



Software Engineering

10636312

# **Software Project**

**Sakancom**

*Dr. Haya Samna*

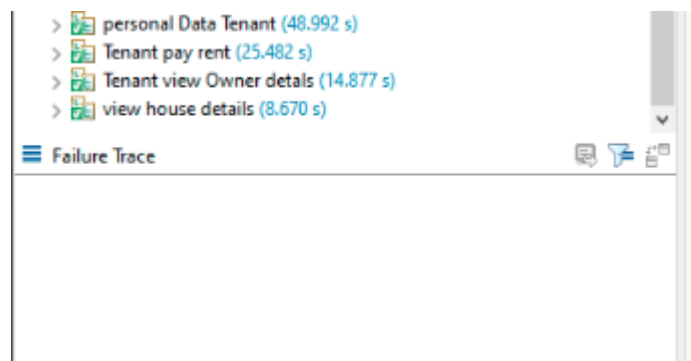
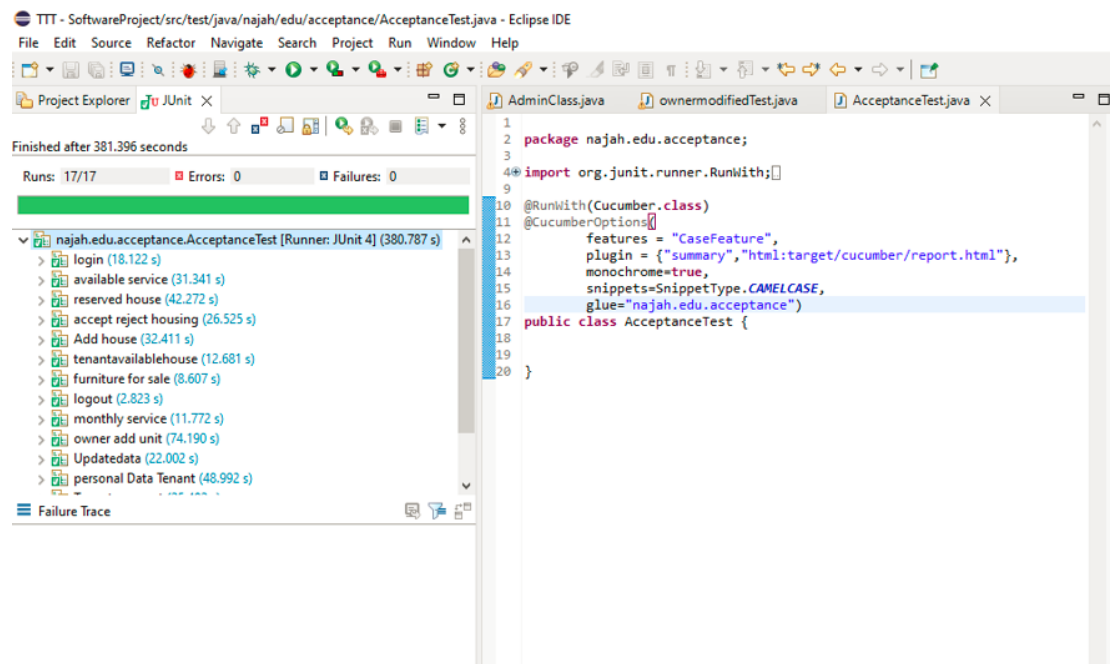
**Student :**

*Areej Sawalha 12028958*

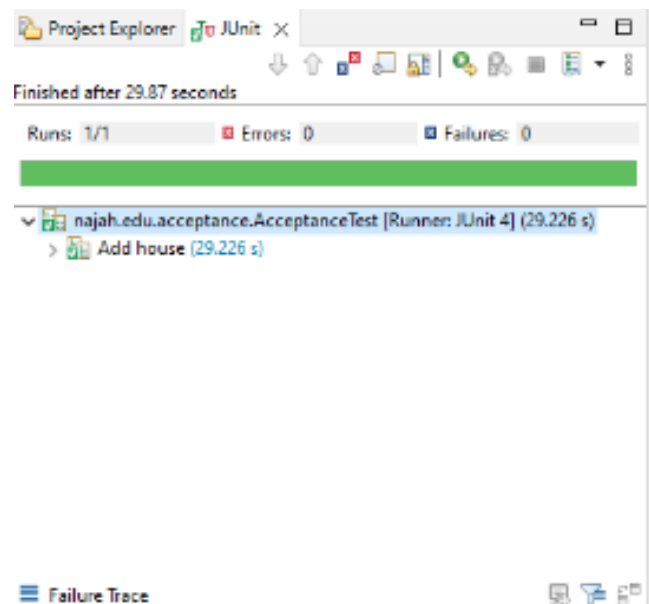
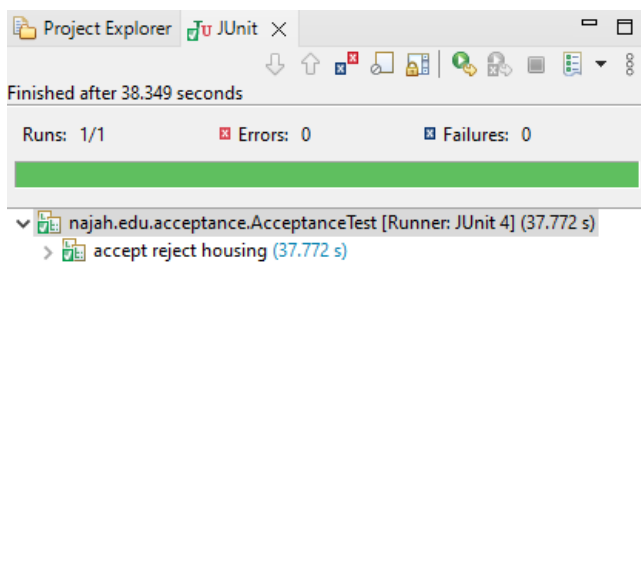
*Baraa Shelbaya 12028169*

The result:

- ▼ 📁 CaseFeature
  - 🌿 accept-reject-housing.feature
  - 🌿 addhouse.feature
  - 🌿 availablehouseTen.feature
  - 🌿 furnituresale.feature
  - 🌿 Login.feature
  - 🌿 logout.feature
  - 🌿 monthlyrent.feature
  - 🌿 owneraddunit.feature
  - 🌿 ownermodified.feature
  - 🌿 Serviceavailable.feature
  - 🌿 showpersonaldataTenant.feature
  - 🌿 tenantpayrent.feature
  - 🌿 TenantReserved.feature
  - 🌿 tenantviewownerdetels.feature
  - 🌿 viewhosuse.feature
- ▼ 📁 Tests
  - > 📄 AcceptanceTest.java
  - > 📄 acceptrejecthousing.java
  - > 📄 AdminAddhouseTest.java
  - > 📄 furnituresaleTest.java
  - > 📄 LoginTest.java
  - > 📄 logoutTest.java
  - > 📄 monthlyrentTest.java
  - > 📄 owneraddhouseTest.java
  - > 📄 ownermodifiedTest.java
  - > 📄 personaldatatenantTest.java
  - > 📄 Serviceavailable.java
  - > 📄 TenantpayrentTest.java
  - > 📄 tenantviewavailablehouse.java
  - > 📄 tenantviewownerdetelsTest.java
  - > 📄 TenentReservedTest.java
  - > 📄 viewhouseTest.java



## ***Test Feature:***



Project Explorer JUnit X

Finished after 8.187 seconds

Runs: 1/1 Errors: 0 Failures: 0

▼ najah.edu.acceptance.AcceptanceTest [Runner: JUnit 4] (7.588 s)  
    > tenantavailablehouse (7.588 s)

Failure Trace

Project Explorer JUnit X

Finished after 5.838 seconds

Runs: 1/1 Errors: 0 Failures: 0

▼ najah.edu.acceptance.AcceptanceTest [Runner: JUnit 4] (5.190 s)  
    > furniture for sale (5.190 s)

Failure Trace

Project Explorer JUnit X

Finished after 16.771 seconds

Runs: 3/3 Errors: 0 Failures: 0

▼ najah.edu.acceptance.AcceptanceTest [Runner: JUnit 4] (16.087 s)  
    > login (16.087 s)

Failure Trace

Project Explorer JUnit X

Finished after 3.102 seconds

Runs: 1/1 Errors: 0 Failures: 0

▼ najah.edu.acceptance.AcceptanceTest [Runner: JUnit 4] (2.467 s)  
    > logout (2.467 s)

Failure Trace

Project Explorer JUnit X

Finished after 6.706 seconds

Runs: 1/1 Errors: 0 Failures: 0

▼ najah.edu.acceptance.AcceptanceTest [Runner: JUnit 4] (6.056 s)  
    > monthly service (6.056 s)

Failure Trace

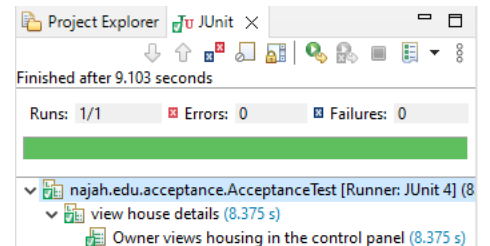
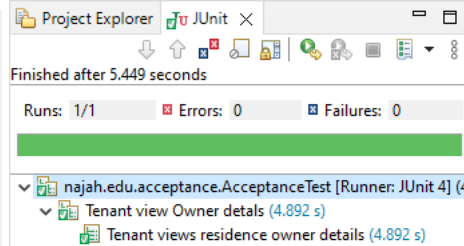
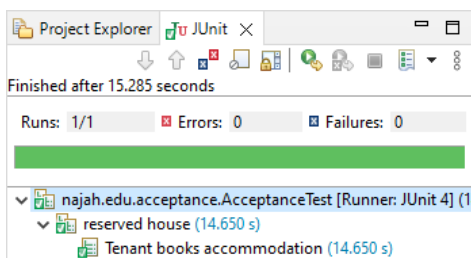
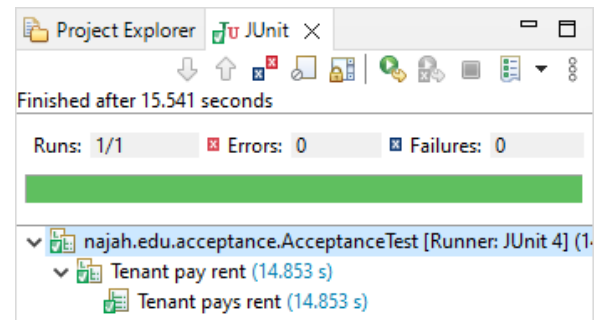
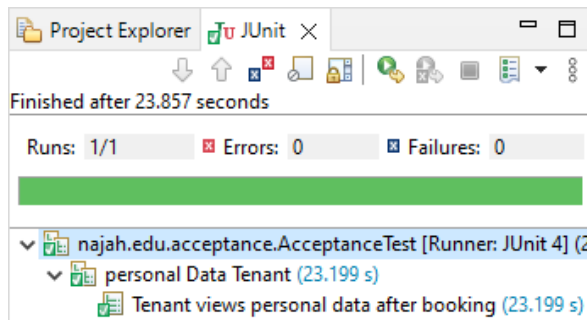
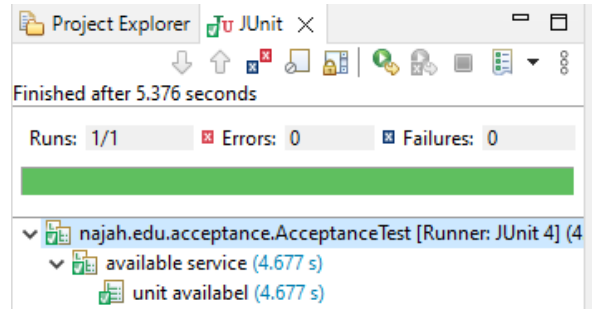
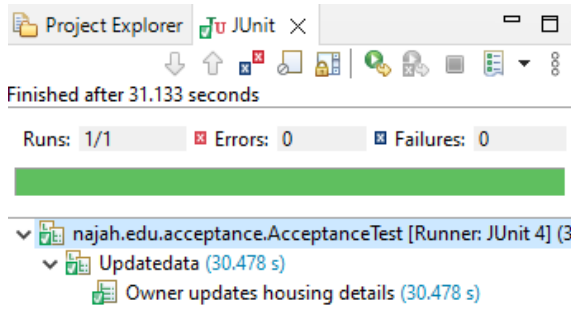
Project Explorer JUnit X

Finished after 51.382 seconds

Runs: 1/1 Errors: 0 Failures: 0

▼ najah.edu.acceptance.AcceptanceTest [Runner: JUnit 4] (50.817 s)  
    > owner add unit (50.817 s)

Failure Trace



In our project we made a 15 features

The system includes 3 menus:

1) **Administrator:**

requests for advertisement of housing and the ability to accept and reject it and add and advertise housing units through the system

2) **owners:**

The owner can add a house and he can also modify the house information The owner can determine the information of the house, is it available, does it contain electricity and water, determines the rent, and shows pictures of the house, and it can do many things related to the house, as it is responsible.

The system that was designed is that there is a building that contains a unit inside it. The owner can determine the characteristics of the unit, the number of rooms, the price, and others. The owner has personal information to contact him in case of booking the house.

3) **tenants**

The tenant can see the available units he can also see the characteristics of the house that he will rent, for example, its price, the number of rooms, and so on he can book the house he wants, and the payment is done. After booking, the tenant can see the characteristics and information of the owner responsible for the unit. After booking, the tenant can see the information he booked and the data he entered. In the event that the tenant has furniture, it can be displayed in case someone wants to buy it.

The work was divided equally between us until we reached this result of the project.

Areej Sawalha:

- 1)Accept-reject-housing
- 2)login and logout
- 3)furniture's Sale
- 4)monthly rent
- 5)Service available
- 6) add house
- 7)view personal information Owner

Baraa Shelbaya:

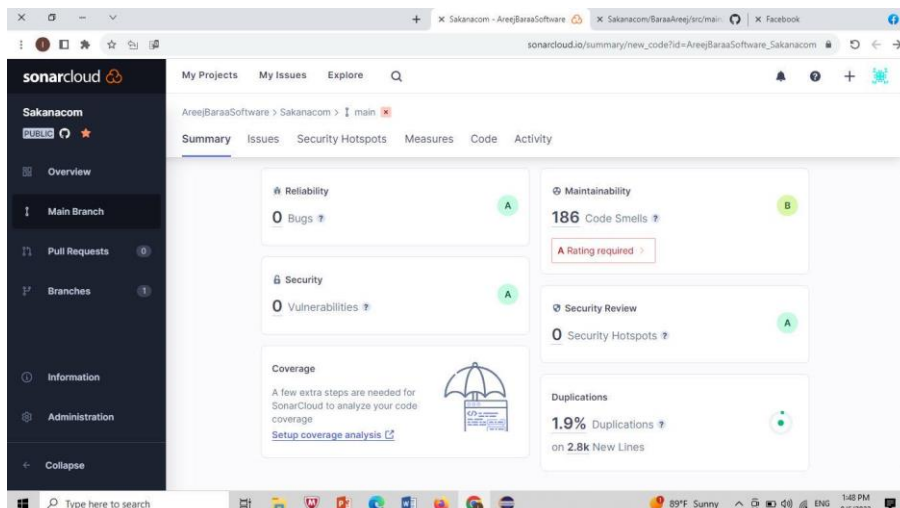
- 1)owner modified house
- 2)owner add unit in the house
- 3)view house
- 4)Tenant Reserved
- 5)view personal information Tenant
- 6) pay Rent
- 7) available house

We created a brand-new organization on GitHub with the intention of moving our repository there. We have incorporated Sonar Cloud into our company to run a complete set of rules encompassing numerous elements of code, such as maintainability, dependability, and security, in order to guarantee the quality and integrity of our code. With the help of this integration, we can automatically check our code for compliance with these standards for every pull request, enhancing the overall quality and security of our codebase.

In the first analysis:



Then, in order to reduce the amount of code smells, we refactored our project.



SonarCloud project summary for Sakanacom (main branch). The interface shows various quality metrics:

- Reliability:** 0 Bugs (A)
- Maintainability:** 166 Code Smells (B), A Rating required
- Security:** 0 Vulnerabilities (A)
- Security Review:** 0 Security Hotspots (A)
- Coverage:** A few extra steps are needed for SonarCloud to analyze your code coverage. [Setup coverage analysis](#)
- Duplications:** 2.4% Duplications on 2.7k New Lines

The left sidebar contains navigation links: Overview, Main Branch, Pull Requests (0), Branches (1), Information, and Administration.

SonarCloud project summary for Sakanacom (main branch) showing a **Passed** Quality Gate. The interface displays the following metrics:

- Reliability:** 0 Bugs (A)
- Maintainability:** 141 Code Smells (A)
- Security:** 0 Vulnerabilities (A)
- Security Review:** 0 Security Hotspots (A)
- Coverage:** A few extra steps are needed for SonarCloud to analyze your code coverage
- Duplications:** 2.4% Duplications

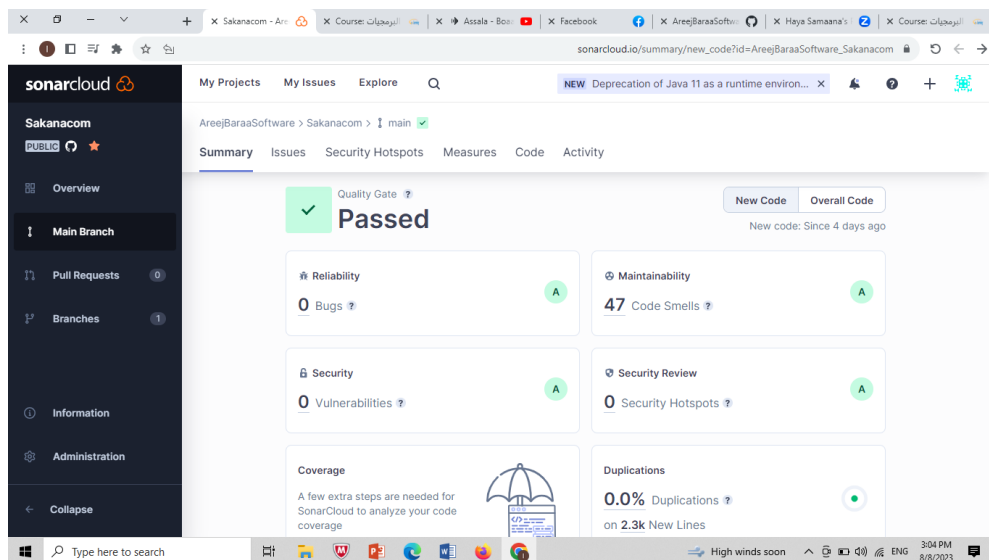
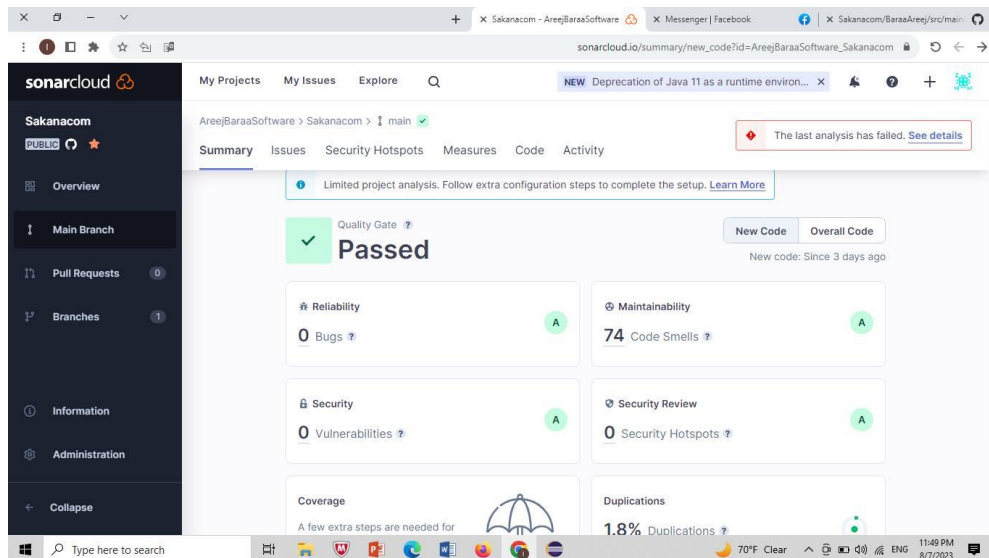
The left sidebar shows a file explorer view of the project structure.

SonarCloud project summary for Sakanacom (main branch) showing a **Passed** Quality Gate. The interface displays the following metrics:

- Reliability:** 0 Bugs (A)
- Maintainability:** 102 Code Smells (A)
- Security:** 0 Vulnerabilities (A)
- Security Review:** 0 Security Hotspots (A)
- Coverage:** A few extra steps are needed for SonarCloud to analyze your code coverage

The left sidebar shows navigation links: Overview, Main Branch, Pull Requests (0), Branches (1), Information, and Administration.





Some example for things we refactored:

1) **System.out.println(" ");** convert it to **LOGGER.log(Level.INFO, " ");**

```
public void displayInfo() {
    for (FurnitureClass f : furnit ) {
        System.out.println("Furniture Type: " +f.getType());
        System.out.println("Material: " + f.getMaterial());
        System.out.println("Price:$" + f.getPrice());
        System.out.println("-----");
    }
}

public void displayInfo() {
    for (FurnitureClass f : furnit ) {
        LOGGER.log(Level.INFO,"Furniture Type:{0}",f.getType()+"");

        LOGGER.log(Level.INFO,"Material:{0}",f.getMaterial()+"");

        LOGGER.log(Level.INFO,"Price:${0} " ,f.getPrice()+"");

        LOGGER.log(Level.INFO,"-----");
    }
}
```

## Result:

```
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: Furniture Type:Table
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: Material:Wood
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: Price:$199.99
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: -----
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: Furniture Type:chiar
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: Material:Wood
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: Price:$200.99
Aug 08, 2023 10:52:41 AM najah.edu.acceptance.FurnitureClass displayInfo
INFO: -----
```

2)

## Duplicate in code

```
public class AdminClass {
    private static final Logger LOGGER = Logger.getLogger(AdminClass.class.getName());
    static final String DASH = " -----";

    for (HousingAdvertisement advertisement : housingAdvertisements) {

        LOGGER.log(Level.INFO, "Title:{0} ", advertisement.getTitle()+" ");
        LOGGER.log(Level.INFO, "Description:{0} ", advertisement.getDescription()+" ");
        LOGGER.log(Level.INFO, "Location:{0} ", advertisement.getLocation()+" ");
        LOGGER.log(Level.INFO, "Owner:{0} ", advertisement.getOwner().getName()+" ");
        LOGGER.log(Level.INFO, "Contact Information: {0}" , advertisement.getOwner().getContactInformation()+" ");
        LOGGER.log(Level.INFO, DASH);
    }
}
```

Because the dash was repeated more than once, we solved the duplex in this way

## 3) Delete the comments because it gives code smells

```
// to view unit
public void viewunit() {

    for (Units Unitlist : unit) {

        LOGGER.log(Level.INFO, "unitnumber: {0}", Unitlist.getUnitNumber()+" ");
    }
}

public void viewunit() {

    for (Units Unitlist : unit) {

        LOGGER.log(Level.INFO, "unitnumber: {0}", Unitlist.getUnitNumber()+" ");
    }
}
```

4) Convert array list to protected or private final static (array list in public is code smell)

```
public ArrayList <TenantClass> Tenant= new ArrayList<TenantClass>();
```

```
private final static List <TenantClass> Tenant= new ArrayList<TenantClass>();
```

5) Convert type variable

```
public static String name;  
public static String phone;  
public static String location;
```

```
private static String name;  
private static String phone;  
private static String location;
```


6) Convert declaration flag

```
public static Boolean owneraddhouse ;  
public static Boolean flagepicture;
```

```
private static boolean owneraddhouse =false ;  
static boolean flagepicture=false;
```

7) Also, naming problems for classes and variables have been solved according to the rules

ownerClass.java

 OwnerClass.java

## Coverage:

After we test our project from test cases, main sources the coverage of code is 91.0

AcceptanceTest (Aug 8, 2023 2:54:59 PM)

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
▼ SoftwareProject	91.0 %	2,843	282	3,125
▼ src/main/java	88.4 %	1,753	229	1,982
▼ najah.edu.acceptance	88.4 %	1,753	229	1,982
> LoginSteps.java	66.7 %	220	110	330
> Units.java	88.7 %	228	29	257
> User.java	0.0 %	0	21	21
> Reservation.java	94.2 %	277	17	294
> FurnitureClass.java	87.7 %	107	15	122
> HousingAdvertisement.java	82.2 %	60	13	73
> OwnerClass.java	78.0 %	39	11	50
> TenantClass.java	0.0 %	0	8	8
> Main.java	96.3 %	104	4	108
> AdminClass.java	99.9 %	718	1	719
▼ src/test/java	95.4 %	1,090	53	1,143
▼ najah.edu.acceptance	95.4 %	1,090	53	1,143
> Login.java	93.0 %	66	5	71
> acceptrejecthousing.java	95.1 %	78	4	82
> AdminAddhouse.java	95.1 %	78	4	82
> Logout.java	93.2 %	55	4	59
> AcceptanceTest.java	0.0 %	0	3	3
> Funituresale.java	95.5 %	63	3	66
> Monthlyrent.java	95.5 %	63	3	66
> Owneraddhouse.java	96.8 %	92	3	95
> Ownermodified.java	96.6 %	85	3	88
> Personaldatenenant.java	96.3 %	79	3	82
> Serviceavailable.java	95.7 %	67	3	70
> Tenantpayrent.java	96.2 %	75	3	78
> tenantviewavailablehouse.java	95.9 %	70	3	73

TTT - SoftwareProject/src/test/java/najah.edu/acceptance/AcceptanceTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Outline | Coverage

AcceptanceTest (Aug 8, 2023 2:54:59 PM)

Finished after 221.511 seconds

Runs: 17/17 | Errors: 0 | Failures: 0

> najah.edu.acceptance.AcceptanceTest [Runner: JUnit]

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
▼ SoftwareProject	91.0 %	2,843	282	3,125
▼ src/main/java	88.4 %	1,753	229	1,982
▼ najah.edu.acceptance	88.4 %	1,753	229	1,982
> LoginSteps.java	66.7 %	220	110	330
> Units.java	88.7 %	228	29	257
> User.java	0.0 %	0	21	21
> Reservation.java	94.2 %	277	17	294
> FurnitureClass.java	87.7 %	107	15	122
> HousingAdvertisement.java	82.2 %	60	13	73
> OwnerClass.java	78.0 %	39	11	50
> TenantClass.java	0.0 %	0	8	8
> Main.java	96.3 %	104	4	108
> AdminClass.java	99.9 %	718	1	719
▼ src/test/java	95.4 %	1,090	53	1,143
▼ najah.edu.acceptance	95.4 %	1,090	53	1,143
> Login.java	93.0 %	66	5	71
> acceptrejecthousing.java	95.1 %	78	4	82
> AdminAddhouse.java	95.1 %	78	4	82
> Logout.java	93.2 %	55	4	59
> AcceptanceTest.java	0.0 %	0	3	3
> Funituresale.java	95.5 %	63	3	66
> Monthlyrent.java	95.5 %	63	3	66
> Owneraddhouse.java	96.8 %	92	3	95
> Ownermodified.java	96.6 %	85	3	88
> Personaldatenenant.java	96.3 %	79	3	82
> Serviceavailable.java	95.7 %	67	3	70
> Tenantpayrent.java	96.2 %	75	3	78
> tenantviewavailablehouse.java	95.9 %	70	3	73
> Tenantviewownerdetails.java	95.9 %	70	3	73
> TenantReserved.java	96.3 %	79	3	82
> Viewhouse.java	95.9 %	70	3	73

Failure Trace

Type here to search

USD/EUR +0.58%

3:00 PM 8/8/2023

# When we build project in Jenkins:

The screenshot shows the Jenkins web interface with the 'Console Output' tab selected. The build process is successful, as indicated by the green checkmark icon. The output text shows the following steps:

- Started by timer
- Running as SYSTEM
- Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Sakanacom
- The recommended git tool is: NONE
- No credentials specified
- Fetching changes from the remote Git repository
- Fetching upstream changes from https://github.com/AreejBaraaSoftware/Sakanacom.git
- Checking out Revision 28b061752205c8dabb7f3a5014eb0cbd6cc378f3 (refs/remotes/origin/main)
- Commit message: "Delete SoftwareProject directory"
- Deleting C:\ProgramData\Jenkins\jenkins\workspace\Sakanacom\target
- Scanning for projects...
- Building SoftwareProject 0.0.1-SNAPSHOT
- from pom.xml
- [ jar ]
- clean:3.2.0:clean (default-clean) @ SoftwareProject
- Deleting C:\ProgramData\Jenkins\jenkins\workspace\Sakanacom\target

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock indicating 8:01 PM on 8/8/2023.

This screenshot is a zoomed-in view of the Jenkins Console Output. It shows the same build process as the first screenshot, with the following key details:

- Started by timer
- Running as SYSTEM
- Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Sakanacom
- The recommended git tool is: NONE
- No credentials specified
- Fetching changes from the remote Git repository
- Fetching upstream changes from https://github.com/AreejBaraaSoftware/Sakanacom.git
- Checking out Revision 28b061752205c8dabb7f3a5014eb0cbd6cc378f3 (refs/remotes/origin/main)
- Commit message: "Delete SoftwareProject directory"
- Deleting C:\ProgramData\Jenkins\jenkins\workspace\Sakanacom\target
- Scanning for projects...
- Building SoftwareProject 0.0.1-SNAPSHOT
- from pom.xml
- [ jar ]
- clean:3.2.0:clean (default-clean) @ SoftwareProject

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  ???? s
[INFO] Finished at:  ???-??-??T?:?:??:03:00
[INFO] -----
Finished: SUCCESS
```

Sakanacom > #52 > Console Output

```
[Sakanacom] $ cmd.exe /C "C:\apache-maven-3.9.4-bin\apache-maven-3.9.4\bin\mvn.cmd clean compile && exit %%ERRORLEVEL%%"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< SoftwareProject:SoftwareProject >-----
[INFO] Building SoftwareProject 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ SoftwareProject ---
[INFO] Deleting C:\ProgramData\Jenkins\.jenkins\workspace\Sakanacom\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ SoftwareProject ---
[WARNING] Using platform encoding (Cp1256 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\.jenkins\workspace\Sakanacom\src\main\resources
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ SoftwareProject ---
[INFO] Changes detected - recompiling the module! :source
[WARNING] File encoding has not been set, using platform encoding Cp1256, i.e. build is platform dependent!
[INFO] Compiling 10 source files with javac [debug target 17] to target\classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time:  ?.??? s
[INFO] Finished at:  ???-?-??T?:?:??+03:00
[INFO] -----
Finished: SUCCESS
```