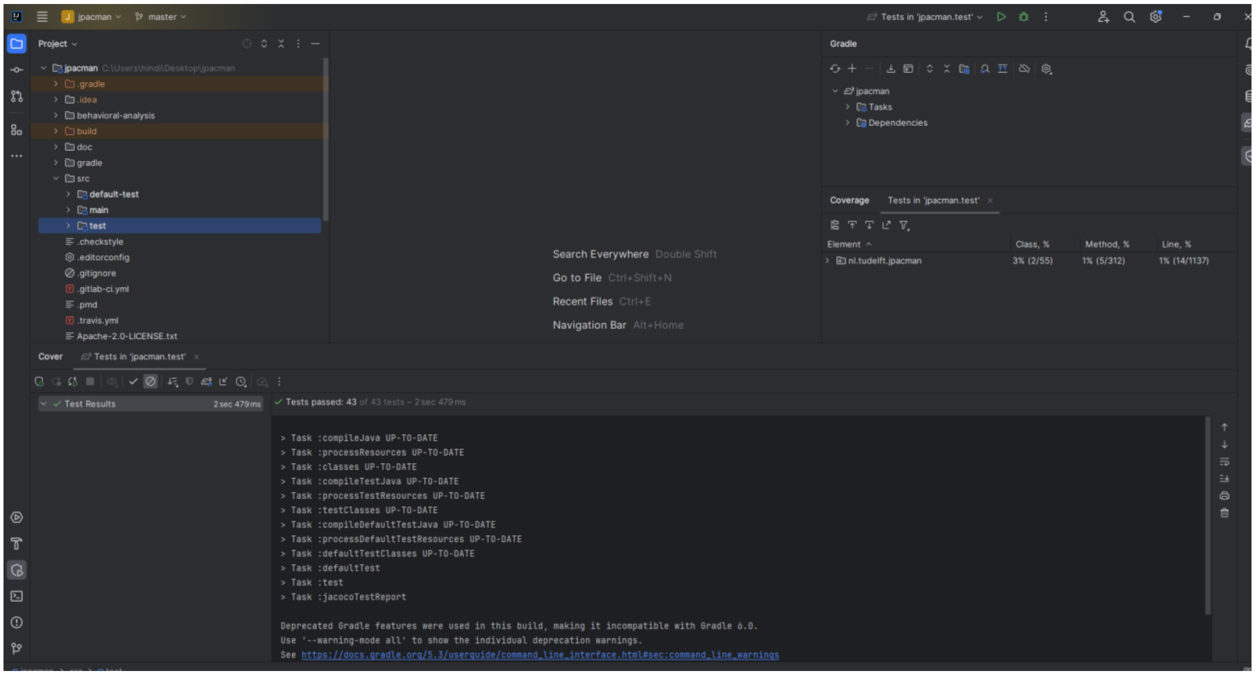
Dynamic Analysis

https://github.com/Intensifiesx/jpacman

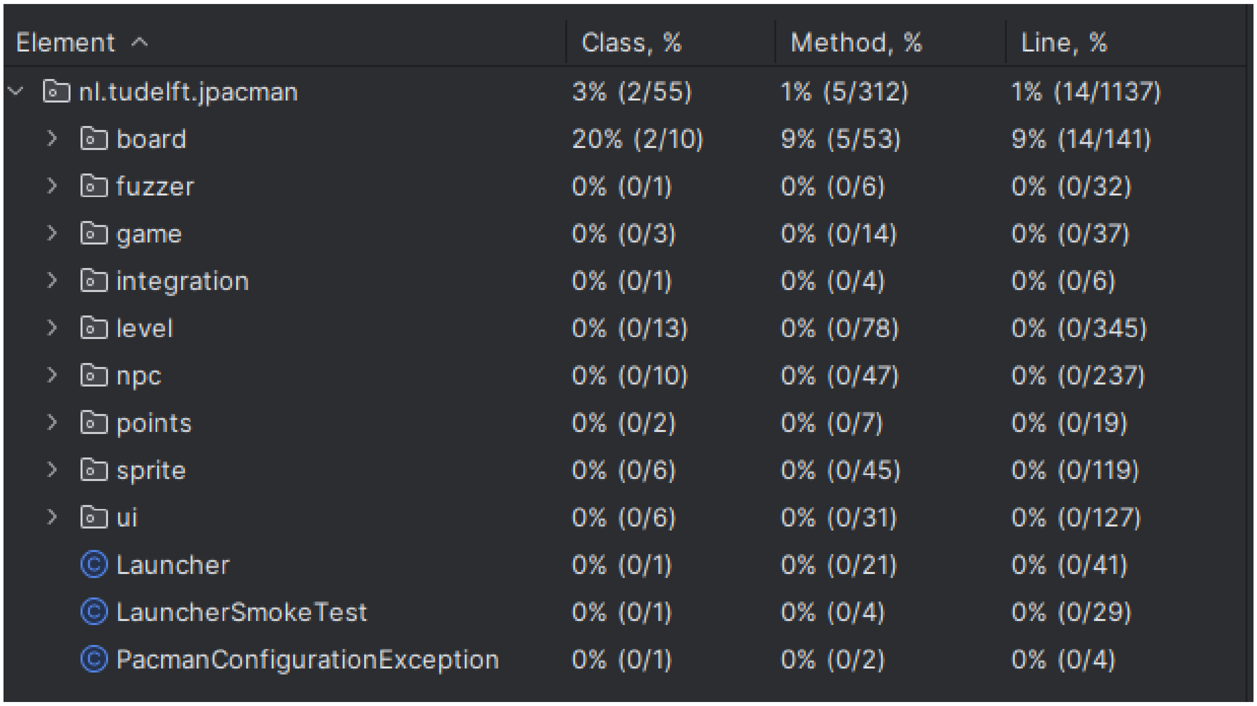
Task 1:



Element ^	Class, %	Method, %	Line, %
nl.tudelft.jpacman	3% (2/55)	1% (5/312)	1% (14/1137)

✓ Tests passed: 43 of 43 tests – 2 sec 479 ms

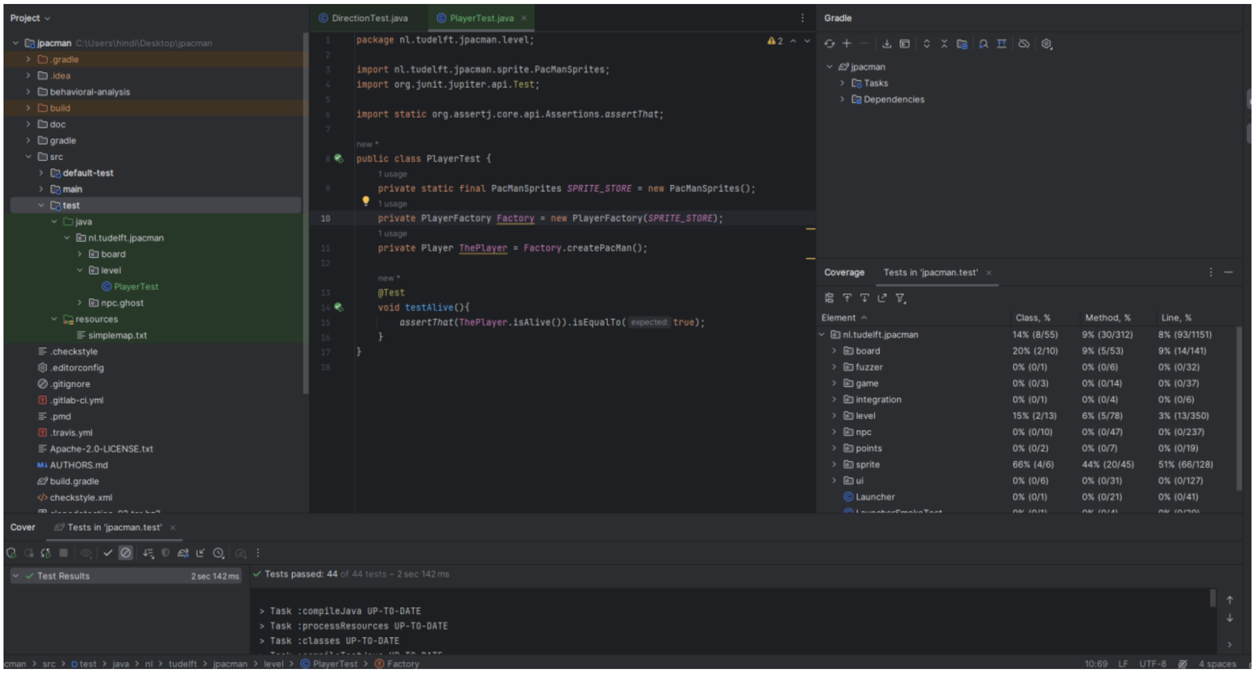
```
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources UP-TO-DATE
> Task :testClasses UP-TO-DATE
> Task :compileDefaultTestJava UP-TO-DATE
> Task :processDefaultTestResources UP-TO-DATE
> Task :defaultTestClasses UP-TO-DATE
> Task :defaultTest
> Task :test
> Task :jacocoTestReport

Deprecated Gradle features were used in this build, making it incompatible with Gradle 6.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/5.3/userguide/command_line_interface.html#sec:command_line_warnings
```



Element ^	Class, %	Method, %	Line, %
nl.tudelft.jpacman	3% (2/55)	1% (5/312)	1% (14/1137)
board	20% (2/10)	9% (5/53)	9% (14/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	0% (0/13)	0% (0/78)	0% (0/345)
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	0% (0/6)	0% (0/45)	0% (0/119)
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

As we can see on the right, the coverage is terrible for classes, methods, and lines.

