An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Department of Computer Engineering

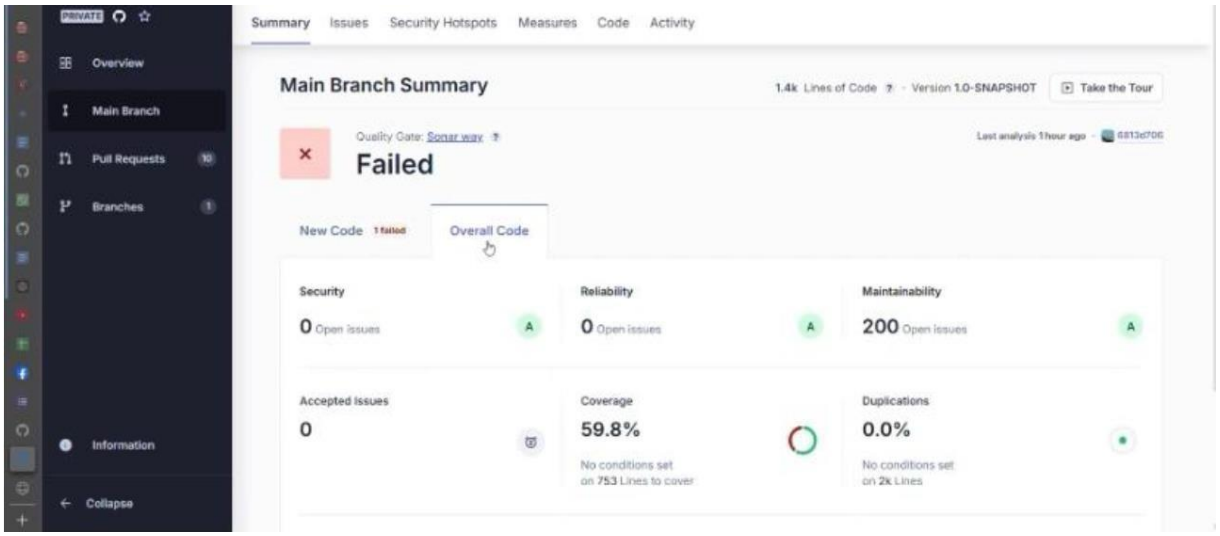
Course Name: Software Engineering **Course Number: 10636312**

Assignment Report

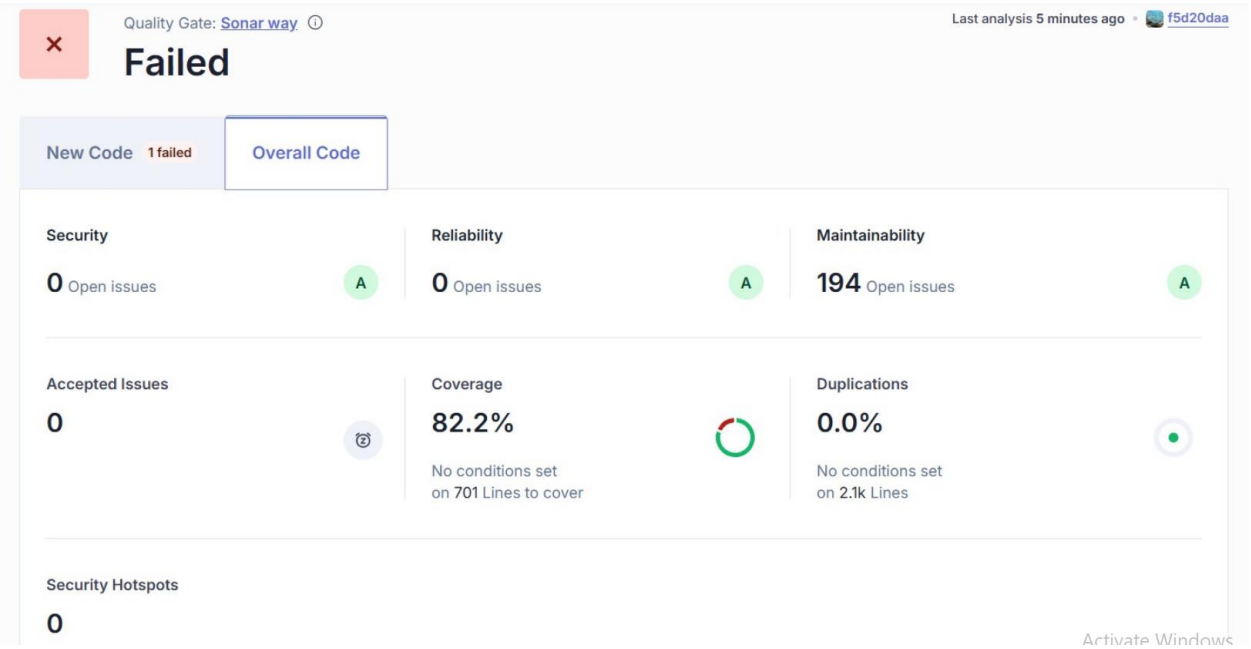
Instructor: Dr. Haya Samaana
Academic Year: 2024 - 2025
Semester: 2nd

