

| | | |
|---|---|--|
| An-Najah National University Faculty of Engineering and IT Electrical and Computer Engineering Department |  | جامعة النجاح الوطنية كلية الهندسة وتكنولوجيا المعلومات دائرة الهندسة الكهربائية والحاسوب |
|---|---|--|

Software Engineering Project Report

| | |
|-----------------|---|
| Students | Najat Mansour (12028099) Mohammad Alawneh (12028067) |
| Instructor Name | Dr. Haya Samaana |
| Performed In | 02.08.2023 |

Content

- GitHub Link
- Run all the testcases
- Code Coverage from IntelliJ Ultimate
- CI/CD using GitHub actions, Sonar Cloud and Jenkins
- Hot topics in our project 😊!
- Models UML

GitHub Link

<https://github.com/SakancomOrganization/Sakancom>

| | |
|--------------|----------------------|
| Organization | SakancomOrganization |
| Owner | mo-alawneh |
| Repository | Sakancom |

Run all the testcases

We have:

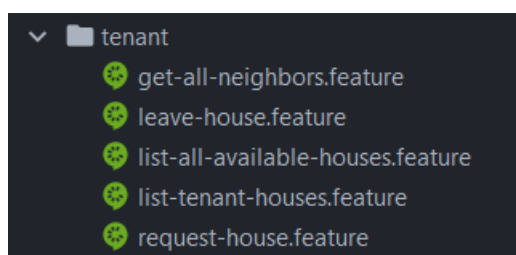
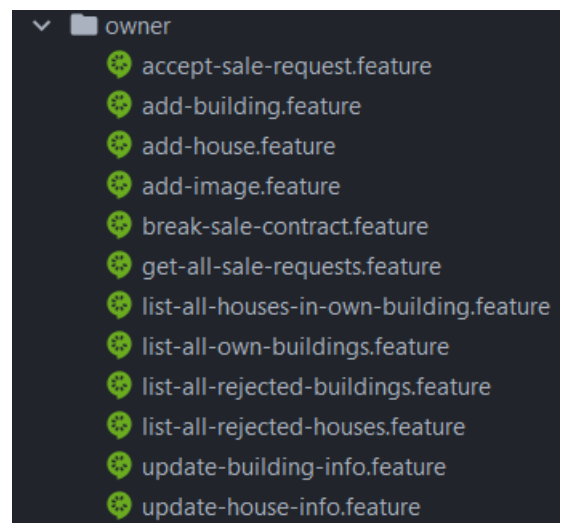
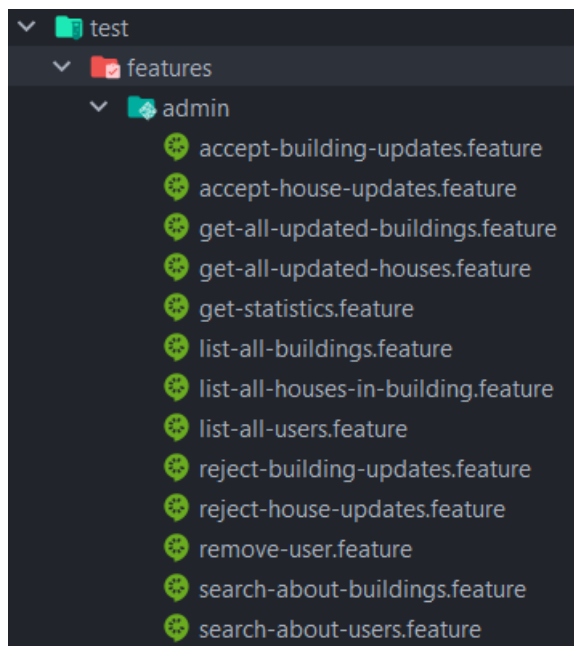
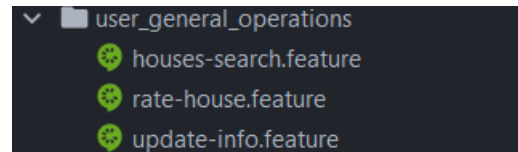
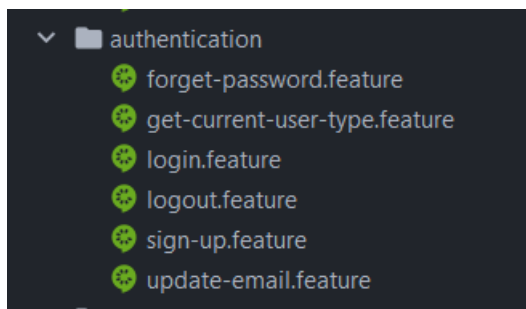
1. **273 testcase (test)**