Task 2:



```java
package nl.tudelft.jpacman.level;

import nl.tudelft.jpacman.sprite.PacManSprites;
import org.junit.jupiter.api.Test;

import static org.assertj.core.api.Assertions.assertThat;

public class PlayerTest {
    private static final PacManSprites SPRITE_STORE = new PacManSprites();
    private PlayerFactory Factory = new PlayerFactory(SPRITE_STORE);
    private Player ThePlayer = Factory.createPacMan();

    @Test
    void testAlive(){
        assertThat(ThePlayer.isAlive()).isEqualTo( expected: true);
    }
}
```

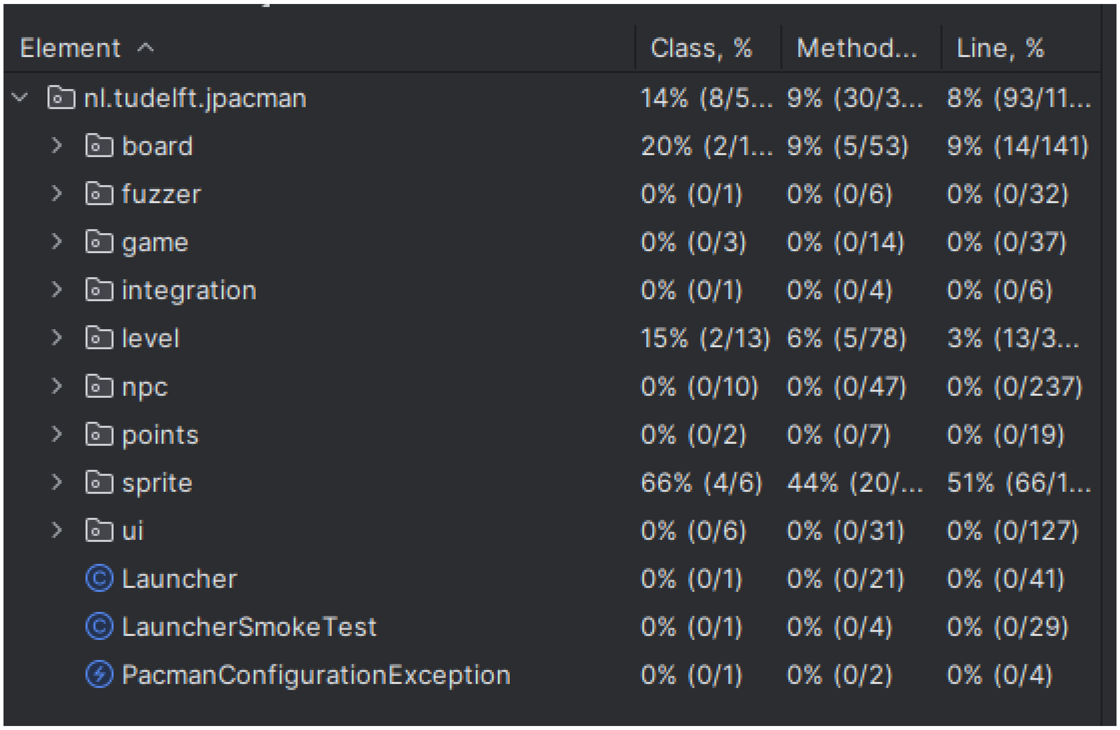
Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	14% (8/55)	9% (30/312)	8% (93/1151)
board	20% (2/10)	9% (5/53)	9% (14/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2/13)	6% (5/78)	3% (13/350)
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4/6)	44% (20/45)	51% (66/128)
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)

Tests passed: 44 of 44 tests – 2 sec 142 ms

```
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
```

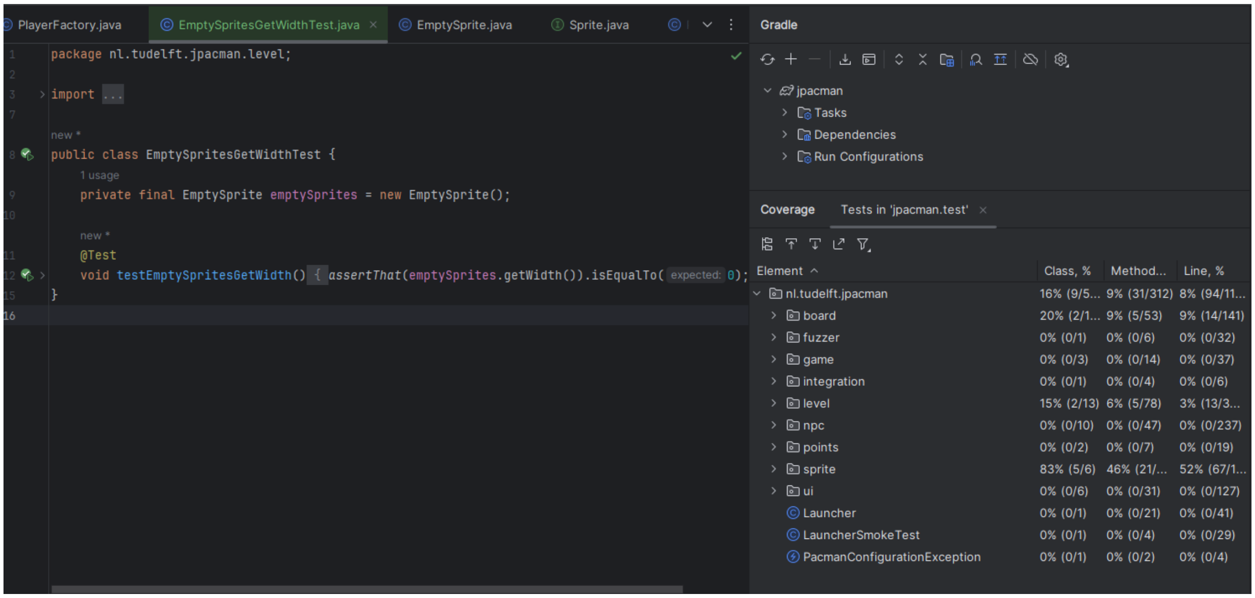
Task 2.1:

Before Testing Method 1



Element	Class, %	Method...	Line, %
nl.tudelft.jpacman	14% (8/5...	9% (30/3...	8% (93/11...
board	20% (2/1...	9% (5/53)	9% (14/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2/13)	6% (5/78)	3% (13/3...
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4/6)	44% (20/...	51% (66/1...
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

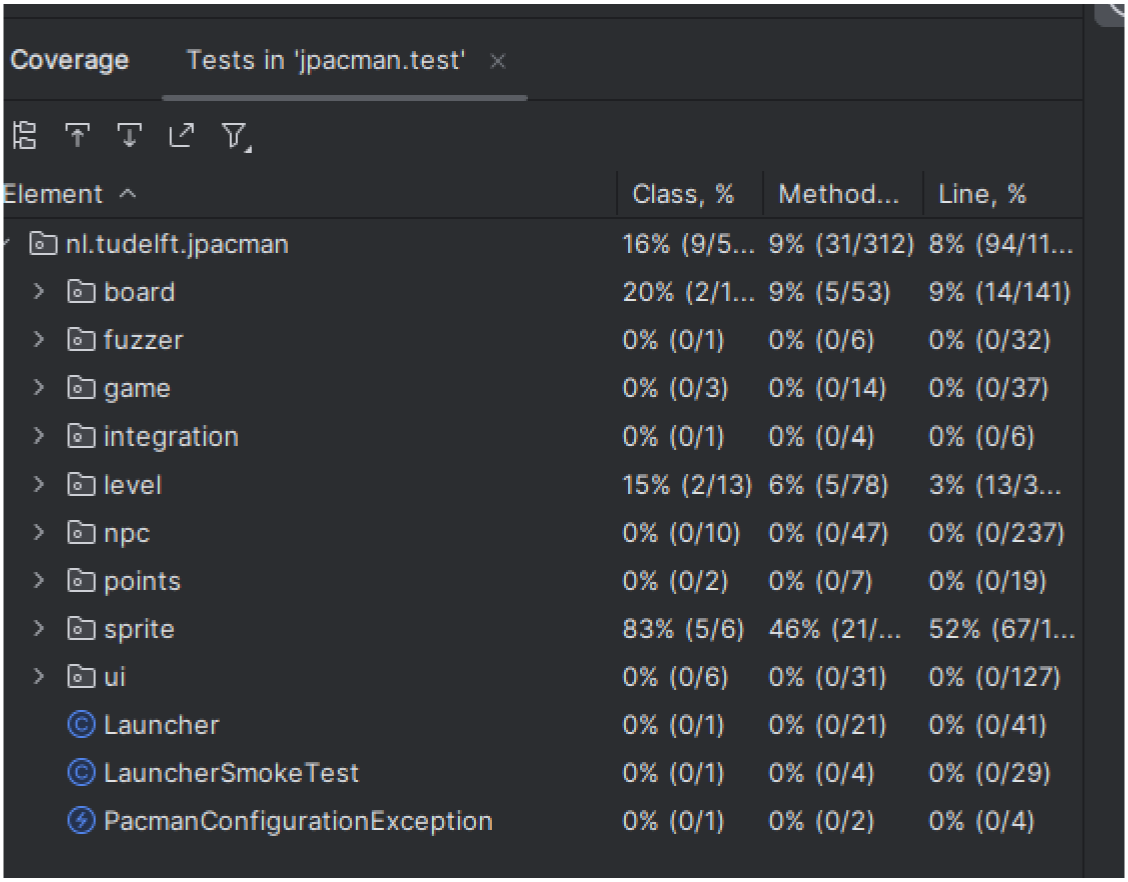
After Testing Method 1: EmptySprite.getWidth()



```java
package nl.tudelft.jpacman.level;

import ...

new *
public class EmptySpritesGetWidthTest {
    1 usage
    private final EmptySprite emptySprites = new EmptySprite();

    new *
    @Test
    void testEmptySpritesGetWidth() { assertThat(emptySprites.getWidth()).isEqualTo( expected: 0);
}