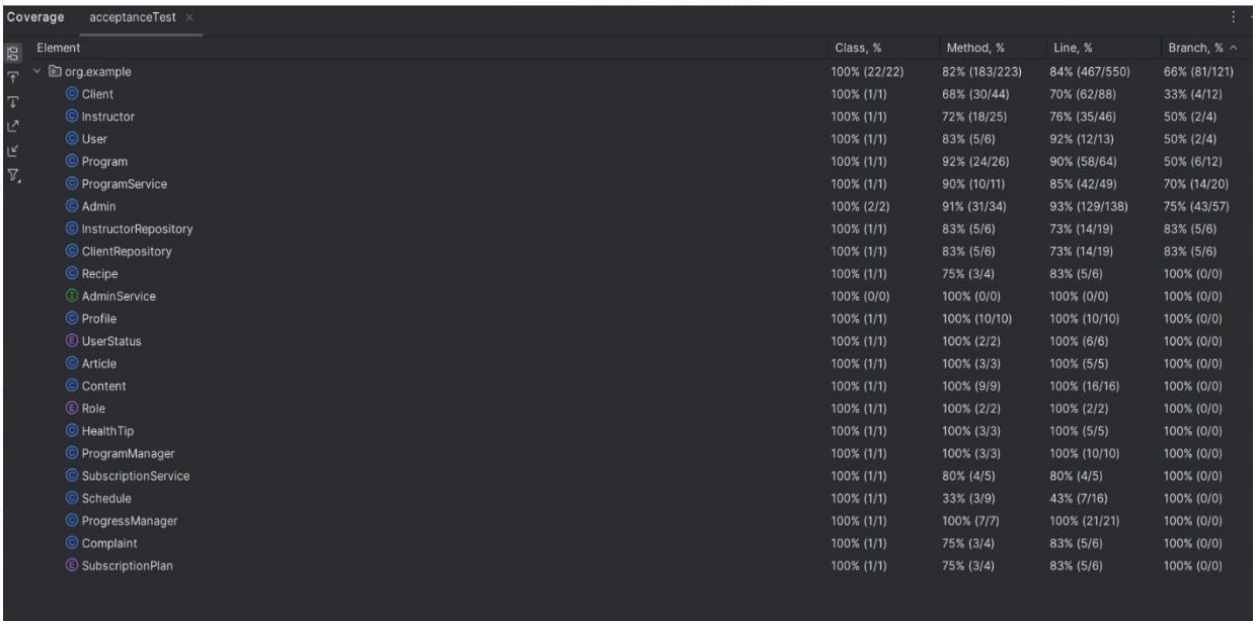
Students	
1- Taqwa Odeh	2- Ghayda Saify
3- Tala Alhendi	

- **The results before refactoring :**



- The enhanced code before doing the 4 bad smells refactoring (Coverage >81%)