✔ Tests passed: 273 of 273 tests – 10 sec 931 ms

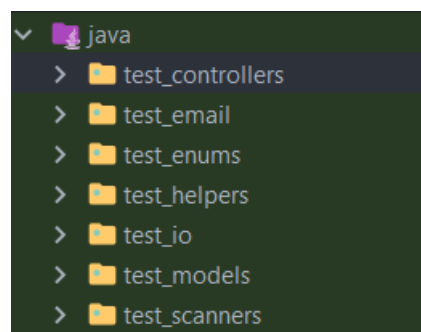
```
> ✔ TestColors
> ✔ TestCustomizedScanners
> ✔ TestFileSystemMakerWithMockito
> ✔ TestSaleContract
> ✔ TestHouse
> ✔ TestLocation
> ✔ TestHouseRate
> ✔ TestYmlHandler
> ✔ TestBuilding
> ✔ TestContactInfo
> ✔ TestEmailServiceWithMockito
> ✔ TestUser
> ✔ TestRunnerForTenant
> ✔ TestPasswordChecker
```

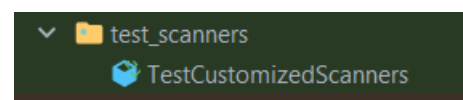
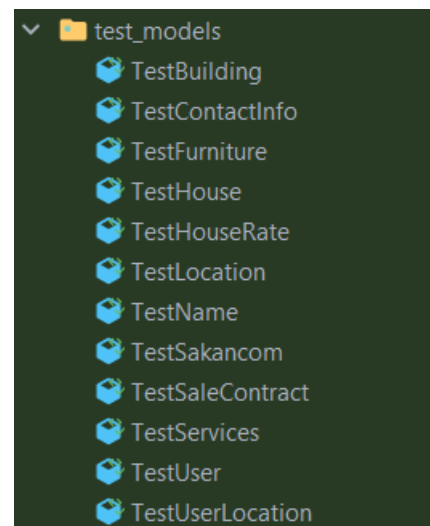