```

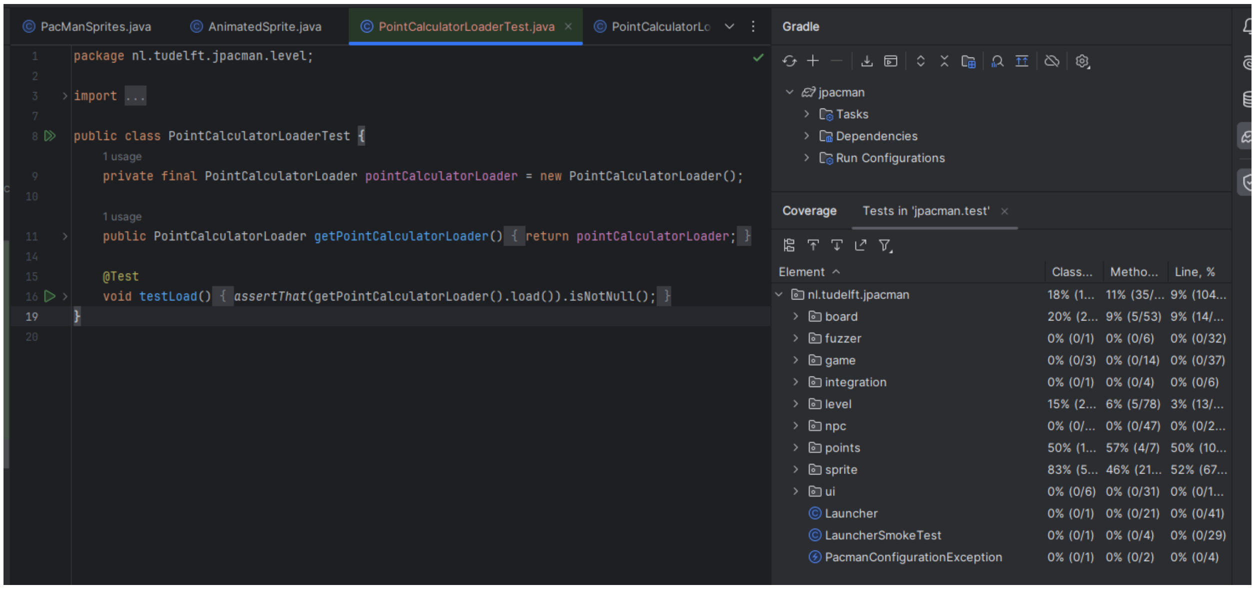
Element	Class, %	Method...	Line, %
nl.tudelft.jpacman	16% (9/5...	9% (31/312)	8% (94/11...
board	20% (2/1...	9% (5/53)	9% (14/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2/13)	6% (5/78)	3% (13/3...
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	83% (5/6)	46% (21/...	52% (67/1...
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

Before Testing Method 2



Element	Class, %	Method...	Line, %
nl.tudelft.jpacman	16% (9/5...	9% (31/312)	8% (94/11...
board	20% (2/1...	9% (5/53)	9% (14/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2/13)	6% (5/78)	3% (13/3...
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	83% (5/6)	46% (21/...	52% (67/1...
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

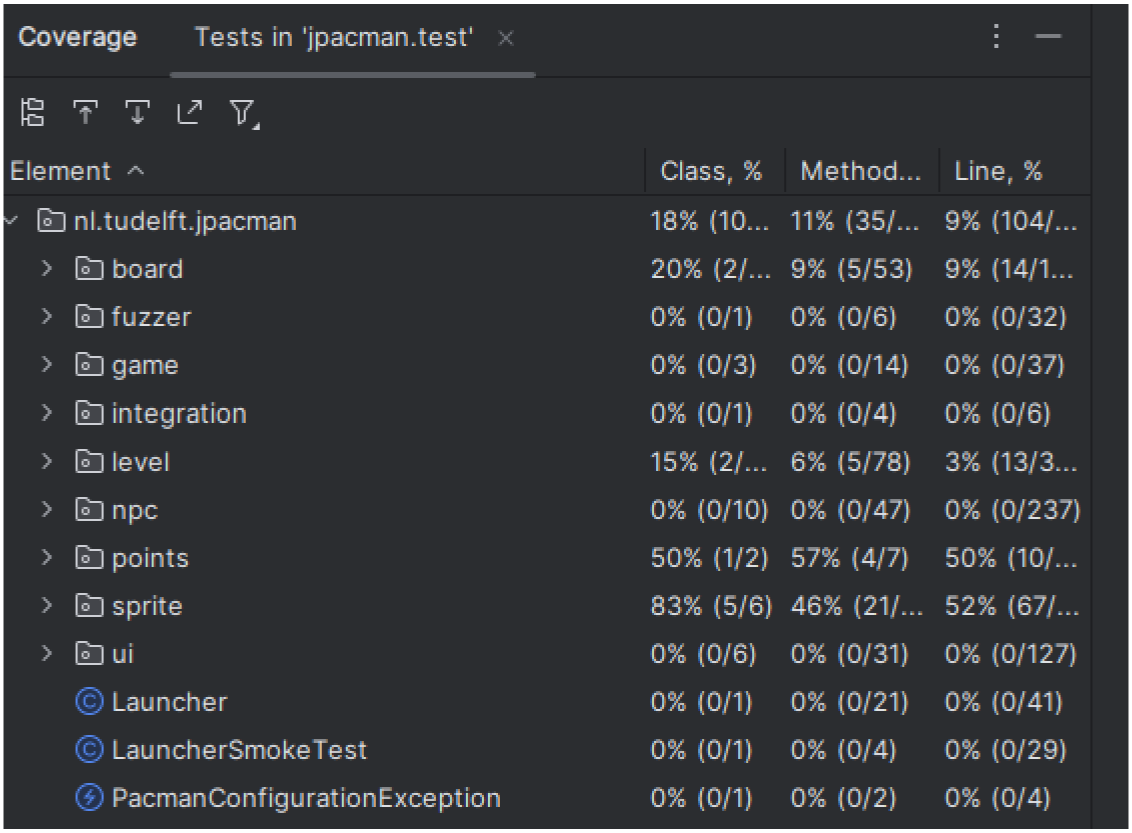
After Testing Method 2: PointCalculatorLoader.load()



```java
package nl.tudelft.jpacman.level;

import ...

public class PointCalculatorLoaderTest {
    1 usage
    private final PointCalculatorLoader pointCalculatorLoader = new PointCalculatorLoader();

    1 usage
    public PointCalculatorLoader getPointCalculatorLoader() { return pointCalculatorLoader; }

    @Test
    void testLoad() { assertThat(getPointCalculatorLoader().load()).isNotNull(); }
}
```

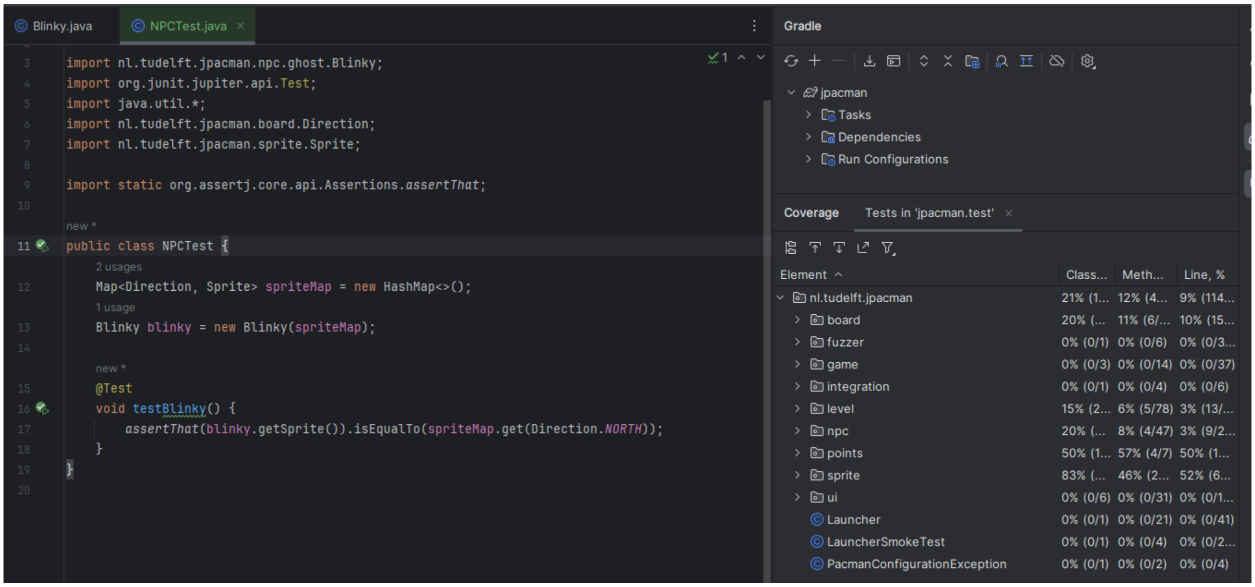
Element	Class...	Metho...	Line, %
nl.tudelft.jpacman	18% (1...	11% (35/...	9% (104...
board	20% (2...	9% (5/53)	9% (14/...
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2...	6% (5/78)	3% (13/...
npc	0% (0/...	0% (0/47)	0% (0/2...
points	50% (1...	57% (4/7)	50% (10...
sprite	83% (5...	46% (21...	52% (67...
ui	0% (0/6)	0% (0/31)	0% (0/1...
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