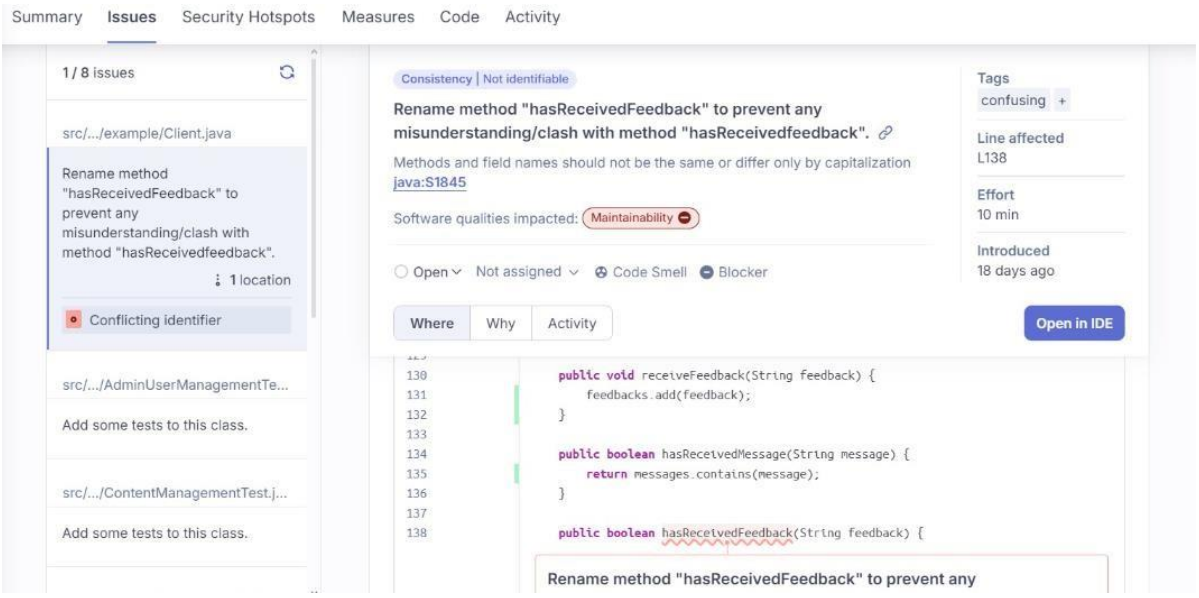


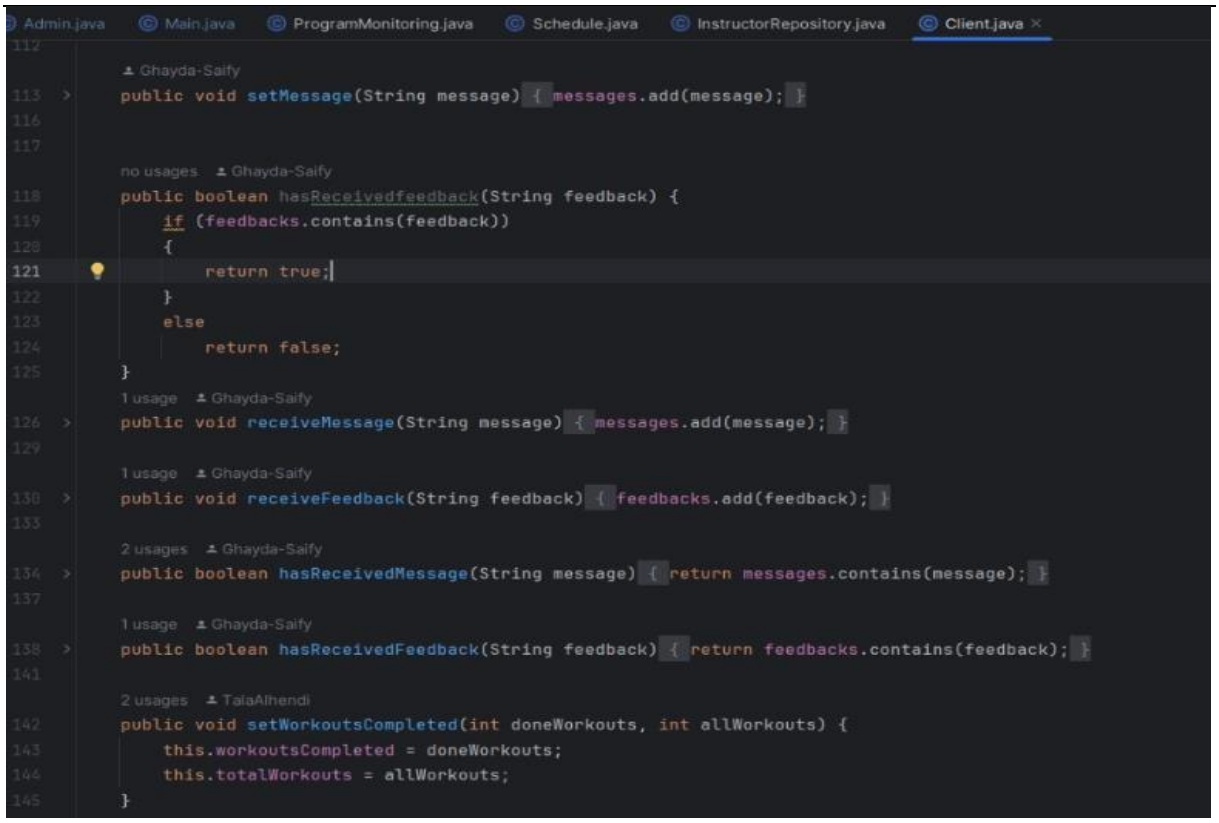
Coverage: acceptanceTest

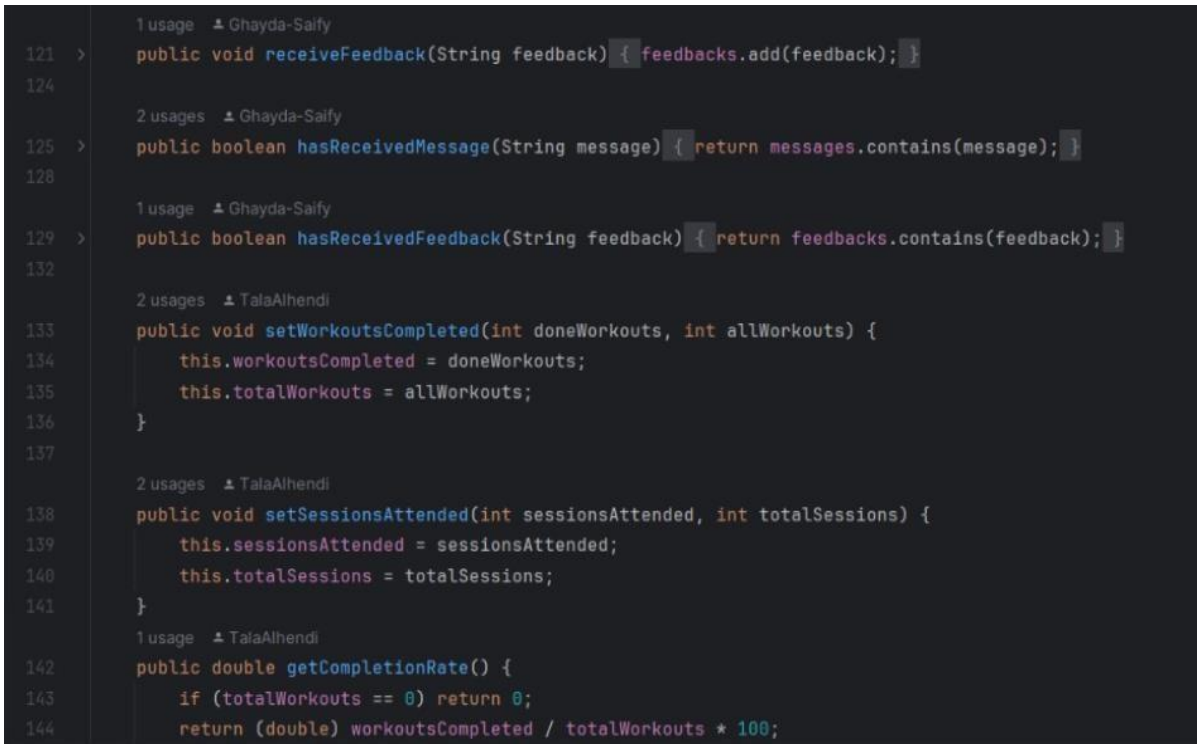
Element	Class, %	Method, %	Line, %	Branch, %
org.example	100% (22/22)	82% (183/223)	84% (467/550)	66% (81/121)
Client	100% (1/1)	68% (30/44)	70% (62/88)	33% (4/12)
Instructor	100% (1/1)	72% (18/25)	76% (35/46)	50% (2/4)
User	100% (1/1)	83% (5/6)	92% (12/13)	50% (2/4)
Program	100% (1/1)	92% (24/26)	90% (58/64)	50% (6/12)
ProgramService	100% (1/1)	90% (10/11)	85% (42/49)	70% (14/20)
Admin	100% (2/2)	91% (31/34)	93% (129/138)	75% (43/57)
InstructorRepository	100% (1/1)	83% (5/6)	73% (14/19)	83% (5/6)
ClientRepository	100% (1/1)	83% (5/6)	73% (14/19)	83% (5/6)
Recipe	100% (1/1)	75% (3/4)	83% (5/6)	100% (0/0)
AdminService	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
Profile	100% (1/1)	100% (10/10)	100% (10/10)	100% (0/0)
UserStatus	100% (1/1)	100% (2/2)	100% (6/6)	100% (0/0)
Article	100% (1/1)	100% (3/3)	100% (5/5)	100% (0/0)
Content	100% (1/1)	100% (9/9)	100% (16/16)	100% (0/0)
Role	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
HealthTip	100% (1/1)	100% (3/3)	100% (5/5)	100% (0/0)
ProgramManager	100% (1/1)	100% (3/3)	100% (10/10)	100% (0/0)
SubscriptionService	100% (1/1)	80% (4/5)	80% (4/5)	100% (0/0)
Schedule	100% (1/1)	33% (3/9)	43% (7/16)	100% (0/0)
ProgressManager	100% (1/1)	100% (7/7)	100% (21/21)	100% (0/0)
Complaint	100% (1/1)	75% (3/4)	83% (5/6)	100% (0/0)
SubscriptionPlan	100% (1/1)	75% (3/4)	83% (5/6)	100% (0/0)

Refactoring 4 bad smells :