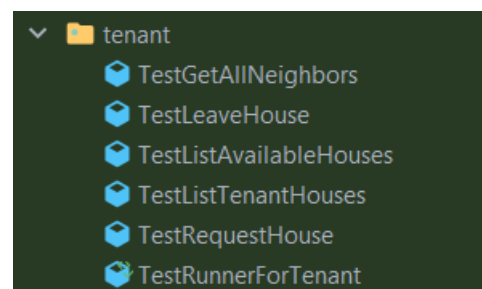
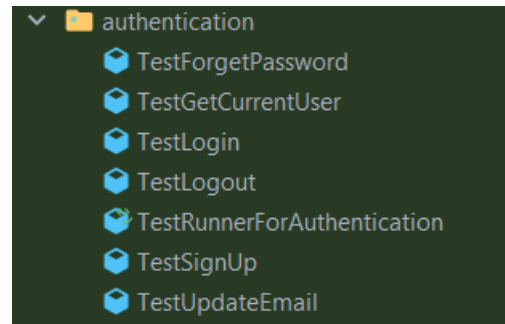
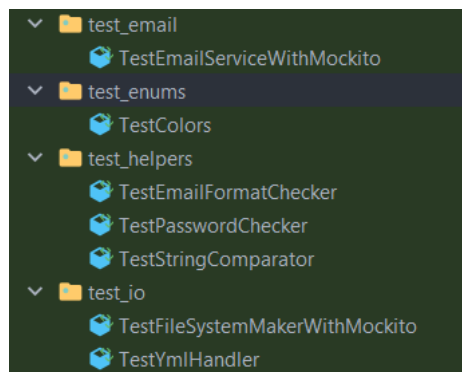
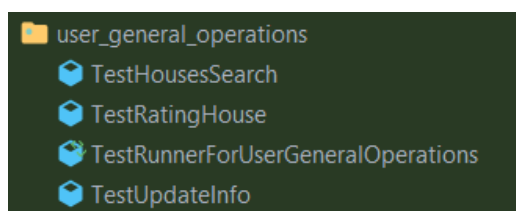
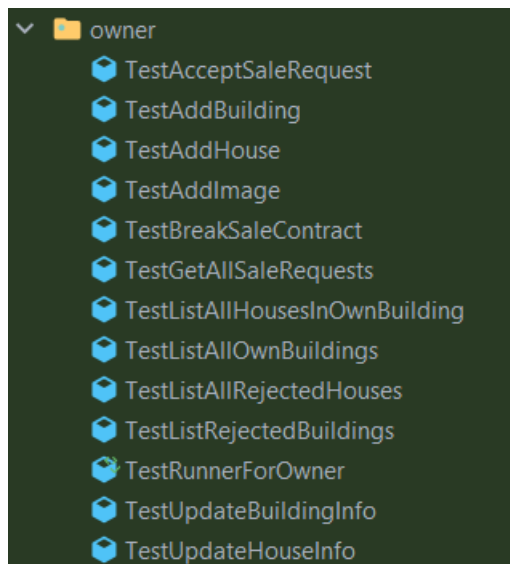
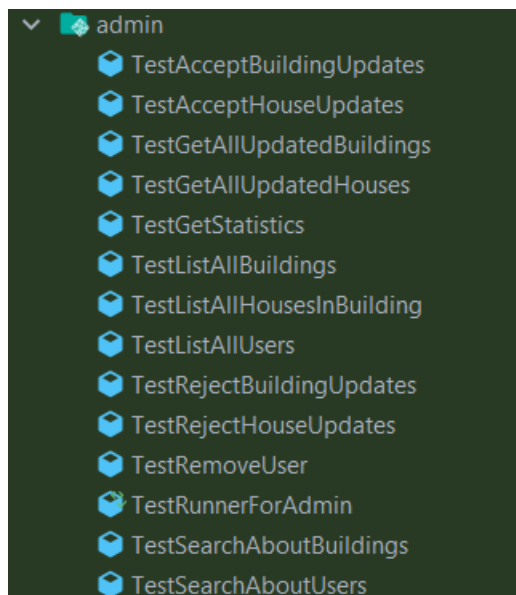
```
> ✔ TestSakancom
> ✔ TestFurniture
> ✔ TestServices
> ✔ TestEmailFormatChecker
> ✔ TestRunnerForUserGeneralOperations
> ✔ TestRunnerForOwner
> ✔ TestRunnerForAdmin
> ✔ TestRunnerForAuthentication
> ✔ TestStringComparator
> ✔ TestUserLocation
> ✔ TestName
```