Before Testing Method 3



Element	Class, %	Method...	Line, %
nl.tudelft.jpacman	18% (10...	11% (35/...	9% (104/...
board	20% (2/...	9% (5/53)	9% (14/1...
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2/...	6% (5/78)	3% (13/3...
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	50% (1/2)	57% (4/7)	50% (10/...
sprite	83% (5/6)	46% (21/...	52% (67/...
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

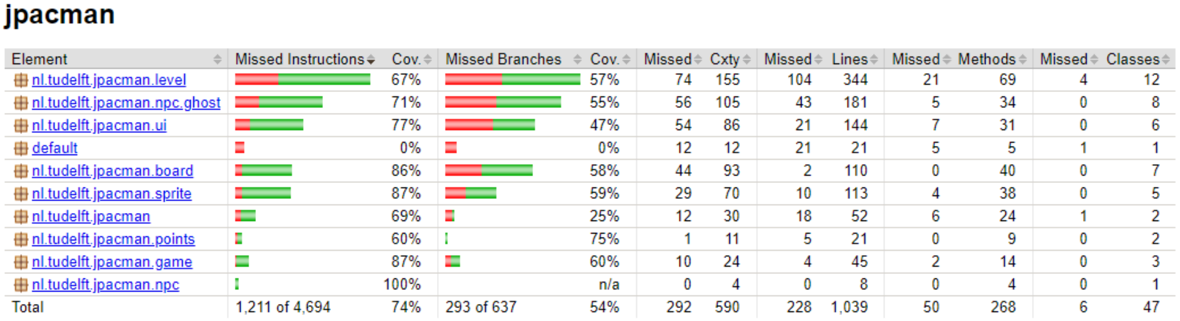
After Testing Method 3: Blinky



```java
import nl.tudelft.jpacman.npc.ghost.Blinky;
import org.junit.jupiter.api.Test;
import java.util.*;
import nl.tudelft.jpacman.board.Direction;
import nl.tudelft.jpacman.sprite.Sprite;

import static org.assertj.core.api.Assertions.assertThat;

new *
public class NPCTest {
    2 usages
    Map<Direction, Sprite> spriteMap = new HashMap<>();
    1 usage
    Blinky blinky = new Blinky(spriteMap);

    new *
    @Test
    void testBlinky() {
        assertThat(blinky.getSprite()).isEqualTo(spriteMap.get(Direction.NORTH));
    }
}
```

Element	Class...	Meth...	Line, %
nl.tudelft.jpacman	21% (1...	12% (4...	9% (114...
board	20% (...	11% (6/...	10% (15...
fuzzer	0% (0/1)	0% (0/6)	0% (0/3...
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2...	6% (5/78)	3% (13/...
npc	20% (...	8% (4/47)	3% (9/2...
points	50% (1...	57% (4/7)	50% (1...
sprite	83% (...	46% (2...	52% (6...
ui	0% (0/6)	0% (0/31)	0% (0/1...
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/2...
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

Task 3:



jpacman

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
nl.tudelft.jpacman.level		67%		57%	74	155	104	344	21	69	4	12
nl.tudelft.jpacman.npc.ghost		71%		55%	56	105	43	181	5	34	0	8
nl.tudelft.jpacman.ui		77%		47%	54	86	21	144	7	31	0	6
default		0%		0%	12	12	21	21	5	5	1	1
nl.tudelft.jpacman.board		86%		58%	44	93	2	110	0	40	0	7
nl.tudelft.jpacman.sprite		87%		59%	29	70	10	113	4	38	0	5
nl.tudelft.jpacman		69%		25%	12	30	18	52	6	24	1	2
nl.tudelft.jpacman.points		60%		75%	1	11	5	21	0	9	0	2
nl.tudelft.jpacman.game		87%		60%	10	24	4	45	2	14	0	3
nl.tudelft.jpacman.npc		100%		n/a	0	4	0	8	0	4	0	1
Total	1,211 of 4,694	74%	293 of 637	54%	292	590	228	1,039	50	268	6	47

Questions:

Are the coverage results from JaCoCo similar to the ones you got from IntelliJ in the last task? Why so or why not?

JaCoCo and IntelliJ are completely different. For example, the level in JaCoCo is 67% coverage but in IntelliJ it's 15% coverage. JaCoCo is more specific on each element.

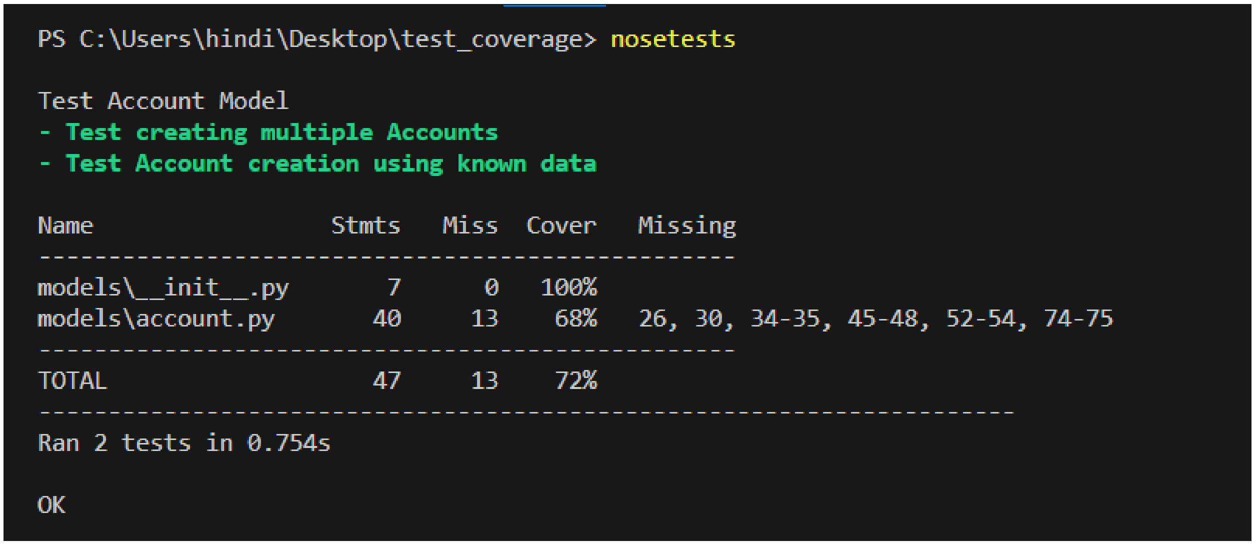
Did you find helpful the source code visualization from JaCoCo on uncovered branches?

Yes, the colors help a lot to see what missing coverage is. When we see percentage, it doesn't feel as great of an impact.

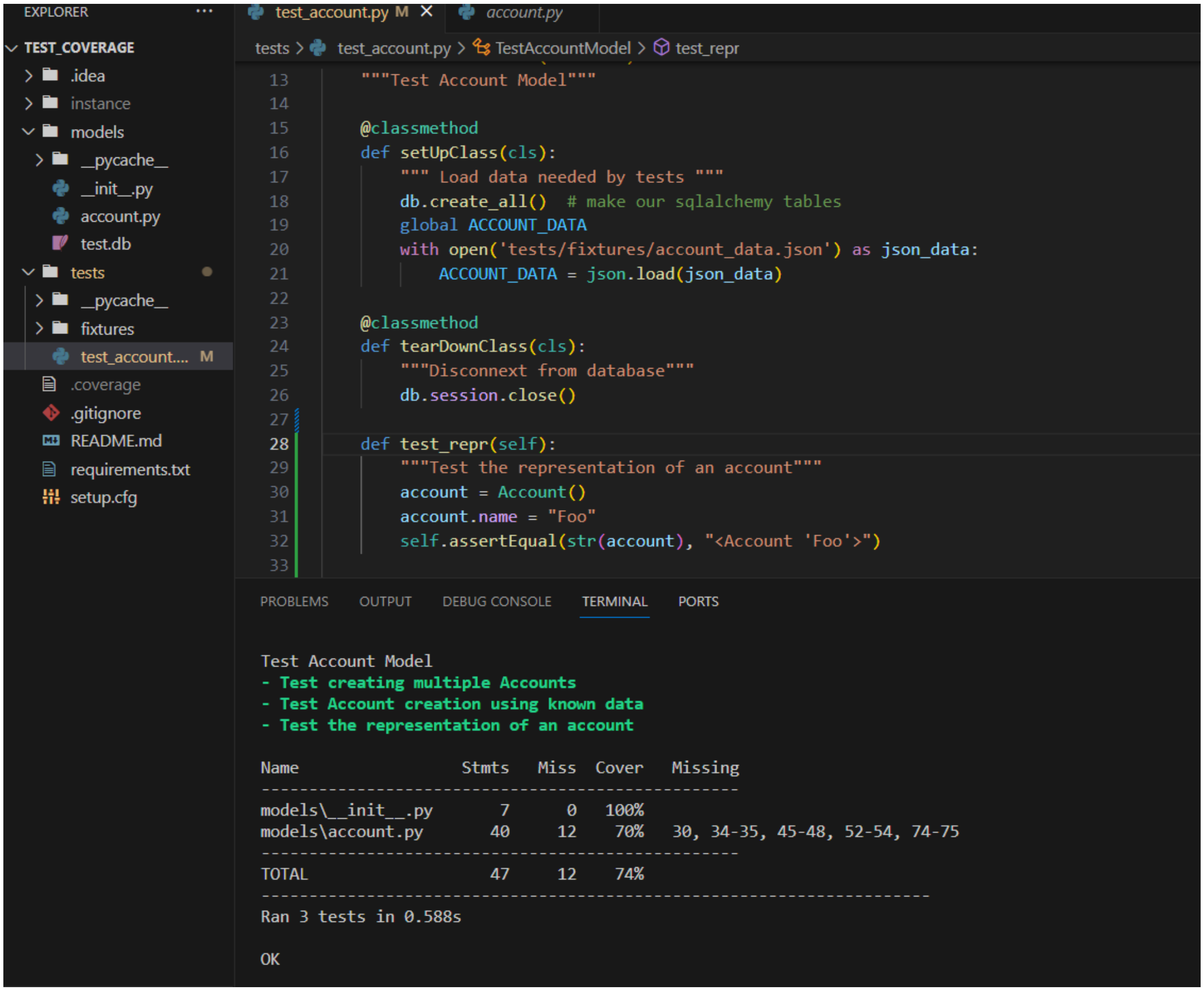
Which visualization did you prefer and why? IntelliJ's coverage window or JaCoCo's report?