- **Renaming bad smell:**




Admin.java
Main.java
ProgramMonitoring.java
Schedule.java
InstructorRepository.java
Client.java
Ghayda-Saify
public void setMessage(String message) { messages.add(message); }
no usages Ghayda-Saify
public boolean hasReceivedfeedback(String feedback) {
if (feedbacks.contains(feedback))
{
return true;
}
else
return false;
}
1 usage Ghayda-Saify
public void receiveMessage(String message) { messages.add(message); }
1 usage Ghayda-Saify
public void receiveFeedback(String feedback) { feedbacks.add(feedback); }
2 usages Ghayda-Saify
public boolean hasReceivedMessage(String message) { return messages.contains(message); }
1 usage Ghayda-Saify
public boolean hasReceivedFeedback(String feedback) { return feedbacks.contains(feedback); }
2 usages TalaAlhendi
public void setWorkoutsCompleted(int doneWorkouts, int allWorkouts) {
this.workoutsCompleted = doneWorkouts;
this.totalWorkouts = allWorkouts;
}


1 usage Ghayda-Saify
public void receiveFeedback(String feedback) { feedbacks.add(feedback); }
2 usages Ghayda-Saify
public boolean hasReceivedMessage(String message) { return messages.contains(message); }
1 usage Ghayda-Saify
public boolean hasReceivedFeedback(String feedback) { return feedbacks.contains(feedback); }
2 usages TalaAlhendi
public void setWorkoutsCompleted(int doneWorkouts, int allWorkouts) {
this.workoutsCompleted = doneWorkouts;
this.totalWorkouts = allWorkouts;
}
2 usages TalaAlhendi
public void setSessionsAttended(int sessionsAttended, int totalSessions) {
this.sessionsAttended = sessionsAttended;
this.totalSessions = totalSessions;
}
1 usage TalaAlhendi
public double getCompletionRate() {
if (totalWorkouts == 0) return 0;
return (double) workoutsCompleted / totalWorkouts * 100;

- **Duplication bad smell:**

Before:

```
        List<Map<String, String>> revenueReport = new ArrayList<>();
        for (Program program : programs) {
            double revenue = program
.getClientsEnrolled().size() *Double.parseDouble( program.getPrice());
            revenueReport.add(Map.of(
                    "Program Name", program.getTitle(),
```

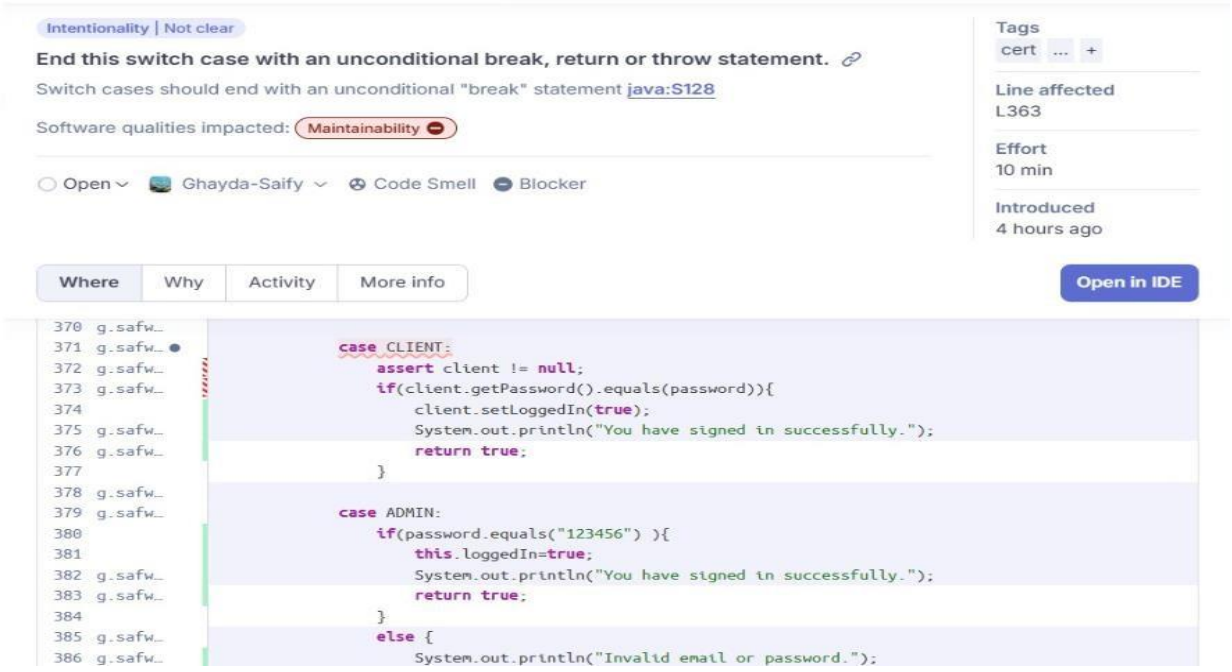
Define a constant instead of duplicating this literal "Program Name" 4 times.