2. 39 feature files



3. 64 test class

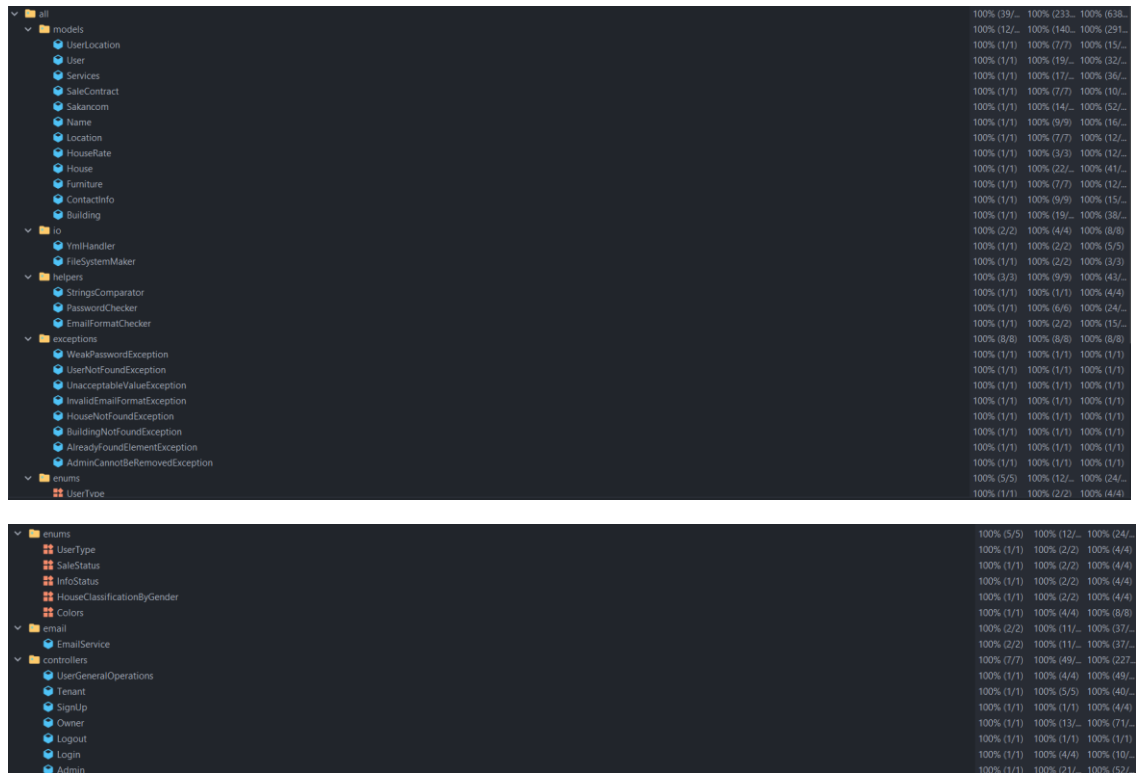






Code Coverage from IntelliJ Ultimate

We got a 100% line -statement- coverage!



Current scope: all classes

Overall Coverage Summary

| Package | Class, % | Method, % | Line, % |
|-------------|--------------|----------------|----------------|
| all classes | 100% (39/39) | 100% (234/234) | 100% (638/638) |

Coverage Breakdown

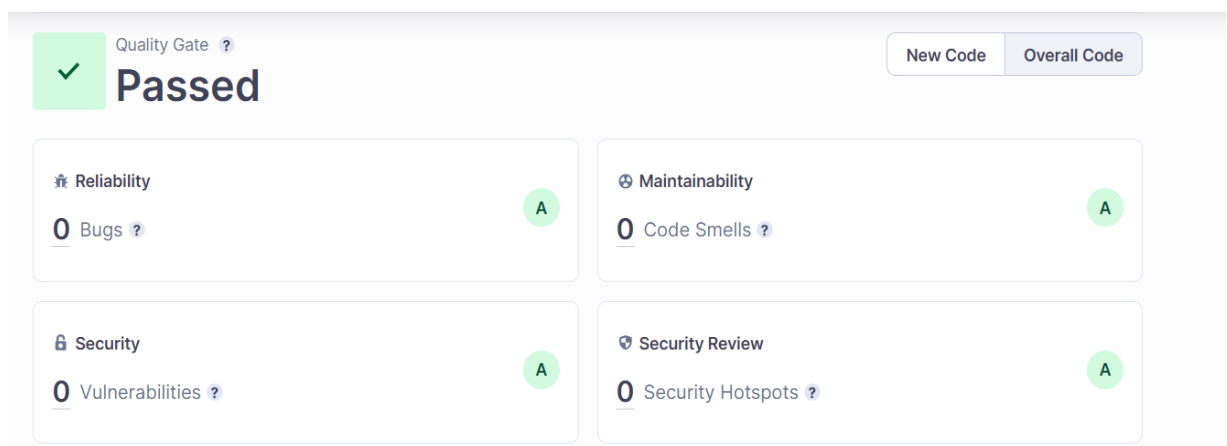
| Package | Class, % | Method, % | Line, % |
|-------------|--------------|----------------|----------------|
| controllers | 100% (7/7) | 100% (49/49) | 100% (227/227) |
| email | 100% (2/2) | 100% (11/11) | 100% (37/37) |
| enums | 100% (5/5) | 100% (12/12) | 100% (24/24) |
| exceptions | 100% (8/8) | 100% (8/8) | 100% (8/8) |
| helpers | 100% (3/3) | 100% (9/9) | 100% (43/43) |
| io | 100% (2/2) | 100% (5/5) | 100% (8/8) |
| models | 100% (12/12) | 100% (140/140) | 100% (291/291) |

generated on 2023-08-08 19:27



CI/CD using GitHub actions, Sonar Cloud and Jenkins

4. GitHub actions





NOTE: We used **SonarLint** plugin on IntelliJ since we have been starting the project, it is a plugin on IntelliJ that tracks the code smells in time, so we have 0 code smells from the first time 😊 !