It is hard to say. JaCoCo has nice colors, but IntelliJ has a cleaner look. If JaCoCo had a better cleaner look, then it would topple IntelliJ.

Task 4:

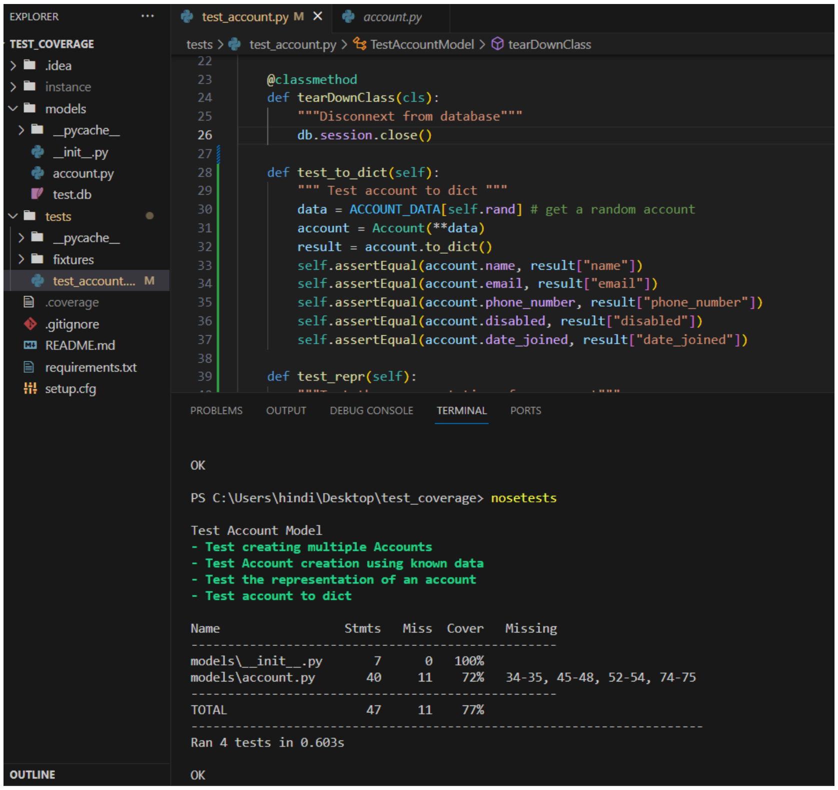


```
PS C:\Users\hindi\Desktop\test_coverage> nosetests

Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data

Name                    Stmts   Miss  Cover   Missing
-----------------------------------------------------
models\__init__.py          7      0   100%
models\account.py          40     13    68%   26, 30, 34-35, 45-48, 52-54, 74-75
-----------------------------------------------------
TOTAL                      47     13    72%
----------------------------------------------------------------------
Ran 2 tests in 0.754s

OK
```



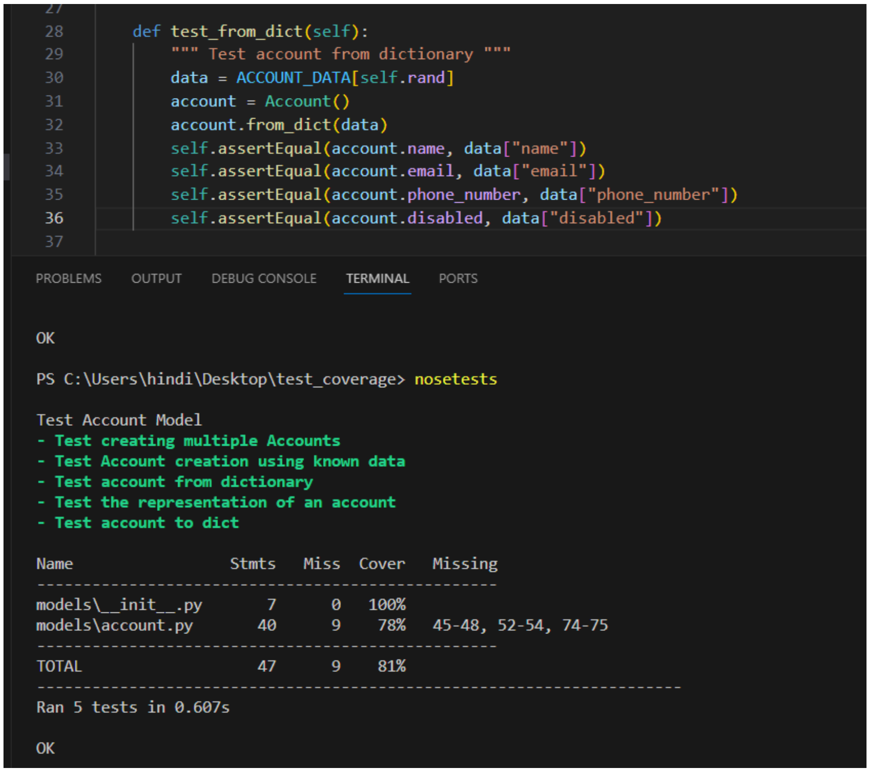
```python
    """Test Account Model"""

    @classmethod
    def setUpClass(cls):
        """ Load data needed by tests """
        db.create_all()  # make our sqlalchemy tables
        global ACCOUNT_DATA
        with open('tests/fixtures/account_data.json') as json_data:
            ACCOUNT_DATA = json.load(json_data)

    @classmethod
    def tearDownClass(cls):
        """Disconnext from database"""
        db.session.close()

    def test_repr(self):
        """Test the representation of an account"""
        account = Account()
        account.name = "Foo"
        self.assertEqual(str(account), "<Account 'Foo'>")
```

```
Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data
- Test the representation of an account

Name                    Stmts   Miss  Cover   Missing
-----------------------------------------------------
models\__init__.py          7      0   100%
models\account.py          40     12    70%   30, 34-35, 45-48, 52-54, 74-75
-----------------------------------------------------
TOTAL                      47     12    74%
----------------------------------------------------------------------
Ran 3 tests in 0.588s

OK
```



