**بسم الله الرحمن الرحیم**

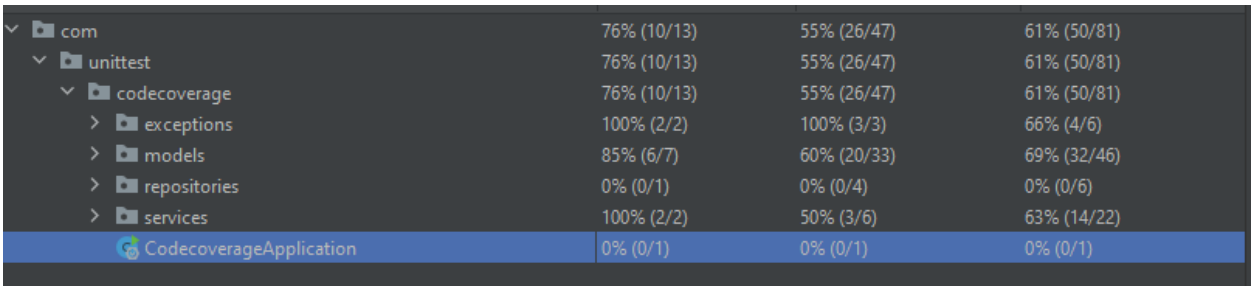
**محمدرضا یوسف پور** 97106324

**سید مهدی فقیه** 97106198

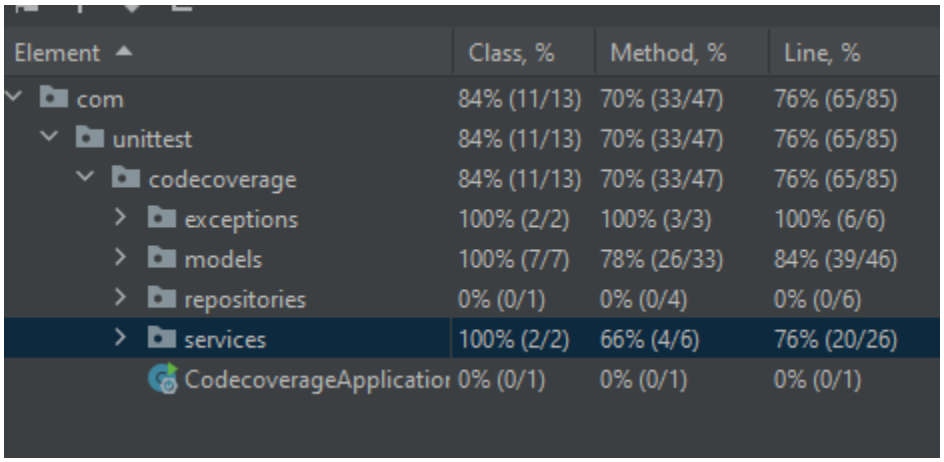
**درصد مشارکت هر دو عضو تیم در انجام آزمایش یکسان بوده است.**

**ادرس گیتهاب :** https://github.com/SMahdiFaghih/SoftwareEngineeringLab-E6

**در ابتدا تصویر زیر نمایانگر میزان** code coverage **در پروژه بوده است.**




**بعد از اعمال تغییراتی در کلاس** trafficBehaviorServiceTest **و اضافه کردن تابع** testTraffic **و افزودن یک حالت به** footpassengerCrossTheStreet **در کلاس** TrafficBehaviorServiceImpl **به نتیجه ای که در تصویر زیر مشاهده می کنید رسیدیم.**