5. GitHub actions

| 33 workflow runs | | Event ▼ | Status ▼ | Branch ▼ | Actor ▼ |
|------------------|--|---------|------------------------|----------|---------|
| ✓ | add main comment Build #52: Commit 7d1c8a2 pushed by mo-alawneh | main | 18 hours ago 1m 30s | ... | |
| ✓ | final version Build #51: Commit 5371262 pushed by mo-alawneh | main | 18 hours ago 1m 8s | ... | |
| ✓ | remove bad smells Build #50: Commit 03e1e21 pushed by mo-alawneh | main | 18 hours ago 1m 24s | ... | |
| ✓ | add user general operations menu Build #49: Commit 9bb3cfb pushed by mo-alawneh | main | 18 hours ago 1m 12s | ... | |
| ✓ | test scanners Build #48: Commit 383c9ff pushed by mo-alawneh | main | yesterday 1m 32s | ... | |
| ✓ | test add dirs and files Build #47: Commit 0ad9c53 pushed by mo-alawneh | main | yesterday 1m 38s | ... | |
| ✓ | fix the bug Build #46: Commit a8ce07f pushed by mo-alawneh | main | yesterday 1m 13s | ... | |

These workflow runs are done automatically by using a YAML script.

6. Jenkins

1. Static run

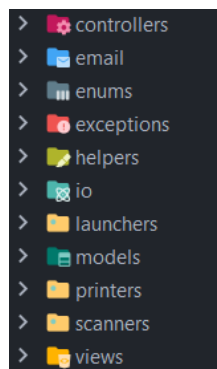


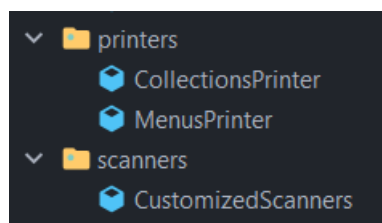
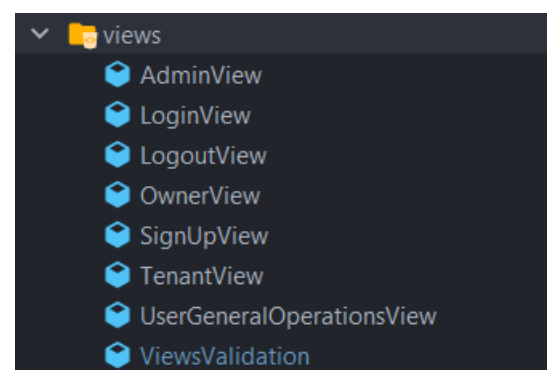
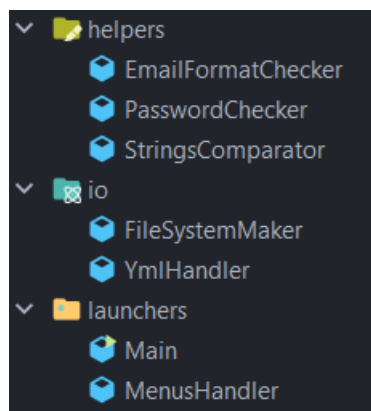
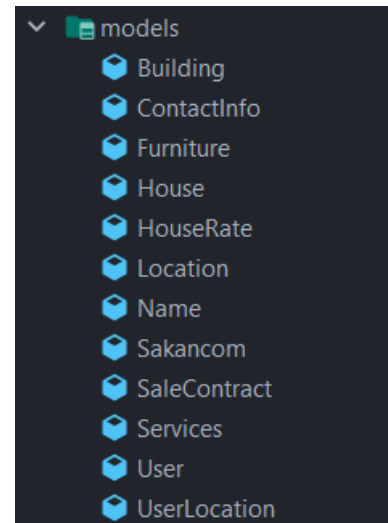
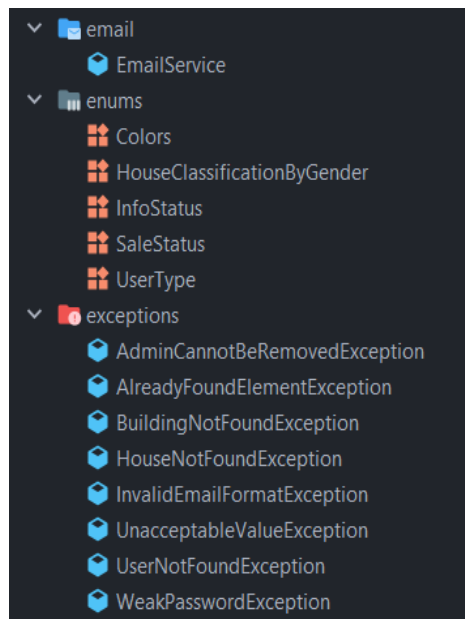
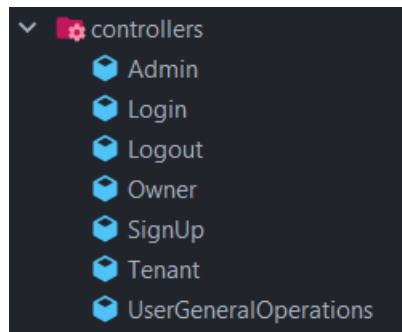
2. Run every 1 mins

| | |
|---------|--------------------------------------|
| ✓ #1265 | Aug 8, 2023, 7:42 PM |
| ✓ #1264 | Aug 8, 2023, 7:41 PM |
| ✓ #1263 | Aug 8, 2023, 7:40 PM |
| ✓ #1262 | Aug 8, 2023, 7:39 PM |
| ✓ #1261 | Aug 8, 2023, 7:38 PM |
| ✓ #1260 | Aug 8, 2023, 7:37 PM |
| ✓ #1259 | Aug 8, 2023, 7:36 PM |
| ✓ #1258 | Aug 8, 2023, 7:35 PM |
| ✓ #1257 | Aug 8, 2023, 7:34 PM |
| ✓ #1256 | Aug 8, 2023, 7:33 PM |
| ✓ #1255 | Aug 8, 2023, 7:32 PM |
| ✓ #1254 | Aug 8, 2023, 7:31 PM |
| ✓ #1253 | Aug 8, 2023, 7:30 PM |
| ✓ #1252 | Aug 8, 2023, 7:29 PM |
| ✓ #1251 | Aug 8, 2023, 7:28 PM |
| ✓ #1250 | Aug 8, 2023, 7:27 PM |
| ✓ #1249 | Aug 8, 2023, 7:26 PM |
| ✓ #1248 | Aug 8, 2023, 7:25 PM |

Hot topics in out project

- We design our project using MVC, so we have the following packages:
 1. Models. (See the models UML on the last page [Press here](#))
 2. Controllers
 3. Views
 4. helpers
 5. Email
 6. Enums
 7. Exceptions
 8. io
 9. Scanners
 10. Printers
 11. Launchers





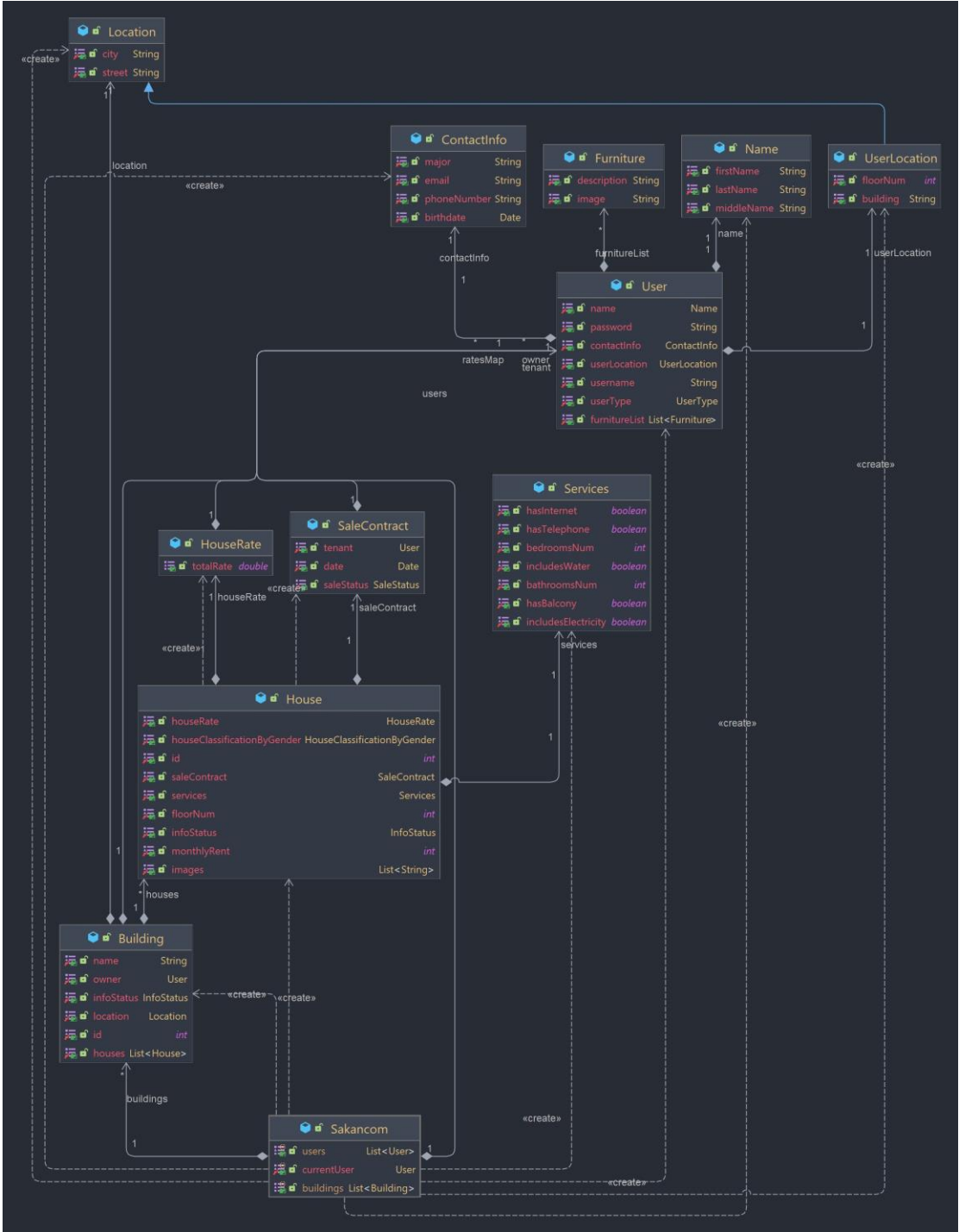
- We send an HTML email service using **JavaMailServer**.
- Put the sender email address and token in a separate **YML file** and read them using **YMLHandler** class, the file also contains a path to the images parent directory. -see the next point-

```

1 # Email Configurations
2 fromEmail: mo.a.alawneh@gmail.com
3 password: oalhpnxavixrdpq
4
5 # Images Parent Directory Path
6 path: "C:\\Users\\HITECH\\IdeaProjects\\Sakancom\\src\\main\\resources\\images\\building_"

```

- We have a dynamic generation for each added building that contains a dynamic generated directories for each added houses inside the building, and the generated houses directories contains a dynamic generated images file. We use **FilesystemMaker** class to handle these tasks.
- Why we got 100% code coverage? 😊 we use the Mockito frame work to test three things:
 - a. **Email service:** to prevent sending actual emails during the testing.
 - b. **Scanners:**
 - ✓ Allow CI/CD servers to run all the testcases.
 - ✓ Prevent the test from waiting any user input.
 - c. **Making directories and files:** CI/CD servers does not have any access for the O.S. file system, so we need a fake (mock) one.
- We made a good **OOP design** -**DON'T forget to see the UML**-, a **good** structure (several folders and packages) and try to applied the SOLID principles as can as possible on the **packages**, **classes**, **methods**, **features** and **scenarios** levels.



THE END!

THANK YOU!