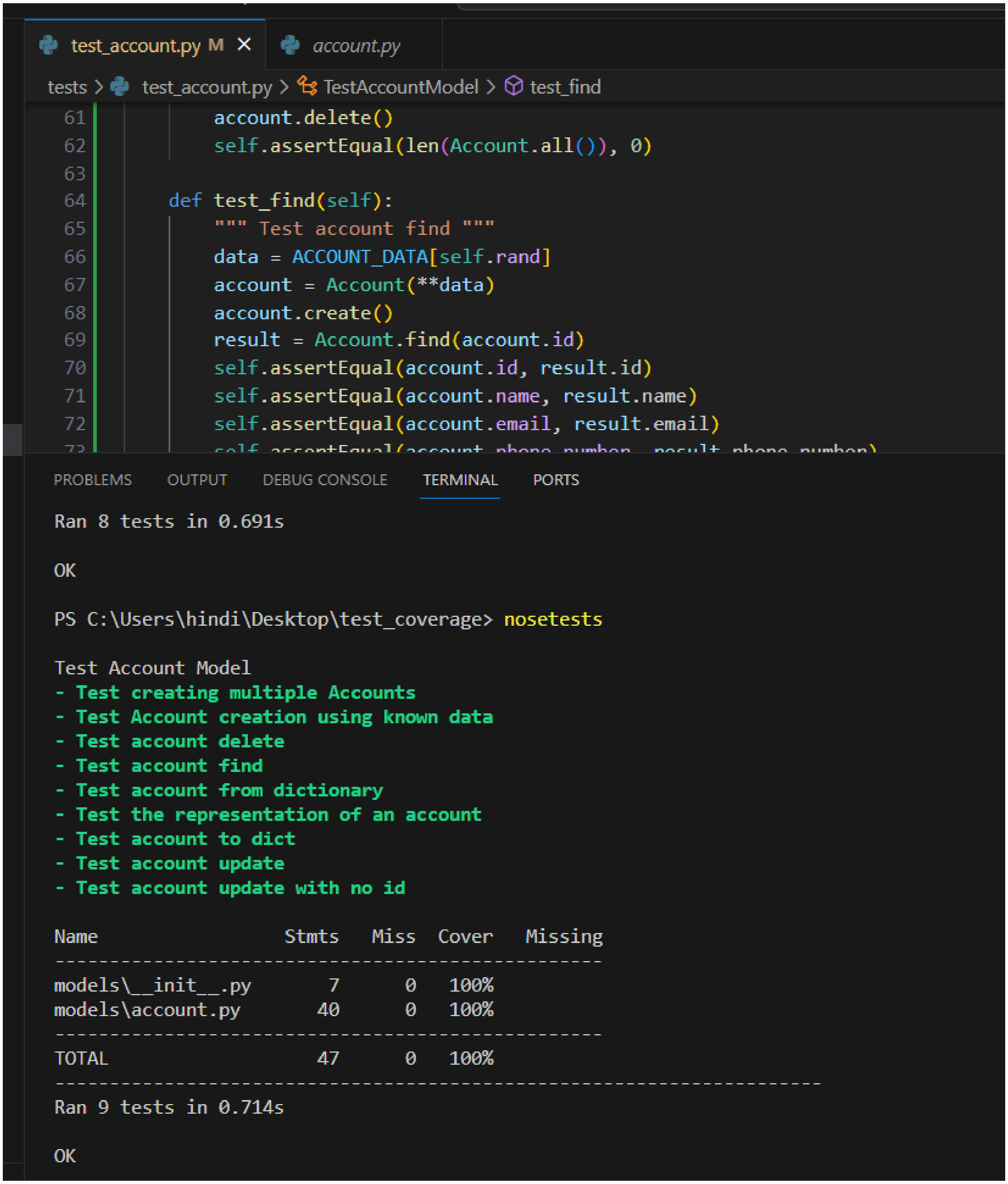
```python

    @classmethod
    def tearDownClass(cls):
        """Disconnext from database"""
        db.session.close()

    def test_to_dict(self):
        """ Test account to dict """
        data = ACCOUNT_DATA[self.rand] # get a random account
        account = Account(**data)
        result = account.to_dict()
        self.assertEqual(account.name, result["name"])
        self.assertEqual(account.email, result["email"])
        self.assertEqual(account.phone_number, result["phone_number"])
        self.assertEqual(account.disabled, result["disabled"])
        self.assertEqual(account.date_joined, result["date_joined"])

    def test_repr(self):
```

```
OK

PS C:\Users\hindi\Desktop\test_coverage> nosetests

Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data
- Test the representation of an account
- Test account to dict

Name                   Stmts   Miss  Cover   Missing
----------------------------------------------------
models\__init__.py         7      0   100%
models\account.py         40     11    72%   34-35, 45-48, 52-54, 74-75
----------------------------------------------------
TOTAL                     47     11    77%
----------------------------------------------------------------------
Ran 4 tests in 0.603s

OK
```



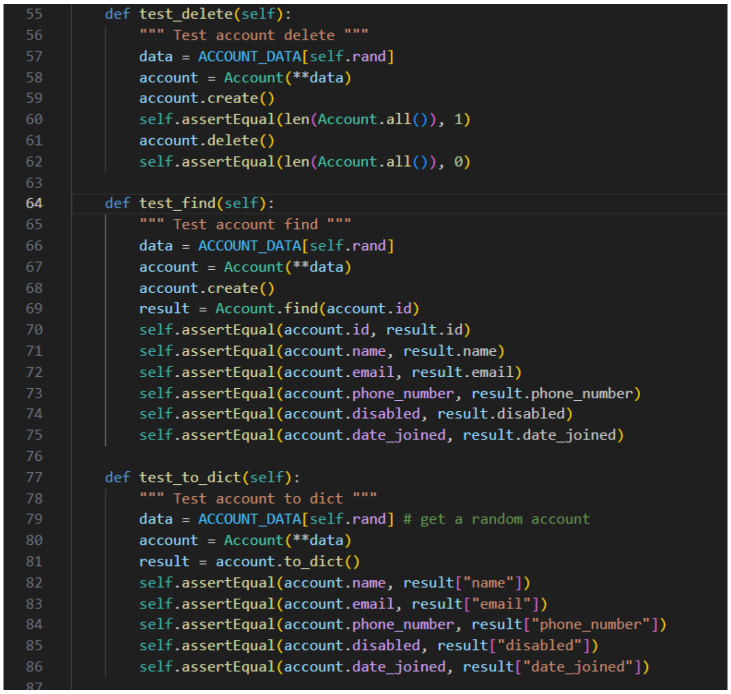
```python
    def test_from_dict(self):
        """ Test account from dictionary """
        data = ACCOUNT_DATA[self.rand]
        account = Account()
        account.from_dict(data)
        self.assertEqual(account.name, data["name"])
        self.assertEqual(account.email, data["email"])
        self.assertEqual(account.phone_number, data["phone_number"])
        self.assertEqual(account.disabled, data["disabled"])

```

```
OK

PS C:\Users\hindi\Desktop\test_coverage> nosetests

Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data
- Test account from dictionary
- Test the representation of an account
- Test account to dict

Name                   Stmts   Miss  Cover   Missing
----------------------------------------------------
models\__init__.py         7      0   100%
models\account.py         40      9    78%   45-48, 52-54, 74-75
----------------------------------------------------
TOTAL                     47      9    81%
----------------------------------------------------------------------
Ran 5 tests in 0.607s

OK
```

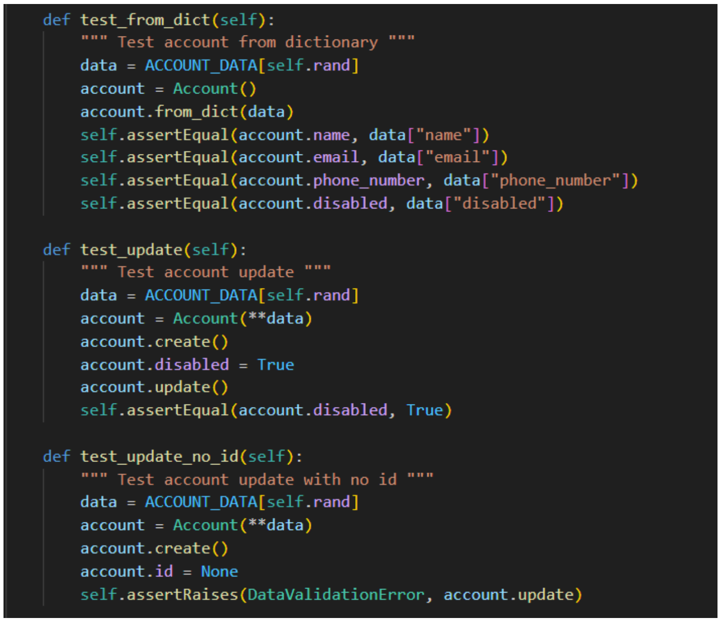


```python
        account.delete()
        self.assertEqual(len(Account.all()), 0)

    def test_find(self):
        """ Test account find """
        data = ACCOUNT_DATA[self.rand]
        account = Account(**data)
        account.create()
        result = Account.find(account.id)
        self.assertEqual(account.id, result.id)
        self.assertEqual(account.name, result.name)
        self.assertEqual(account.email, result.email)
        self.assertEqual(account.phone_number, result.phone_number)
```

```
Ran 8 tests in 0.691s

OK

PS C:\Users\hindi\Desktop\test_coverage> nosetests

Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data
- Test account delete
- Test account find
- Test account from dictionary
- Test the representation of an account
- Test account to dict
- Test account update
- Test account update with no id

Name                     Stmts   Miss  Cover   Missing
------------------------------------------------------
models\__init__.py           7      0   100%
models\account.py           40      0   100%
------------------------------------------------------
TOTAL                       47      0   100%
----------------------------------------------------------------------
Ran 9 tests in 0.714s

OK
```