**تابع اضافه شده به صورت زیر است.**

```
@Test
public void testTraffic()
{
    Traffic currentTraffic = new Traffic();
    currentTraffic.setIntenseCarTraffic(false);
    currentTraffic.setMinSpeedAllowed((short) 30);
    currentTraffic.setMaxSpeedAllowed((short) 60);
    currentTraffic.setStreetDirectionFlow(StreetDirectionFlow.TWO_WAY);

    Footpassenger currentFootPassengerBehavior = new Footpassenger();
    currentFootPassengerBehavior.setCrossedTheRoad(true);
    currentFootPassengerBehavior.setLookedToTheLeft(false);
    currentFootPassengerBehavior.setLookedToTheRight(false);
    currentFootPassengerBehavior.setCrossedTrafficLigth(TrafficLigth.GREEN);

    Assertions.assertThatThrownBy(() -> trafficBehaviorService.footpassengerCrossTheStreet(currentTraffic, currentFootPassengerBehavior))
            .isInstanceOf(BehaviorException.class)
            .hasMessageContaining( description: "What are you doing man?");

}
```

**و حالت جدید اضافه شده به صورت زیر است**

```
if(currentTrafic.getStreetDirectionFlow() == StreetDirectionFlow.TWO_WAY &&
        !currentFootpassengerBehavior.lookedToTheLeft() &&
        !currentFootpassengerBehavior.lookedToTheRight()) {
    throw new BehaviorException("What are you doing man?");
}
```

**سپس در کلاس** personServiceTest 3 **تابع جدید که تصاویر آن ها را مشاهده می کنید را اضافه کردیم و نتیجه ی زیر را به ارمغان آورد.**

| Element | Class, % | Method, % | Line, % |
|---|---|---|---|
| com | 84% (11/13) | 74% (35/47) | 80% (68/85) |
| unittest | 84% (11/13) | 74% (35/47) | 80% (68/85) |
| codecoverage | 84% (11/13) | 74% (35/47) | 80% (68/85) |
| exceptions | 100% (2/2) | 100% (3/3) | 100% (6/6) |
| models | 100% (7/7) | 78% (26/33) | 84% (39/46) |
| repositories | 0% (0/1) | 0% (0/4) | 0% (0/6) |
| services | 100% (2/2) | 100% (6/6) | 88% (23/26) |
| CodecoverageApplication | 0% (0/1) | 0% (0/1) | 0% (0/1) |

**توابع اضافه شده را در تصاویر زیر مشاهده می کنید. توابع اضافه شده در تصاویر زیر برای متد های** delete و update و get **کلاس** personServiceImpl **می باشد.**

```
@Test
public void testPerson() {

    List<String> expectedErrors = Lists.newArrayList( ...elements: "Name is required");
    String expectedMessage = String.join( delimiter: ";", expectedErrors);


    assertThatThrownBy(() -> service.delete( id: "")) AbstractThrowableAssert<capture of ?, capture of ? extends Throwable>
            .isInstanceOf(PersonException.class) capture of ?
            .hasFieldOrPropertyWithValue( name: "errors", expectedErrors)
            .hasMessage(expectedMessage);
}
```

```
@Test
public void UpdatePerson() {
    List<String> expectedErrors = Lists.newArrayList( ...elements: "Gender is required");
    String expectedMessage = String.join( delimiter: ";", expectedErrors);

    Person person = new Person();
    person.setName("Kambiz");
    person.setAge(23);
    person.setGender(Gender.M);

    person.setGender(null);

    assertThatThrownBy(() -> service.update(person)) AbstractThrowableAssert<capture of ?, capture of ? extends Throwable>
            .isInstanceOf(PersonException.class) capture of ?
            .hasFieldOrPropertyWithValue( name: "errors", expectedErrors)
            .hasMessage(expectedMessage);

}
```

```
@Test
public void TestGetPerson() {

    List<String> expectedErrors = Lists.newArrayList( ...elements: "Name is required");
    String expectedMessage = String.join( delimiter: ";", expectedErrors);


    assertThatThrownBy(() -> service.get("")) AbstractThrowableAssert<capture of ?, capture of ? extends Throwable>
            .isInstanceOf(PersonException.class) capture of ?
            .hasFieldOrPropertyWithValue( name: "errors", expectedErrors)
            .hasMessage(expectedMessage);
}
```

**کارهای انجام شده در تصاویر بالا باعث بهبود** code coverage **در کلاس های پکیج های** services **و** models **شده است**