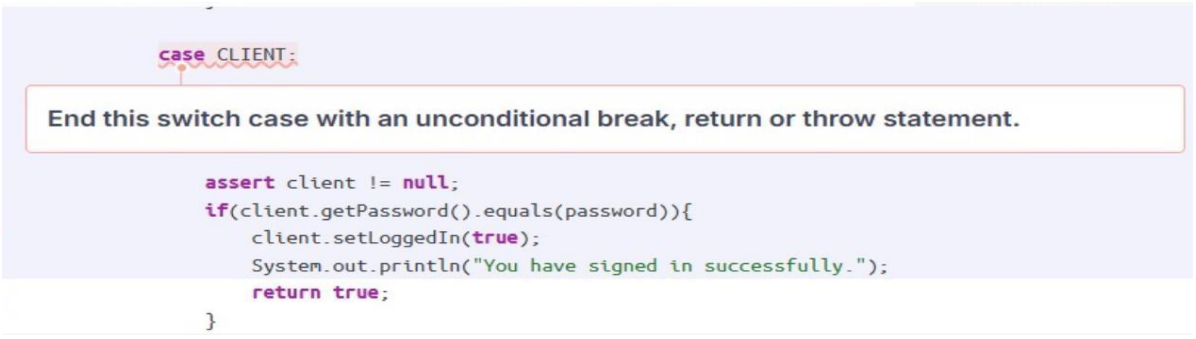
```
                    "Revenue", String.valueOf(revenue)
            ));
        }
        return revenueReport;
    }
    /**
     * Retrieves the statuses of all programs as a table, represented as a list of
maps.
     * Each map contains the following keys:
     * - "Program NAME" - the name of the program.
     * - "Status" - the status of the program ("Upcoming", "Active", or
"Completed").
```


After:

```
     * - "Active": The program is currently active.
     * - "Completed": The program has already finished.
     *
     * @return a list of maps where each map represents a program and its corresponding status.
     */
    1 usage   Ghayda-Saify *
    public List<Map<String, String>> getProgramStatusesAsTable() {
        List<Map<String,String>> resutl = new ArrayList<>();
        for(Program p : programs){
            if(p.getStartDate().after(new Date())){
                resutl.add(Map.of(
                        programNameDuplicated, p.getTitle(),
                        statusDuplicated,  v2: "Upcoming"
                ));
            }
            else if(p.getEndtDate().after(new Date())){
                resutl.add(Map.of(
                        programNameDuplicated, p.getTitle(),
                        statusDuplicated,  v2: "Active"
                ));
            }
            else if(p.getEndtDate().before(new Date())){
                resutl.add(Map.of(
                        programNameDuplicated, p.getTitle(),
                        statusDuplicated,  v2: "Completed"
                ));
            }
        }
```

- End the switch case with an unconditional break , return of throw statement





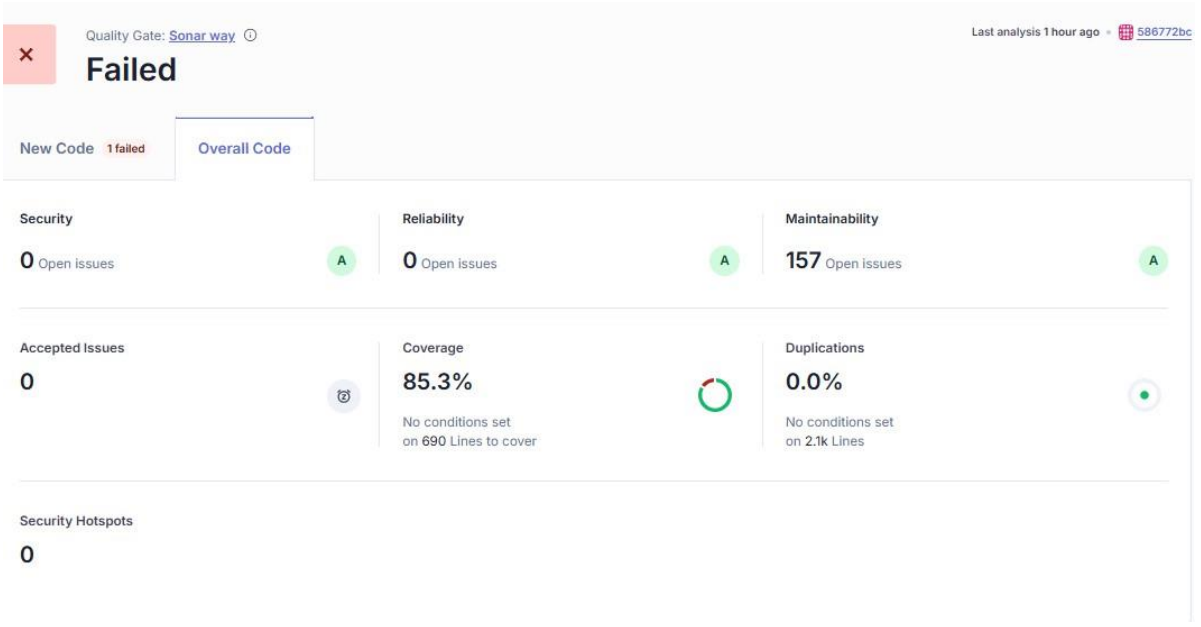

```
switch (role){
    case INSTRUCTOR:
        assert instructor != null;
        if(instructor.getPassword().equals(PASSWORD)){
            instructor.setLoggedIn(true);
            System.out.println("You have signed in successfully.");
            return true;
        }
        return false;

    case CLIENT:
        assert client != null;
        if(client.getPassword().equals(PASSWORD)){
            client.setLoggedIn(true);
            System.out.println("You have signed in successfully.");
            return true;
        }
        return false;
```