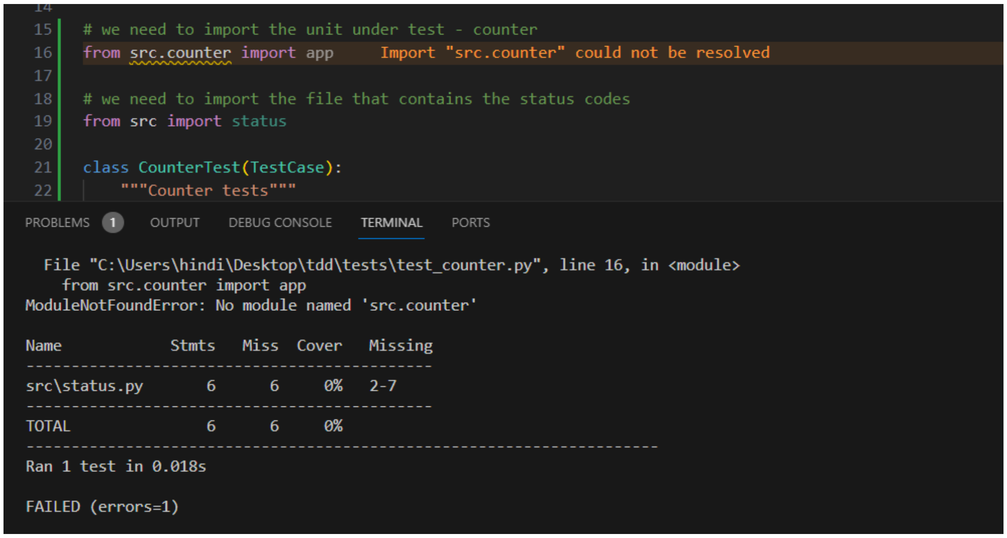
```python
    def test_delete(self):
        """ Test account delete """
        data = ACCOUNT_DATA[self.rand]
        account = Account(**data)
        account.create()
        self.assertEqual(len(Account.all()), 1)
        account.delete()
        self.assertEqual(len(Account.all()), 0)

    def test_find(self):
        """ Test account find """
        data = ACCOUNT_DATA[self.rand]
        account = Account(**data)
        account.create()
        result = Account.find(account.id)
        self.assertEqual(account.id, result.id)
        self.assertEqual(account.name, result.name)
        self.assertEqual(account.email, result.email)
        self.assertEqual(account.phone_number, result.phone_number)
        self.assertEqual(account.disabled, result.disabled)
        self.assertEqual(account.date_joined, result.date_joined)

    def test_to_dict(self):
        """ Test account to dict """
        data = ACCOUNT_DATA[self.rand] # get a random account
        account = Account(**data)
        result = account.to_dict()
        self.assertEqual(account.name, result["name"])
        self.assertEqual(account.email, result["email"])
        self.assertEqual(account.phone_number, result["phone_number"])
        self.assertEqual(account.disabled, result["disabled"])
        self.assertEqual(account.date_joined, result["date_joined"])
```



```python
    def test_from_dict(self):
        """ Test account from dictionary """
        data = ACCOUNT_DATA[self.rand]
        account = Account()
        account.from_dict(data)
        self.assertEqual(account.name, data["name"])
        self.assertEqual(account.email, data["email"])
        self.assertEqual(account.phone_number, data["phone_number"])
        self.assertEqual(account.disabled, data["disabled"])

    def test_update(self):
        """ Test account update """
        data = ACCOUNT_DATA[self.rand]
        account = Account(**data)
        account.create()
        account.disabled = True
        account.update()
        self.assertEqual(account.disabled, True)

    def test_update_no_id(self):
        """ Test account update with no id """
        data = ACCOUNT_DATA[self.rand]
        account = Account(**data)
        account.create()
        account.id = None
        self.assertRaises(DataValidationError, account.update)
```

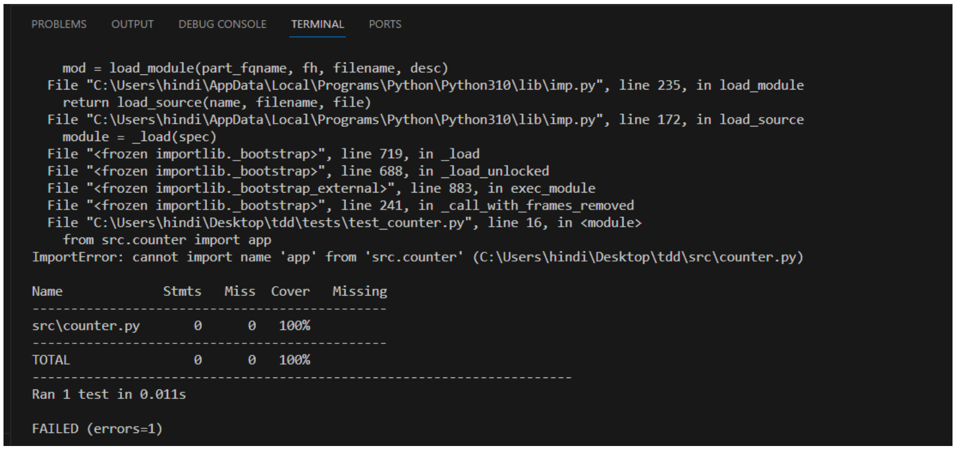
https://github.com/Intensifiesx/test_coverage

Task 5:

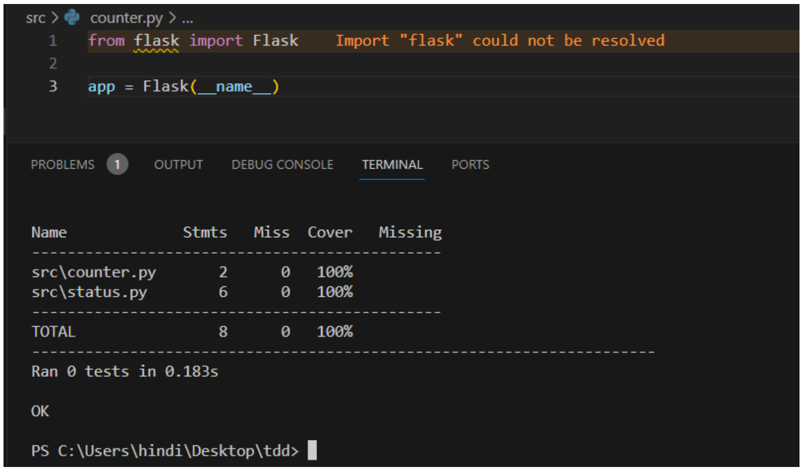


```
# we need to import the unit under test - counter
from src.counter import app      Import "src.counter" could not be resolved

# we need to import the file that contains the status codes
from src import status

class CounterTest(TestCase):
    """Counter tests"""
```

```
  File "C:\Users\hindi\Desktop\tdd\tests\test_counter.py", line 16, in <module>
    from src.counter import app
ModuleNotFoundError: No module named 'src.counter'

Name            Stmts   Miss  Cover   Missing
---------------------------------------------
src\status.py       6      6     0%   2-7
---------------------------------------------
TOTAL               6      6     0%
----------------------------------------------------------------------
Ran 1 test in 0.018s

FAILED (errors=1)
```

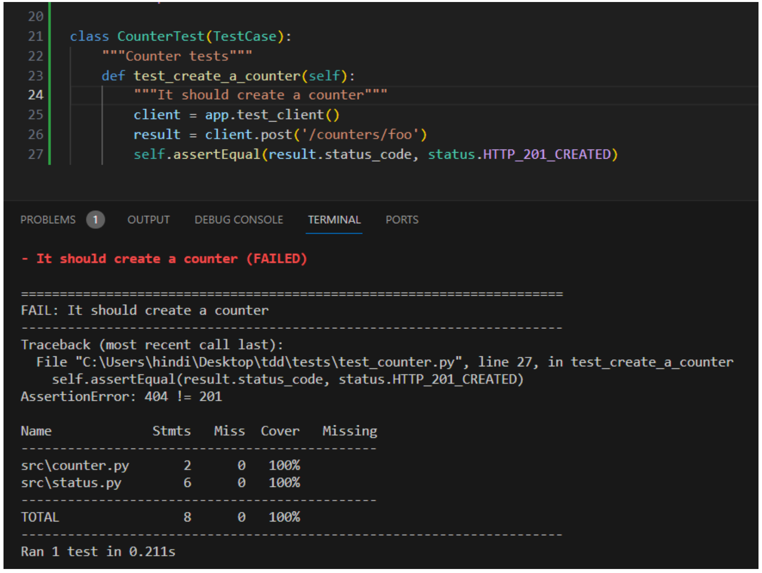


```
    mod = load_module(part_fqname, fh, filename, desc)
  File "C:\Users\hindi\AppData\Local\Programs\Python\Python310\lib\imp.py", line 235, in load_module
    return load_source(name, filename, file)
  File "C:\Users\hindi\AppData\Local\Programs\Python\Python310\lib\imp.py", line 172, in load_source
    module = _load(spec)
  File "<frozen importlib._bootstrap>", line 719, in _load
  File "<frozen importlib._bootstrap>", line 688, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 883, in exec_module
  File "<frozen importlib._bootstrap>", line 241, in _call_with_frames_removed
  File "C:\Users\hindi\Desktop\tdd\tests\test_counter.py", line 16, in <module>
    from src.counter import app
ImportError: cannot import name 'app' from 'src.counter' (C:\Users\hindi\Desktop\tdd\src\counter.py)

Name             Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py       0      0   100%
----------------------------------------------
TOTAL                0      0   100%
----------------------------------------------------------------------
Ran 1 test in 0.011s

FAILED (errors=1)
```



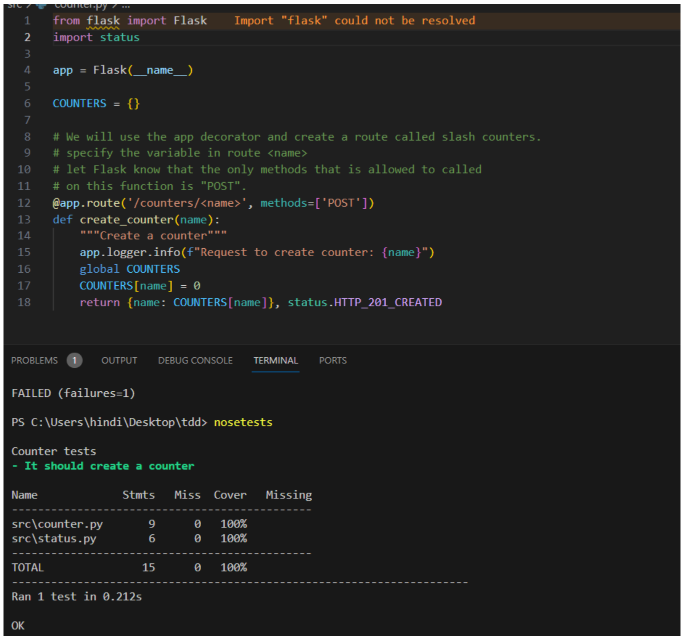
```
from flask import Flask    Import "flask" could not be resolved

app = Flask(__name__)
```

```
Name              Stmts   Miss  Cover   Missing
-----------------------------------------------
src\counter.py        2      0   100%
src\status.py         6      0   100%
-----------------------------------------------
TOTAL                 8      0   100%
----------------------------------------------------------------------
Ran 0 tests in 0.183s

OK

PS C:\Users\hindi\Desktop\tdd>
```



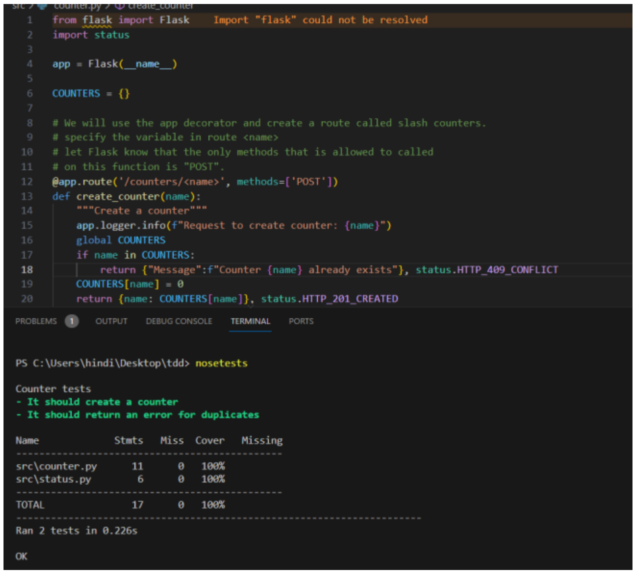
```

class CounterTest(TestCase):
    """Counter tests"""
    def test_create_a_counter(self):
        """It should create a counter"""
        client = app.test_client()
        result = client.post('/counters/foo')
        self.assertEqual(result.status_code, status.HTTP_201_CREATED)
```

```
- It should create a counter (FAILED)

======================================================================
FAIL: It should create a counter
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\hindi\Desktop\tdd\tests\test_counter.py", line 27, in test_create_a_counter
    self.assertEqual(result.status_code, status.HTTP_201_CREATED)
AssertionError: 404 != 201

Name              Stmts   Miss  Cover   Missing
-----------------------------------------------
src\counter.py        2      0   100%
src\status.py         6      0   100%
-----------------------------------------------
TOTAL                 8      0   100%
----------------------------------------------------------------------
Ran 1 test in 0.211s
```



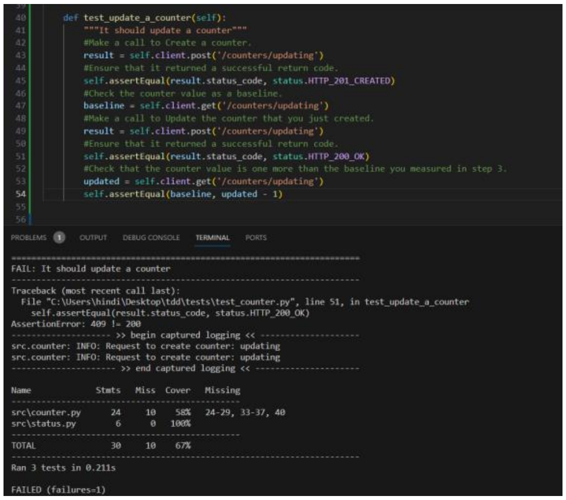
```
from flask import Flask    Import "flask" could not be resolved
import status

app = Flask(__name__)

COUNTERS = {}

# We will use the app decorator and create a route called slash counters.
# specify the variable in route <name>
# let Flask know that the only methods that is allowed to called
# on this function is "POST".
@app.route('/counters/<name>', methods=['POST'])
def create_counter(name):
    """Create a counter"""
    app.logger.info(f"Request to create counter: {name}")
    global COUNTERS
    COUNTERS[name] = 0
    return {name: COUNTERS[name]}, status.HTTP_201_CREATED
```

```
FAILED (failures=1)

PS C:\Users\hindi\Desktop\tdd> nosetests

Counter tests
- It should create a counter

Name              Stmts   Miss  Cover   Missing
-----------------------------------------------
src\counter.py        9      0   100%
src\status.py         6      0   100%
-----------------------------------------------
TOTAL                15      0   100%
----------------------------------------------------------------------
Ran 1 test in 0.212s

OK
```



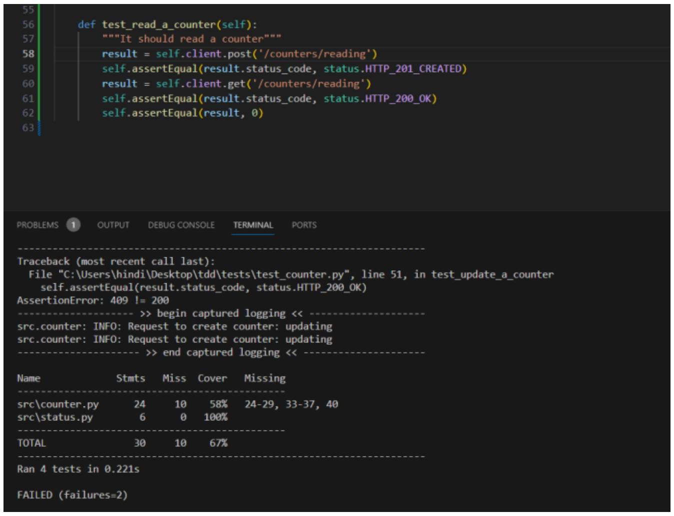
```python
from flask import Flask    Import "flask" could not be resolved
import status

app = Flask(__name__)

COUNTERS = {}

# We will use the app decorator and create a route called slash counters.
# specify the variable in route <name>
# let Flask know that the only methods that is allowed to called
# on this function is "POST".
@app.route('/counters/<name>', methods=['POST'])
def create_counter(name):
    """Create a counter"""
    app.logger.info(f"Request to create counter: {name}")
    global COUNTERS
    if name in COUNTERS:
        return {"Message":f"Counter {name} already exists"}, status.HTTP_409_CONFLICT
    COUNTERS[name] = 0
    return {name: COUNTERS[name]}, status.HTTP_201_CREATED
```

```
PS C:\Users\hindi\Desktop\tdd> nosetests

Counter tests
- It should create a counter
- It should return an error for duplicates

Name             Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py      11      0   100%
src\status.py        6      0   100%
----------------------------------------------
TOTAL               17      0   100%
----------------------------------------------------------------------
Ran 2 tests in 0.226s

OK
```



```python
    def test_update_a_counter(self):
        """It should update a counter"""
        #Make a call to Create a counter.
        result = self.client.post('/counters/updating')
        #Ensure that it returned a successful return code.
        self.assertEqual(result.status_code, status.HTTP_201_CREATED)
        #Check the counter value as a baseline.
        baseline = self.client.get('/counters/updating')
        #Make a call to Update the counter that you just created.
        result = self.client.post('/counters/updating')
        #Ensure that it returned a successful return code.
        self.assertEqual(result.status_code, status.HTTP_200_OK)
        #Check that the counter value is one more than the baseline you measured in step 3.
        updated = self.client.get('/counters/updating')
        self.assertEqual(baseline, updated - 1)
```

```
======================================================================
FAIL: It should update a counter
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\hindi\Desktop\tdd\tests\test_counter.py", line 51, in test_update_a_counter
    self.assertEqual(result.status_code, status.HTTP_200_OK)
AssertionError: 409 != 200
-------------------- >> begin captured logging << --------------------
src.counter: INFO: Request to create counter: updating
src.counter: INFO: Request to create counter: updating
--------------------- >> end captured logging << ---------------------

Name             Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py      24     10    58%   24-29, 33-37, 40
src\status.py        6      0   100%
----------------------------------------------
TOTAL               30     10    67%
----------------------------------------------------------------------
Ran 3 tests in 0.211s

FAILED (failures=1)
```



```python
    def test_read_a_counter(self):
        """It should read a counter"""
        result = self.client.post('/counters/reading')
        self.assertEqual(result.status_code, status.HTTP_201_CREATED)
        result = self.client.get('/counters/reading')
        self.assertEqual(result.status_code, status.HTTP_200_OK)
        self.assertEqual(result, 0)
```

```
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\hindi\Desktop\tdd\tests\test_counter.py", line 51, in test_update_a_counter
    self.assertEqual(result.status_code, status.HTTP_200_OK)
AssertionError: 409 != 200
-------------------- >> begin captured logging << --------------------
src.counter: INFO: Request to create counter: updating
src.counter: INFO: Request to create counter: updating
--------------------- >> end captured logging << ---------------------

Name             Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py      24     10    58%   24-29, 33-37, 40
src\status.py        6      0   100%
----------------------------------------------
TOTAL               30     10    67%
----------------------------------------------------------------------
Ran 4 tests in 0.221s

FAILED (failures=2)
```

I give up