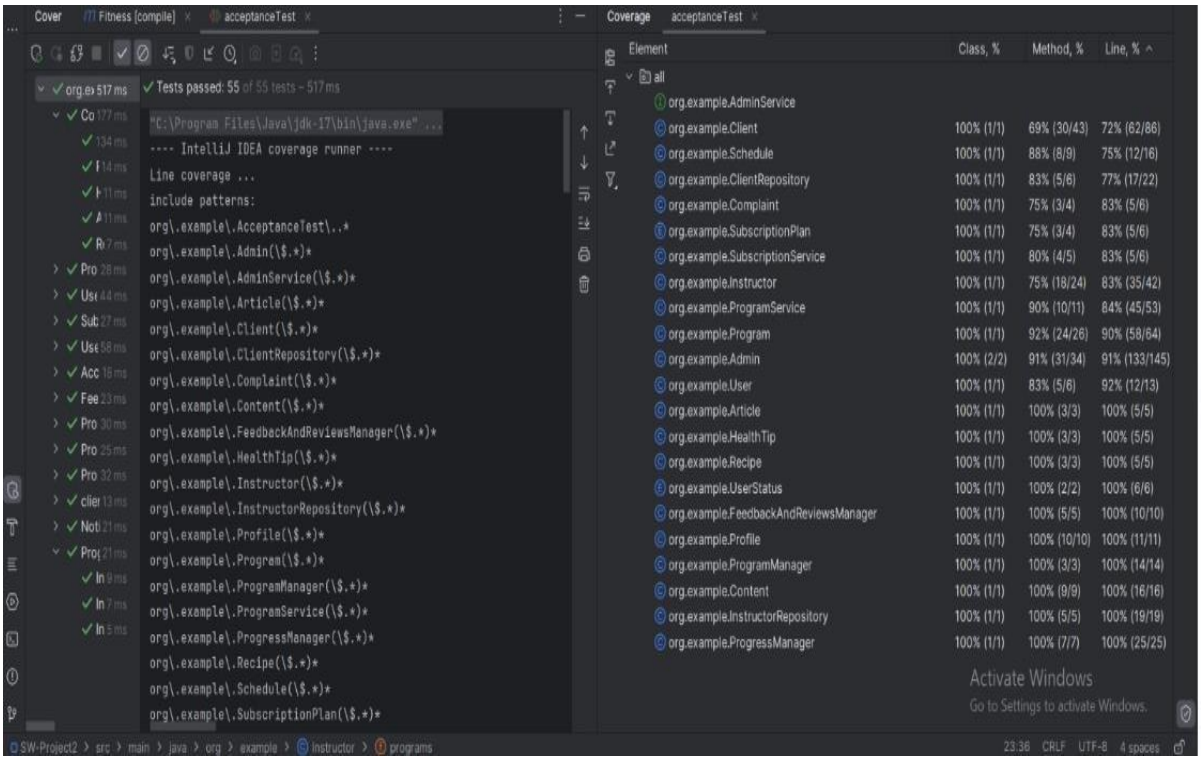
The result after refactoring :

Coverage | acceptanceTest

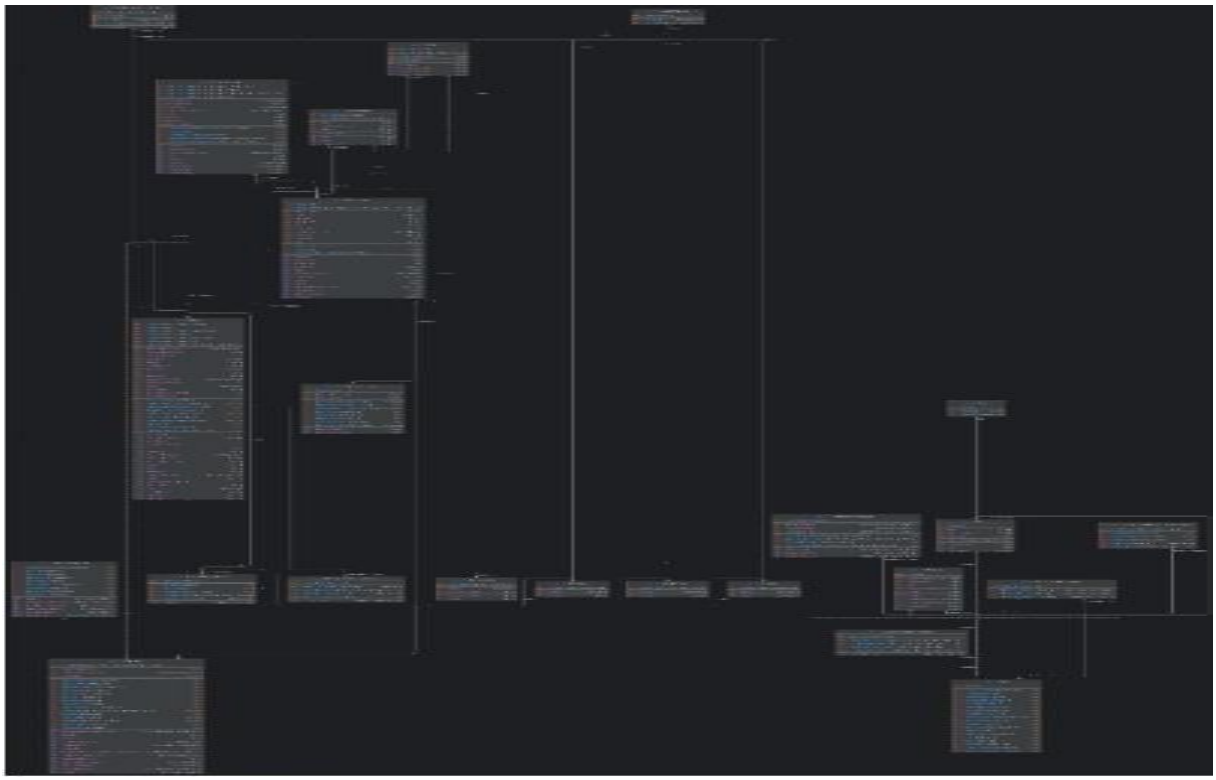
Element	Class, %	Method, %	Line, %	Branch, %
org.example	100% (23/23)	86% (193/224)	87% (482/551)	66% (81/121)
Client	100% (1/1)	68% (30/44)	70% (62/88)	33% (4/12)
Instructor	100% (1/1)	75% (18/24)	83% (35/42)	50% (2/4)
User	100% (1/1)	83% (5/6)	92% (12/13)	50% (2/4)
Program	100% (1/1)	92% (24/26)	90% (58/64)	50% (6/12)
ProgramService	100% (1/1)	90% (10/11)	85% (42/49)	70% (14/20)
Admin	100% (2/2)	93% (31/33)	92% (131/141)	75% (43/57)
InstructorRepository	100% (1/1)	100% (5/5)	100% (16/16)	83% (5/6)
ClientRepository	100% (1/1)	83% (5/6)	73% (14/19)	83% (5/6)
FeedbackAndReviewsManager	100% (1/1)	100% (5/5)	100% (6/6)	100% (0/0)
Recipe	100% (1/1)	100% (3/3)	100% (5/5)	100% (0/0)
AdminService	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
Profile	100% (1/1)	100% (10/10)	100% (10/10)	100% (0/0)
UserStatus	100% (1/1)	100% (2/2)	100% (6/6)	100% (0/0)
Article	100% (1/1)	100% (3/3)	100% (5/5)	100% (0/0)
Content	100% (1/1)	100% (9/9)	100% (16/16)	100% (0/0)
Role	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
HealthTip	100% (1/1)	100% (3/3)	100% (5/5)	100% (0/0)
ProgramManager	100% (1/1)	100% (3/3)	100% (10/10)	100% (0/0)
SubscriptionService	100% (1/1)	80% (4/5)	80% (4/5)	100% (0/0)
Schedule	100% (1/1)	88% (8/9)	75% (12/16)	100% (0/0)
ProgressManager	100% (1/1)	100% (7/7)	100% (21/21)	100% (0/0)
Complaint	100% (1/1)	75% (3/4)	83% (5/6)	100% (0/0)
SubscriptionPlan	100% (1/1)	75% (3/4)	83% (5/6)	100% (0/0)



All test Cases True :



The generated UML diagram:



[The UML diagram pic for clearer version, open it with draw.io for better clarity](#).

Thank You Dr. Haya 😊

Feature Completed ✅
Phase 2 😳: