Assignment 2

Introduction

To improve our code we had to refactor some methods and classes. To do so, we used the MetricsTree Intellij plugin, which shows metrics on a class- and method-level. This way, scanning through classes and methods, we could find the most problematic ones. The MetricsTree plugin uses a color system, where green = great, yellow = ok, orange = bad, red = terrible. We looked for orange and red metrics and improved the ones that needed to be improved.

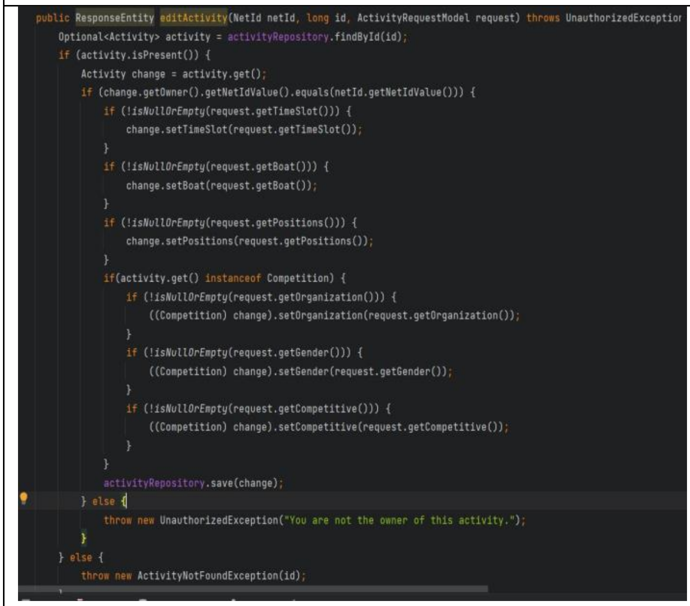
We left some of them as they were because it would not make sense to change anything, e.g. we did not change **Number Of Parameters** in any method because we needed all of them, and the parameters were not general enough to create one object and pass it instead.

Method-level refactoring

We noticed that our methods mainly had a problem with **Lines of Code** and **Cyclomatic Complexity.** To improve both metrics we used **Extract Method Refactoring**, as moving some code to different methods decreases lines of code and also makes the method less complex.

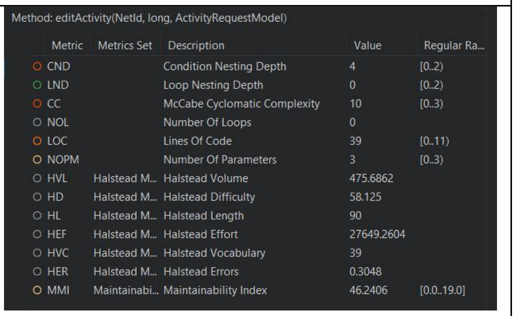
1. Activity-microservice/src/main/java/nl/tudelft/sem/template/example/domain/ActivityServiceEdit.java - **editActivity**

Code Before



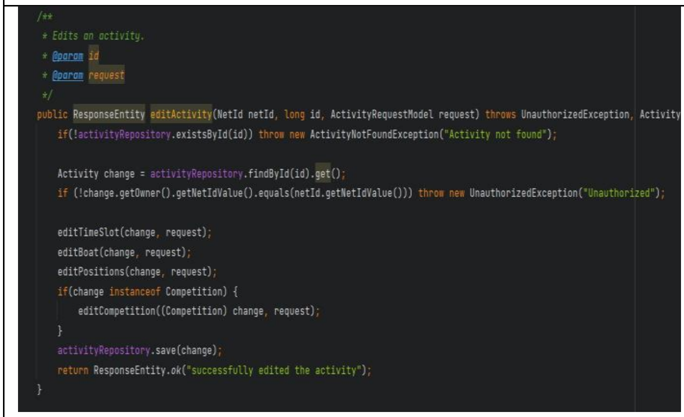
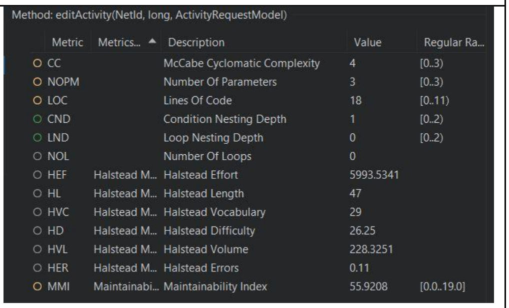
```
public ResponseEntity editActivity(NetId netId, long id, ActivityRequestModel request) throws UnauthorizedExceptior
    Optional<Activity> activity = activityRepository.findById(id);
    if (activity.isPresent()) {
        Activity change = activity.get();
        if (change.getOwner().getNetIdValue().equals(netId.getNetIdValue())) {
            if (!isNullOrEmpty(request.getTimeSlot())) {
                change.setTimeSlot(request.getTimeSlot());
            }
            if (!isNullOrEmpty(request.getBoat())) {
                change.setBoat(request.getBoat());
            }
            if (!isNullOrEmpty(request.getPositions())) {
                change.setPositions(request.getPositions());
            }
            if(activity.get() instanceof Competition) {
                if (!isNullOrEmpty(request.getOrganization())) {
                    ((Competition) change).setOrganization(request.getOrganization());
                }
                if (!isNullOrEmpty(request.getGender())) {
                    ((Competition) change).setGender(request.getGender());
                }
                if (!isNullOrEmpty(request.getCompetitive())) {
                    ((Competition) change).setCompetitive(request.getCompetitive());
                }
            }
            activityRepository.save(change);
        } else {
            throw new UnauthorizedException("You are not the owner of this activity.");
        }
    } else {
        throw new ActivityNotFoundException(id);
```

Metrics Before



Method: editActivity(NetId, long, ActivityRequestModel)

	Metric	Metrics Set	Description	Value	Regular Ra...
○	CND		Condition Nesting Depth	4	[0..2)
○	LND		Loop Nesting Depth	0	[0..2)
○	CC		McCabe Cyclomatic Complexity	10	[0..3)
○	NOL		Number Of Loops	0	
○	LOC		Lines Of Code	39	[0..11)
○	NOPM		Number Of Parameters	3	[0..3)
○	HVL	Halstead M...	Halstead Volume	475.6862	
○	HD	Halstead M...	Halstead Difficulty	58.125	
○	HL	Halstead M...	Halstead Length	90	
○	HEF	Halstead M...	Halstead Effort	27649.2604	
○	HVC	Halstead M...	Halstead Vocabulary	39	
○	HER	Halstead M...	Halstead Errors	0.3048	
○	MMI	Maintainabi...	Maintainability Index	46.2406	[0.0..19.0]

Code After	Metrics After
	

```java
/**
 * Edits an activity.
 * @param id
 * @param request
 */
public ResponseEntity editActivity(NetId netId, long id, ActivityRequestModel request) throws UnauthorizedException, Activity
    if(!activityRepository.existsById(id)) throw new ActivityNotFoundException("Activity not found");

    Activity change = activityRepository.findById(id).get();
    if (!change.getOwner().getNetIdValue().equals(netId.getNetIdValue())) throw new UnauthorizedException("Unauthorized");

    editTimeSlot(change, request);
    editBoat(change, request);
    editPositions(change, request);
    if(change instanceof Competition) {
        editCompetition((Competition) change, request);
    }
    activityRepository.save(change);
    return ResponseEntity.ok("successfully edited the activity");
}
```

Method: editActivity(NetId, long, ActivityRequestModel)

Metric	Metrics... ▲	Description	Value	Regular Ra...
○ CC		McCabe Cyclomatic Complexity	4	[0..3)
○ NOPM		Number Of Parameters	3	[0..3)
○ LOC		Lines Of Code	18	[0..11)
○ CND		Condition Nesting Depth	1	[0..2)
○ LND		Loop Nesting Depth	0	[0..2)
○ NOL		Number Of Loops	0	
○ HEF	Halstead M...	Halstead Effort	5993.5341	
○ HL	Halstead M...	Halstead Length	47	
○ HVC	Halstead M...	Halstead Vocabulary	29	
○ HD	Halstead M...	Halstead Difficulty	26.25	
○ HVL	Halstead M...	Halstead Volume	228.3251	
○ HER	Halstead M...	Halstead Errors	0.11	
○ MMI	Maintainabi...	Maintainability Index	55.9208	[0.0..19.0]

There were a few red and orange metrics which needed to be improved. Specifically:

- Condition Nesting Depth = 4
- Cyclomatic Complexity = 10
- Lines of code = 39

To improve **Condition Nesting Depth**, we moved the “if” statements and first checked expressions that could throw exceptions. Then, if no exceptions were thrown, the method was allowed to change the edited fields. To remove nesting when checking if the activity is a Competition, we created a separate method for editing competitions’ fields.

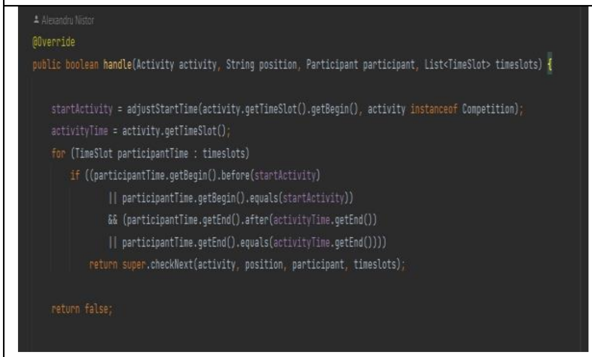
To improve **Cyclomatic Complexity**, instead of using if statements for editing each field, we created separate methods to do that.

The above actions also improved **Lines of code**. New values are as follows:

- Condition Nesting Depth = 1
- Cyclomatic Complexity = 4
- Lines of code = 18

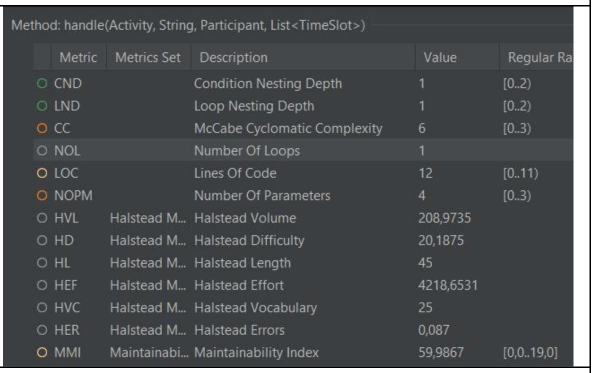
2. matcher-microservice/src/main/java/nl/tudelft/sem/template/example/domain/chainOfResponsability/TimeSlotValidator.java - **handle**

Code Before



```
Alexandru Nistor
@Override
public boolean handle(Activity activity, String position, Participant participant, List<TimeSlot> timeslots) {

    startActivity = adjustStartTime(activity.getTimeSlot().getBegin(), activity instanceof Competition);
    activityTime = activity.getTimeSlot();
    for (TimeSlot participantTime : timeslots)
        if ((participantTime.getBegin().before(startActivity)
                || participantTime.getBegin().equals(startActivity))
                && (participantTime.getEnd().after(activityTime.getEnd())
                || participantTime.getEnd().equals(activityTime.getEnd())))
            return super.checkNext(activity, position, participant, timeslots);

    return false;
```

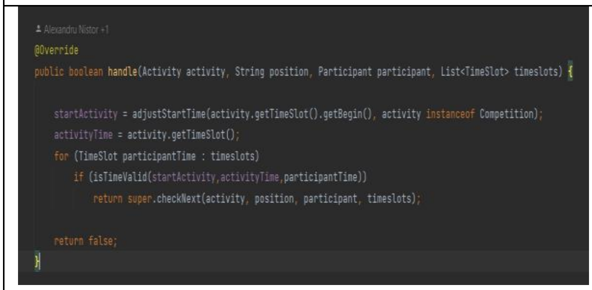
Metrics Before



Method: handle(Activity, String, Participant, List<TimeSlot>)

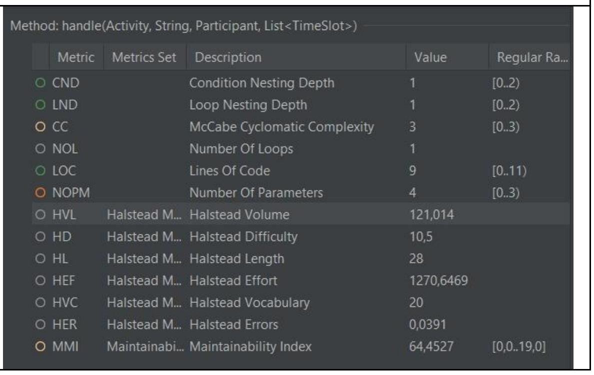
Metric	Metrics Set	Description	Value	Regular Ra
○ CND		Condition Nesting Depth	1	[0..2)
○ LND		Loop Nesting Depth	1	[0..2)
○ CC		McCabe Cyclomatic Complexity	6	[0..3)
○ NOL		Number Of Loops	1	
○ LOC		Lines Of Code	12	[0..11)
○ NOPM		Number Of Parameters	4	[0..3)
○ HVL	Halstead M...	Halstead Volume	208,9735	
○ HD	Halstead M...	Halstead Difficulty	20,1875	
○ HL	Halstead M...	Halstead Length	45	
○ HEF	Halstead M...	Halstead Effort	4218,6531	
○ HVC	Halstead M...	Halstead Vocabulary	25	
○ HER	Halstead M...	Halstead Errors	0,087	
○ MMI	Maintainabi...	Maintainability Index	59,9867	[0,0..19,0]

Code After



```
Alexandru Nistor +1
@Override
public boolean handle(Activity activity, String position, Participant participant, List<TimeSlot> timeslots) {

    startActivity = adjustStartTime(activity.getTimeSlot().getBegin(), activity instanceof Competition);
    activityTime = activity.getTimeSlot();
    for (TimeSlot participantTime : timeslots)
        if (isTimeValid(startActivity,activityTime,participantTime))
            return super.checkNext(activity, position, participant, timeslots);

    return false;
}
```

Metrics After



Method: handle(Activity, String, Participant, List<TimeSlot>)

Metric	Metrics Set	Description	Value	Regular Ra...
○ CND		Condition Nesting Depth	1	[0..2)
○ LND		Loop Nesting Depth	1	[0..2)
○ CC		McCabe Cyclomatic Complexity	3	[0..3)
○ NOL		Number Of Loops	1	
○ LOC		Lines Of Code	9	[0..11)
○ NOPM		Number Of Parameters	4	[0..3)
○ HVL	Halstead M...	Halstead Volume	121,014	
○ HD	Halstead M...	Halstead Difficulty	10,5	
○ HL	Halstead M...	Halstead Length	28	
○ HEF	Halstead M...	Halstead Effort	1270,6469	
○ HVC	Halstead M...	Halstead Vocabulary	20	
○ HER	Halstead M...	Halstead Errors	0,0391	
○ MMI	Maintainabi...	Maintainability Index	64,4527	[0,0..19,0]

There were a few orange metrics which needed to be improved. Specifically:

- Cyclomatic Complexity = 6
- Lines of code = 12

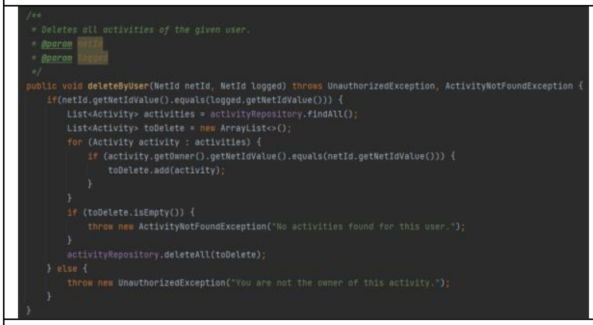
To improve the Cyclomatic Complexity, instead of verifying if the user is available to participate in the activity in a specific timeslot, we created a new method **isTimeValid** that checks this.

New values are as follows:

- Cyclomatic Complexity = 3
- Lines of code = 9

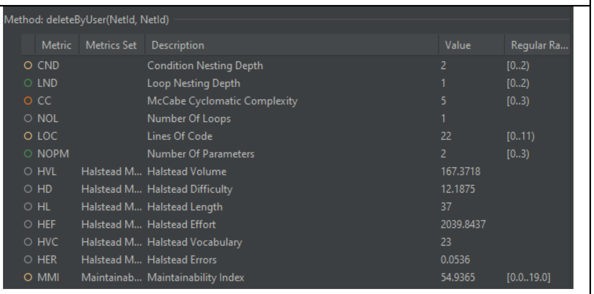
3. Activity-microservice/src/main/java/nl/tudelft/sem/template/example/domain/ActivityServiceCreateDelete.java - **deleteByUser**

Code Before



```java
/**
 * Deletes all activities of the given user.
 * @param netId
 * @param logged
 */
public void deleteByUser(NetId netId, NetId logged) throws UnauthorizedException, ActivityNotFoundException {
    if(netId.getNetIdValue().equals(logged.getNetIdValue())) {
        List<Activity> activities = activityRepository.findAll();
        List<Activity> toDelete = new ArrayList<>();
        for (Activity activity : activities) {
            if (activity.getOwner().getNetIdValue().equals(netId.getNetIdValue())) {
                toDelete.add(activity);
            }
        }
        if (toDelete.isEmpty()) {
            throw new ActivityNotFoundException("No activities found for this user.");
        }
        activityRepository.deleteAll(toDelete);
    } else {
        throw new UnauthorizedException("You are not the owner of this activity.");
    }
}
```

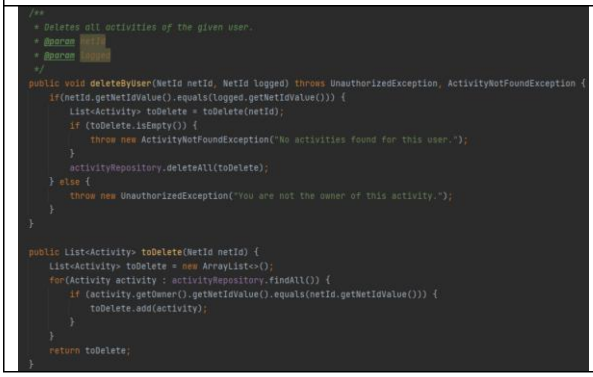
Metrics Before



Method: deleteByUser(NetId, NetId)

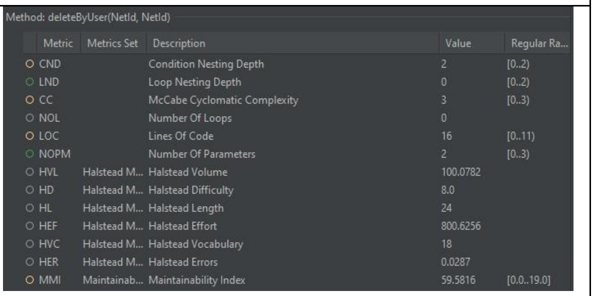
Metric	Metrics Set	Description	Value	Regular Ra...
CND		Condition Nesting Depth	2	[0..2)
LND		Loop Nesting Depth	1	[0..2)
CC		McCabe Cyclomatic Complexity	5	[0..3)
NOL		Number Of Loops	1	
LOC		Lines Of Code	22	[0..11)
NOPM		Number Of Parameters	2	[0..3)
HVL	Halstead M...	Halstead Volume	167.3718	
HD	Halstead M...	Halstead Difficulty	12.1875	
HL	Halstead M...	Halstead Length	37	
HEF	Halstead M...	Halstead Effort	2039.8437	
HVC	Halstead M...	Halstead Vocabulary	23	
HER	Halstead M...	Halstead Errors	0.0536	
MMI	Maintainab...	Maintainability Index	54.9365	[0.0..19.0]

Code After



```java
/**
 * Deletes all activities of the given user.
 * @param netId
 * @param logged
 */
public void deleteByUser(NetId netId, NetId logged) throws UnauthorizedException, ActivityNotFoundException {
    if(netId.getNetIdValue().equals(logged.getNetIdValue())) {
        List<Activity> toDelete = toDelete(netId);
        if (toDelete.isEmpty()) {
            throw new ActivityNotFoundException("No activities found for this user.");
        }
        activityRepository.deleteAll(toDelete);
    } else {
        throw new UnauthorizedException("You are not the owner of this activity.");
    }
}

public List<Activity> toDelete(NetId netId) {
    List<Activity> toDelete = new ArrayList<>();
    for(Activity activity : activityRepository.findAll()) {
        if (activity.getOwner().getNetIdValue().equals(netId.getNetIdValue())) {
            toDelete.add(activity);
        }
    }
    return toDelete;
}
```

Metrics After



Method: deleteByUser(NetId, NetId)

Metric	Metrics Set	Description	Value	Regular Ra...
CND		Condition Nesting Depth	2	[0..2)
LND		Loop Nesting Depth	0	[0..2)
CC		McCabe Cyclomatic Complexity	3	[0..3)
NOL		Number Of Loops	0	
LOC		Lines Of Code	16	[0..11)
NOPM		Number Of Parameters	2	[0..3)
HVL	Halstead M...	Halstead Volume	100.0782	
HD	Halstead M...	Halstead Difficulty	8.0	
HL	Halstead M...	Halstead Length	24	
HEF	Halstead M...	Halstead Effort	800.6256	
HVC	Halstead M...	Halstead Vocabulary	18	
HER	Halstead M...	Halstead Errors	0.0287	
MMI	Maintainab...	Maintainability Index	59.5816	[0.0..19.0]

These are the metric statistics before extracting the method:

- Cyclomatic Complexity = 5
- Lines of Code = 22

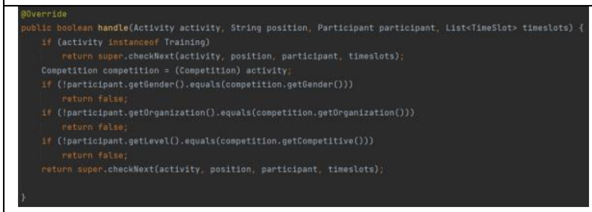
Method **deleteByUser** had a for loop that filters the Activity by user NetId. We created a new method **toDelete** and copied the filtering part. The extracted method has a parameter netId that needs to be filtered out according to and it returns the list of filtered activites. The source method calls **toDelete** whichs gets the returned list of activities and throws an exception if the list is empty. If it’s not empty, the list of activities are deleted from the repository.

Improved:

- Cyclomatic Complexity = 3
- Lines of Code = 16

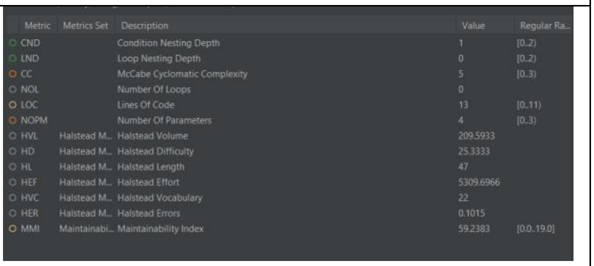
4. matcher-microservice/src/main/java/nl/tudelft/sem/template/example/domain/chainOfResponsability/CompetitionValidator.java - **handle**

Code Before



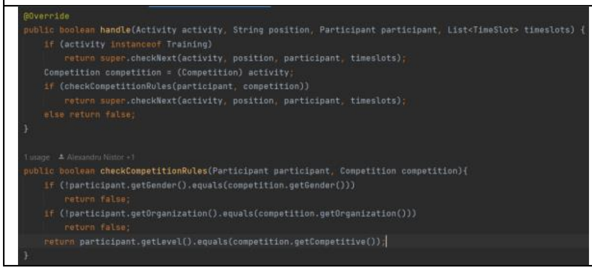
```
@Override
public boolean handle(Activity activity, String position, Participant participant, List<TimeSlot> timeslots) {
    if (activity instanceof Training)
        return super.checkNext(activity, position, participant, timeslots);
    Competition competition = (Competition) activity;
    if (!participant.getGender().equals(competition.getGender()))
        return false;
    if (!participant.getOrganization().equals(competition.getOrganization()))
        return false;
    if (!participant.getLevel().equals(competition.getCompetitive()))
        return false;
    return super.checkNext(activity, position, participant, timeslots);

}
```

Metrics Before



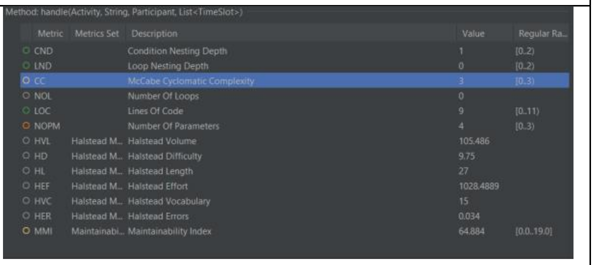
Metric	Metrics Set	Description	Value	Regular Ra...
○ CND		Condition Nesting Depth	1	[0..2)
○ LND		Loop Nesting Depth	0	[0..2)
○ CC		McCabe Cyclomatic Complexity	5	[0..3)
○ NOL		Number Of Loops	0	
○ LOC		Lines Of Code	13	[0..11)
○ NOPM		Number Of Parameters	4	[0..3)
○ HVL	Halstead M...	Halstead Volume	209.5933	
○ HD	Halstead M...	Halstead Difficulty	25.3333	
○ HL	Halstead M...	Halstead Length	47	
○ HEF	Halstead M...	Halstead Effort	5309.6966	
○ HVC	Halstead M...	Halstead Vocabulary	22	
○ HER	Halstead M...	Halstead Errors	0.1015	
○ MMI	Maintainabi...	Maintainability Index	59.2383	[0.0..19.0]

Code After



```
@Override
public boolean handle(Activity activity, String position, Participant participant, List<TimeSlot> timeslots) {
    if (activity instanceof Training)
        return super.checkNext(activity, position, participant, timeslots);
    Competition competition = (Competition) activity;
    if (checkCompetitionRules(participant, competition))
        return super.checkNext(activity, position, participant, timeslots);
    else return false;
}

1 usage  Alexandru Nistor +1
public boolean checkCompetitionRules(Participant participant, Competition competition){
    if (!participant.getGender().equals(competition.getGender()))
        return false;
    if (!participant.getOrganization().equals(competition.getOrganization()))
        return false;
    return participant.getLevel().equals(competition.getCompetitive());
}
```

Metrics After



Method: handle(Activity, String, Participant, List<TimeSlot>)

Metric	Metrics Set	Description	Value	Regular Ra...
○ CND		Condition Nesting Depth	1	[0..2)
○ LND		Loop Nesting Depth	0	[0..2)
○ CC		McCabe Cyclomatic Complexity	3	[0..3)
○ NOL		Number Of Loops	0	
○ LOC		Lines Of Code	9	[0..11)
○ NOPM		Number Of Parameters	4	[0..3)
○ HVL	Halstead M...	Halstead Volume	105.486	
○ HD	Halstead M...	Halstead Difficulty	9.75	
○ HL	Halstead M...	Halstead Length	27	
○ HEF	Halstead M...	Halstead Effort	1028.4889	
○ HVC	Halstead M...	Halstead Vocabulary	15	
○ HER	Halstead M...	Halstead Errors	0.034	
○ MMI	Maintainabi...	Maintainability Index	64.884	[0.0..19.0]

These are the metric statistics before extracting the method:

- Cyclomatic Complexity = 5
- Lines of Code = 13

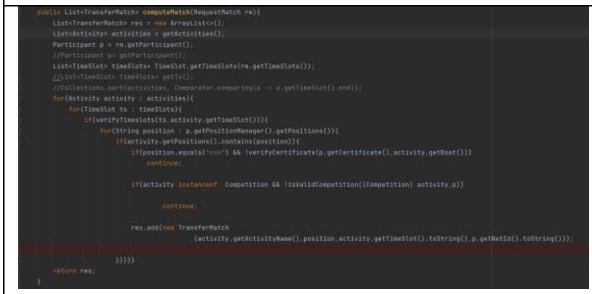
We extracted the method **checkCompetitionRules** that has parameter of participant and competition that checks if the participant meets the requirement of the given competition. In the source method, if an activity is an a competition, it calls **checkCompetitionRules**. If the condition is met, it forwards to the next validator.

Improved

- Cyclomatic Complexity = 3
- Lines of Code = 9

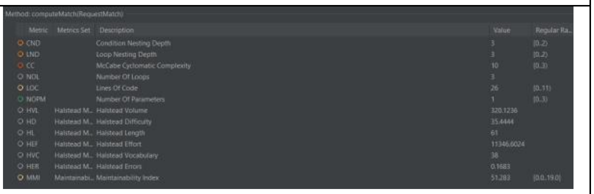
5. matcher-microservice/src/main/java/nl/tudelft/sem/template/example/domain/MatcherService.java - **computeMatch**

Code Before



```java
public List<TransferMatch> computeMatch(RequestMatch rm){
    List<TransferMatch> res = new ArrayList<>();
    List<Activity> activities = getActivities();
    Participant p = rm.getParticipant();
    //Participant p= getParticipant();
    List<TimeSlot> timeSlots= TimeSlot.getTimeSlots(rm.getTimeSlots());
    //List<TimeSlot> timeSlots= getTs();
    //Collections.sort(activities, Comparator.comparing(a -> a.getTimeSlot().end));
    for(Activity activity : activities){
        for(TimeSlot ts : timeSlots){
            if(verifyTimeslots(ts,activity.getTimeSlot())){
                for(String position : p.getPositionManager().getPositions()){
                    if(activity.getPositions().contains(position)){
                        if(position.equals("cox") && !verifyCertificate(p.getCertificate(),activity.getBoat()))
                            continue;

                        if(activity instanceof  Competition && !isValidCompetition((Competition) activity,p))

                                continue;

                        res.add(new TransferMatch
                                (activity.getActivityName(),position,activity.getTimeSlot().toString(),p.getNetId().toString()));

                    }}}}}
    return res;
}
```

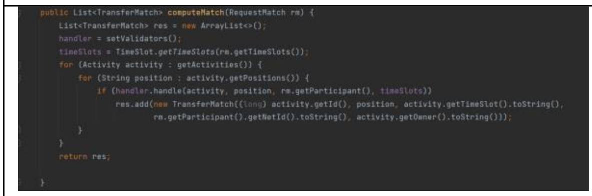
Metrics Before



Method: computeMatch(RequestMatch)

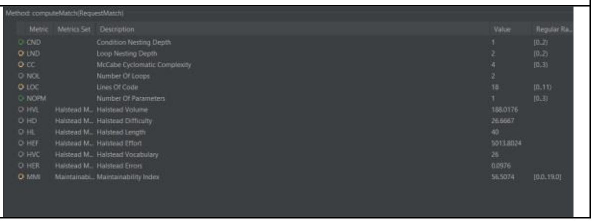
Metric	Metrics Set	Description	Value	Regular Ra...
○ CND		Condition Nesting Depth	3	[0..2)
○ LND		Loop Nesting Depth	3	[0..2)
○ CC		McCabe Cyclomatic Complexity	10	[0..3)
○ NOL		Number Of Loops	3	
○ LOC		Lines Of Code	26	[0..11)
○ NOPM		Number Of Parameters	1	[0..3)
○ HVL	Halstead M...	Halstead Volume	320.1236	
○ HD	Halstead M...	Halstead Difficulty	35.4444	
○ HL	Halstead M...	Halstead Length	61	
○ HEF	Halstead M...	Halstead Effort	11346.6024	
○ HVC	Halstead M...	Halstead Vocabulary	38	
○ HER	Halstead M...	Halstead Errors	0.1683	
○ MMI	Maintainabi...	Maintainability Index	51.283	[0.0..19.0]

Code After



```java
public List<TransferMatch> computeMatch(RequestMatch rm) {
    List<TransferMatch> res = new ArrayList<>();
    handler = setValidators();
    timeSlots = TimeSlot.getTimeSlots(rm.getTimeSlots());
    for (Activity activity : getActivities()) {
        for (String position : activity.getPositions()) {
            if (handler.handle(activity, position, rm.getParticipant(), timeSlots))
                res.add(new TransferMatch((long) activity.getId(), position, activity.getTimeSlot().toString(),
                        rm.getParticipant().getNetId().toString(), activity.getOwner().toString()));
        }
    }
    return res;

}
```

Metrics After



Method: computeMatch(RequestMatch)

Metric	Metrics Set	Description	Value	Regular Ra...
○ CND		Condition Nesting Depth	1	[0..2)
○ LND		Loop Nesting Depth	2	[0..2)
○ CC		McCabe Cyclomatic Complexity	4	[0..3)
○ NOL		Number Of Loops	2	
○ LOC		Lines Of Code	18	[0..11)
○ NOPM		Number Of Parameters	1	[0..3)
○ HVL	Halstead M...	Halstead Volume	188.0176	
○ HD	Halstead M...	Halstead Difficulty	26.6667	
○ HL	Halstead M...	Halstead Length	40	
○ HEF	Halstead M...	Halstead Effort	5013.8024	
○ HVC	Halstead M...	Halstead Vocabulary	26	
○ HER	Halstead M...	Halstead Errors	0.0976	
○ MMI	Maintainabi...	Maintainability Index	56.5074	[0.0..19.0]

These are the metric statistics before extracting the method:

- Cyclomatic Complexity = 10
- Lines of Code = 26
- Condition Nesting Depth = 3

Method **computeMatch** filtered the Activities, Timeslots and Positions using 3 for loops and 4 if statements, 2 of which had a double condition. The former cyclomatic complexity was 10 and we decided that this can be reduced by refactoring using the Chain of Responsibility design pattern. Hence, the dataflow will go through the following chain: TimeValidator, PositionValidator, CompetitionValidator and CertificateValidator. This resulted in having a complexity of 4, using only 2 for loops and 1 if statement. The procedure also improved some other software metrics such as **Lines of code** and **Condition Nesting Depth**.

These are the metric statistics after refactoring:

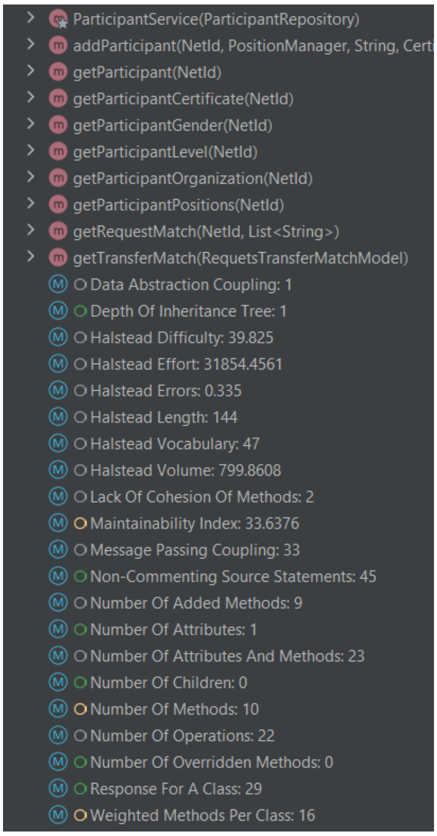
- Cyclomatic Complexity = 4
- Lines of Code = 18
- Condition Nesting Depth = 1

Class-level refactoring

Our classes main problems were the **Number of Methods** and the **Weighted Methods Per Class.** To improve both metrics we used **Extract Class Refactoring**. For each class we wanted to improve, we thought about how to divide the responsibilities of the class to make two or more classes out of it.

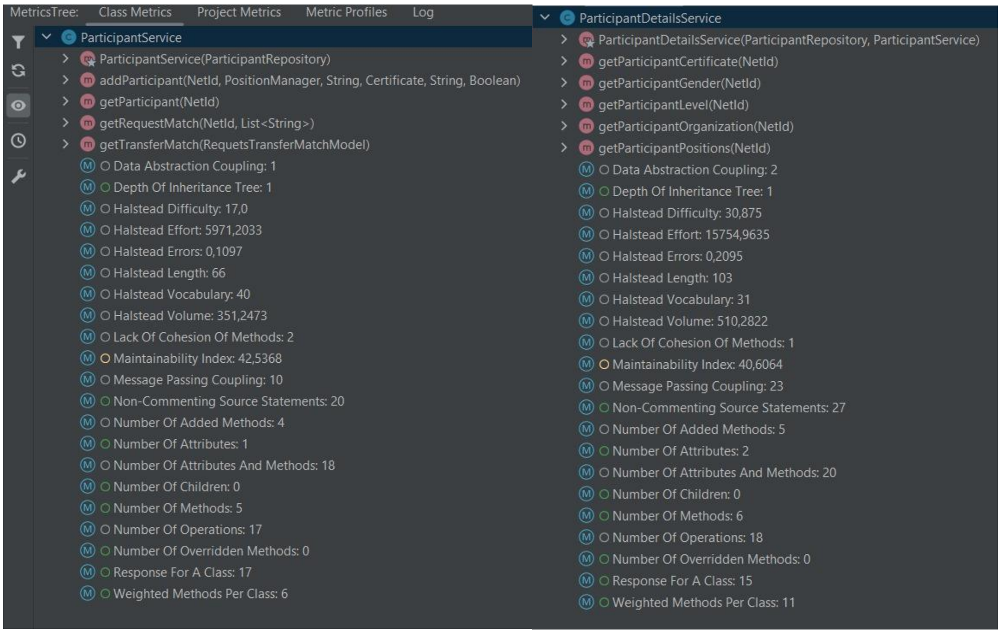
1. ParticipantService class

Output of the metrics tool for the ParticipantService class before the refactoring:



```
ParticipantService(ParticipantRepository)
addParticipant(NetId, PositionManager, String, Cert
getParticipant(NetId)
getParticipantCertificate(NetId)
getParticipantGender(NetId)
getParticipantLevel(NetId)
getParticipantOrganization(NetId)
getParticipantPositions(NetId)
getRequestMatch(NetId, List<String>)
getTransferMatch(RequetsTransferMatchModel)
Data Abstraction Coupling: 1
Depth Of Inheritance Tree: 1
Halstead Difficulty: 39.825
Halstead Effort: 31854.4561
Halstead Errors: 0.335
Halstead Length: 144
Halstead Vocabulary: 47
Halstead Volume: 799.8608
Lack Of Cohesion Of Methods: 2
Maintainability Index: 33.6376
Message Passing Coupling: 33
Non-Commenting Source Statements: 45
Number Of Added Methods: 9
Number Of Attributes: 1
Number Of Attributes And Methods: 23
Number Of Children: 0
Number Of Methods: 10
Number Of Operations: 22
Number Of Overridden Methods: 0
Response For A Class: 29
Weighted Methods Per Class: 16
```

Output of the metrics tool for the ParticipantService and ParticipantDetailsService classes after the refactoring:



```
ParticipantService
  > ParticipantService(ParticipantRepository)
  > addParticipant(NetId, PositionManager, String, Certificate, String, Boolean)
  > getParticipant(NetId)
  > getRequestMatch(NetId, List<String>)
  > getTransferMatch(RequetsTransferMatchModel)
    Data Abstraction Coupling: 1
    Depth Of Inheritance Tree: 1
    Halstead Difficulty: 17,0
    Halstead Effort: 5971,2033
    Halstead Errors: 0,1097
    Halstead Length: 66
    Halstead Vocabulary: 40
    Halstead Volume: 351,2473
    Lack Of Cohesion Of Methods: 2
    Maintainability Index: 42,5368
    Message Passing Coupling: 10
    Non-Commenting Source Statements: 20
    Number Of Added Methods: 4
    Number Of Attributes: 1
    Number Of Attributes And Methods: 18
    Number Of Children: 0
    Number Of Methods: 5
    Number Of Operations: 17
    Number Of Overridden Methods: 0
    Response For A Class: 17
    Weighted Methods Per Class: 6
```

```
ParticipantDetailsService
  > ParticipantDetailsService(ParticipantRepository, ParticipantService)
  > getParticipantCertificate(NetId)
  > getParticipantGender(NetId)
  > getParticipantLevel(NetId)
  > getParticipantOrganization(NetId)
  > getParticipantPositions(NetId)
    Data Abstraction Coupling: 2
    Depth Of Inheritance Tree: 1
    Halstead Difficulty: 30,875
    Halstead Effort: 15754,9635
    Halstead Errors: 0,2095
    Halstead Length: 103
    Halstead Vocabulary: 31
    Halstead Volume: 510,2822
    Lack Of Cohesion Of Methods: 1
    Maintainability Index: 40,6064
    Message Passing Coupling: 23
    Non-Commenting Source Statements: 27
    Number Of Added Methods: 5
    Number Of Attributes: 2
    Number Of Attributes And Methods: 20
    Number Of Children: 0
    Number Of Methods: 6
    Number Of Operations: 18
    Number Of Overridden Methods: 0
    Response For A Class: 15
    Weighted Methods Per Class: 11
```

Before the refactoring, the metric statistics were the following:

- Number of Methods = 10
- Weighted Methods Per Class = 16

To decrease the Number of Methods (10) and Weighted Methods per Class (16) for the ParticipantService class, it was split into two classes: ParticipantService and ParticipantDetailsService. Participant service manages the addition of the participant in the database, getter of the participant and also getters for transfer objects(Transfer Match and RequestMatch). In the ParticipantDetailsService, there are getters for level, gender, certificate, organization and positions.

ParticipantService:

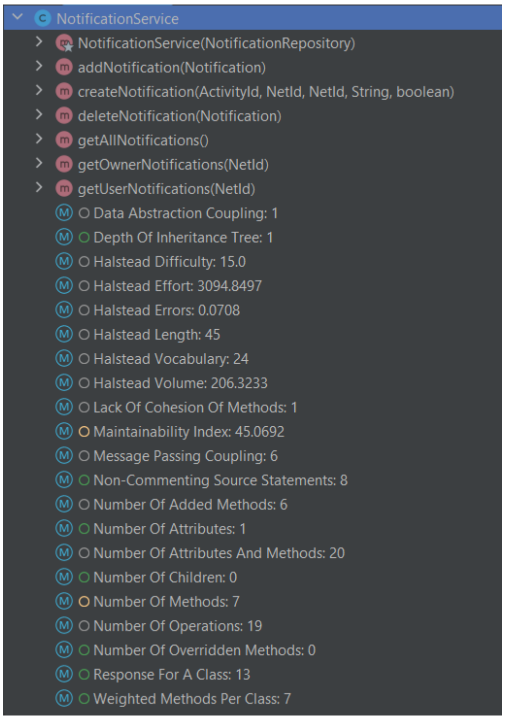
- Number of Methods = 5
- Weighted Methods Per Class = 6

ParticipantDetailService

- Number of Methods = 6
- Weighted Methods Per Class = 11

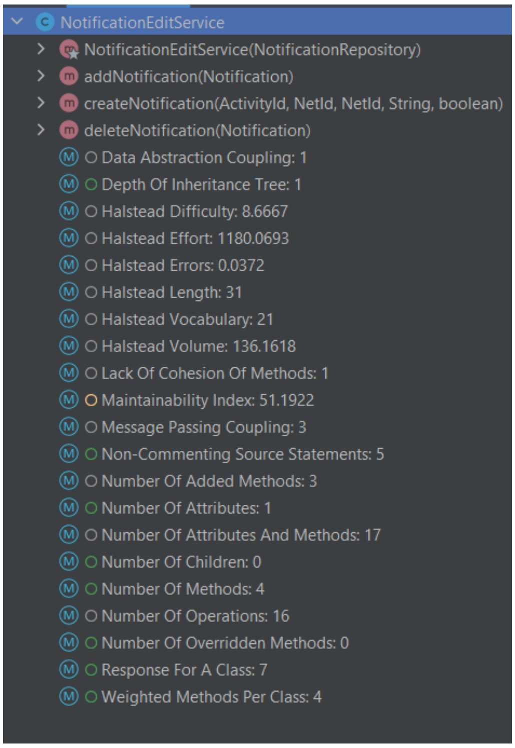
2. NotificationService class

Output of the metrics tool for the NotificationService class before the refactoring:

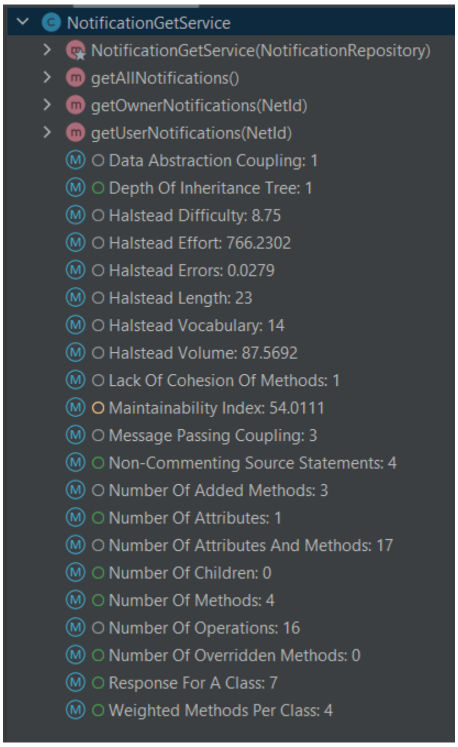


```
NotificationService
  > NotificationService(NotificationRepository)
  > addNotification(Notification)
  > createNotification(ActivityId, NetId, NetId, String, boolean)
  > deleteNotification(Notification)
  > getAllNotifications()
  > getOwnerNotifications(NetId)
  > getUserNotifications(NetId)
    Data Abstraction Coupling: 1
    Depth Of Inheritance Tree: 1
    Halstead Difficulty: 15.0
    Halstead Effort: 3094.8497
    Halstead Errors: 0.0708
    Halstead Length: 45
    Halstead Vocabulary: 24
    Halstead Volume: 206.3233
    Lack Of Cohesion Of Methods: 1
    Maintainability Index: 45.0692
    Message Passing Coupling: 6
    Non-Commenting Source Statements: 8
    Number Of Added Methods: 6
    Number Of Attributes: 1
    Number Of Attributes And Methods: 20
    Number Of Children: 0
    Number Of Methods: 7
    Number Of Operations: 19
    Number Of Overridden Methods: 0
    Response For A Class: 13
    Weighted Methods Per Class: 7
```

Output of the metrics tool for the split classes NotificationEditService and NotificationGetService after the refactoring:



```
NotificationEditService
  > NotificationEditService(NotificationRepository)
  > addNotification(Notification)
  > createNotification(ActivityId, NetId, NetId, String, boolean)
  > deleteNotification(Notification)
    Data Abstraction Coupling: 1
    Depth Of Inheritance Tree: 1
    Halstead Difficulty: 8.6667
    Halstead Effort: 1180.0693
    Halstead Errors: 0.0372
    Halstead Length: 31
    Halstead Vocabulary: 21
    Halstead Volume: 136.1618
    Lack Of Cohesion Of Methods: 1
    Maintainability Index: 51.1922
    Message Passing Coupling: 3
    Non-Commenting Source Statements: 5
    Number Of Added Methods: 3
    Number Of Attributes: 1
    Number Of Attributes And Methods: 17
    Number Of Children: 0
    Number Of Methods: 4
    Number Of Operations: 16
    Number Of Overridden Methods: 0
    Response For A Class: 7
    Weighted Methods Per Class: 4
```



```
NotificationGetService
  > NotificationGetService(NotificationRepository)
  > getAllNotifications()
  > getOwnerNotifications(NetId)
  > getUserNotifications(NetId)
    Data Abstraction Coupling: 1
    Depth Of Inheritance Tree: 1
    Halstead Difficulty: 8.75
    Halstead Effort: 766.2302
    Halstead Errors: 0.0279
    Halstead Length: 23
    Halstead Vocabulary: 14
    Halstead Volume: 87.5692
    Lack Of Cohesion Of Methods: 1
    Maintainability Index: 54.0111
    Message Passing Coupling: 3
    Non-Commenting Source Statements: 4
    Number Of Added Methods: 3
    Number Of Attributes: 1
    Number Of Attributes And Methods: 17
    Number Of Children: 0
    Number Of Methods: 4
    Number Of Operations: 16
    Number Of Overridden Methods: 0
    Response For A Class: 7
    Weighted Methods Per Class: 4
```

Before the refactoring, the metric statistics were the following:

- Number of Methods = 7

- Weighted Methods Per Class = 7

To decrease the Number of Methods (7) NotificationService class, it was split into two classes: NotificationEditService and NotificationGetService. NotificationEditService manages editing of notifications and NotificationGetService is responsible for the retrieval of notifications

NotificationEditService:

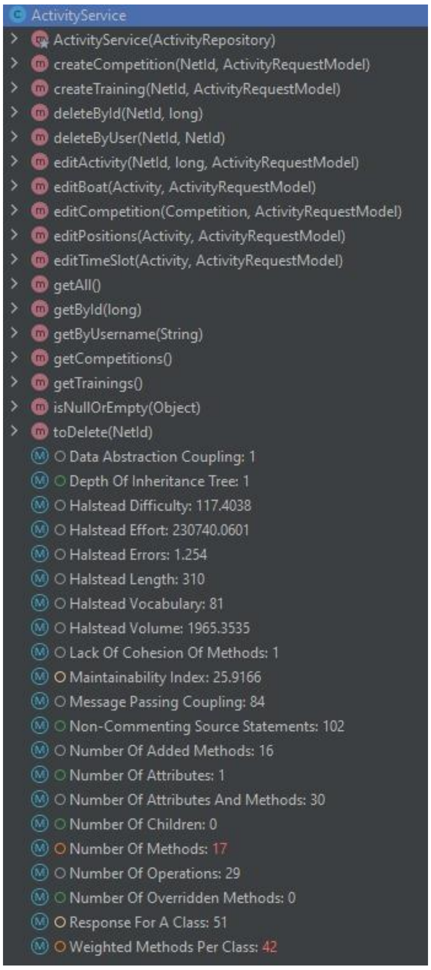
- Number of Methods = 4
- Weighted Methods Per Class = 4

NotificationGetService:

- Number of Methods = 4
- Weighted Methods Per Class = 4

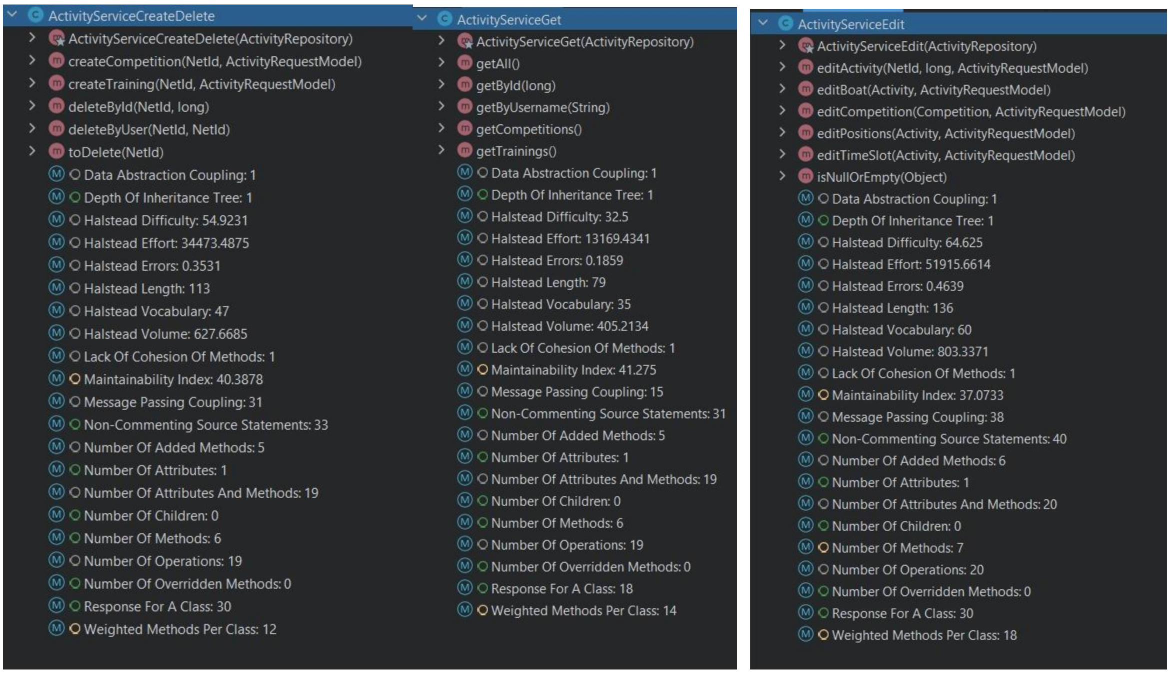
3. ActivityService class

Output of the metrics tool for the ActivityService class before the refactoring:



```
ActivityService
>  ActivityService(ActivityRepository)
>  createCompetition(NetId, ActivityRequestModel)
>  createTraining(NetId, ActivityRequestModel)
>  deleteById(NetId, long)
>  deleteByUser(NetId, NetId)
>  editActivity(NetId, long, ActivityRequestModel)
>  editBoat(Activity, ActivityRequestModel)
>  editCompetition(Competition, ActivityRequestModel)
>  editPositions(Activity, ActivityRequestModel)
>  editTimeSlot(Activity, ActivityRequestModel)
>  getAll()
>  getById(long)
>  getByUsername(String)
>  getCompetitions()
>  getTrainings()
>  isNullOrEmpty(Object)
>  toDelete(NetId)
   Data Abstraction Coupling: 1
   Depth Of Inheritance Tree: 1
   Halstead Difficulty: 117.4038
   Halstead Effort: 230740.0601
   Halstead Errors: 1.254
   Halstead Length: 310
   Halstead Vocabulary: 81
   Halstead Volume: 1965.3535
   Lack Of Cohesion Of Methods: 1
   Maintainability Index: 25.9166
   Message Passing Coupling: 84
   Non-Commenting Source Statements: 102
   Number Of Added Methods: 16
   Number Of Attributes: 1
   Number Of Attributes And Methods: 30
   Number Of Children: 0
   Number Of Methods: 17
   Number Of Operations: 29
   Number Of Overridden Methods: 0
   Response For A Class: 51
   Weighted Methods Per Class: 42
```

Output of the metrics tool for the split classes ActivityServiceCreateDelete, AcitivityServiceGet and ActivityServiceEdit after the refactoring:



```
ActivityServiceCreateDelete
  ActivityServiceCreateDelete(ActivityRepository)
  createCompetition(NetId, ActivityRequestModel)
  createTraining(NetId, ActivityRequestModel)
  deleteById(NetId, long)
  deleteByUser(NetId, NetId)
  toDelete(NetId)
  Data Abstraction Coupling: 1
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 54.9231
  Halstead Effort: 34473.4875
  Halstead Errors: 0.3531
  Halstead Length: 113
  Halstead Vocabulary: 47
  Halstead Volume: 627.6685
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 40.3878
  Message Passing Coupling: 31
  Non-Commenting Source Statements: 33
  Number Of Added Methods: 5
  Number Of Attributes: 1
  Number Of Attributes And Methods: 19
  Number Of Children: 0
  Number Of Methods: 6
  Number Of Operations: 19
  Number Of Overridden Methods: 0
  Response For A Class: 30
  Weighted Methods Per Class: 12
```

```
ActivityServiceGet
  ActivityServiceGet(ActivityRepository)
  getAll()
  getById(long)
  getByUsername(String)
  getCompetitions()
  getTrainings()
  Data Abstraction Coupling: 1
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 32.5
  Halstead Effort: 13169.4341
  Halstead Errors: 0.1859
  Halstead Length: 79
  Halstead Vocabulary: 35
  Halstead Volume: 405.2134
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 41.275
  Message Passing Coupling: 15
  Non-Commenting Source Statements: 31
  Number Of Added Methods: 5
  Number Of Attributes: 1
  Number Of Attributes And Methods: 19
  Number Of Children: 0
  Number Of Methods: 6
  Number Of Operations: 19
  Number Of Overridden Methods: 0
  Response For A Class: 18
  Weighted Methods Per Class: 14
```

```
ActivityServiceEdit
  ActivityServiceEdit(ActivityRepository)
  editActivity(NetId, long, ActivityRequestModel)
  editBoat(Activity, ActivityRequestModel)
  editCompetition(Competition, ActivityRequestModel)
  editPositions(Activity, ActivityRequestModel)
  editTimeSlot(Activity, ActivityRequestModel)
  isNullOrEmpty(Object)
  Data Abstraction Coupling: 1
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 64.625
  Halstead Effort: 51915.6614
  Halstead Errors: 0.4639
  Halstead Length: 136
  Halstead Vocabulary: 60
  Halstead Volume: 803.3371
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 37.0733
  Message Passing Coupling: 38
  Non-Commenting Source Statements: 40
  Number Of Added Methods: 6
  Number Of Attributes: 1
  Number Of Attributes And Methods: 20
  Number Of Children: 0
  Number Of Methods: 7
  Number Of Operations: 20
  Number Of Overridden Methods: 0
  Response For A Class: 30
  Weighted Methods Per Class: 18
```

Before the refactoring, the metric statistics were the following:

- Number of Methods = 17
- Weighted Methods Per Class = 42

To decrease the Number of Methods (17) and Weighted Methods per Class (42) for the ActivityService class, it was split into three classes: ActivityServiceCreateDelete, AcitivityServiceGet and ActivityServiceEdit. ActivityServiceCreateDelete manages all the creation and deletion of activities, ActivityServiceEdit includes endpoints for editing activities and ActivityServiceGet is responsible for retrival of Activities.

ActivityServiceCreateDelete:

- Number of Methods = 6
- Weighted Methods Per Class = 12

ActivityServiceGet:

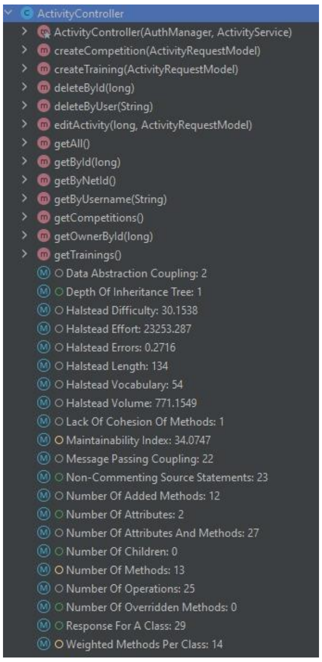
- Number of Methods = 6
- Weighted Methods Per Class = 14

ActivityServiceGet:

- Number of Methods = 7
- Weighted Methods Per Class = 18

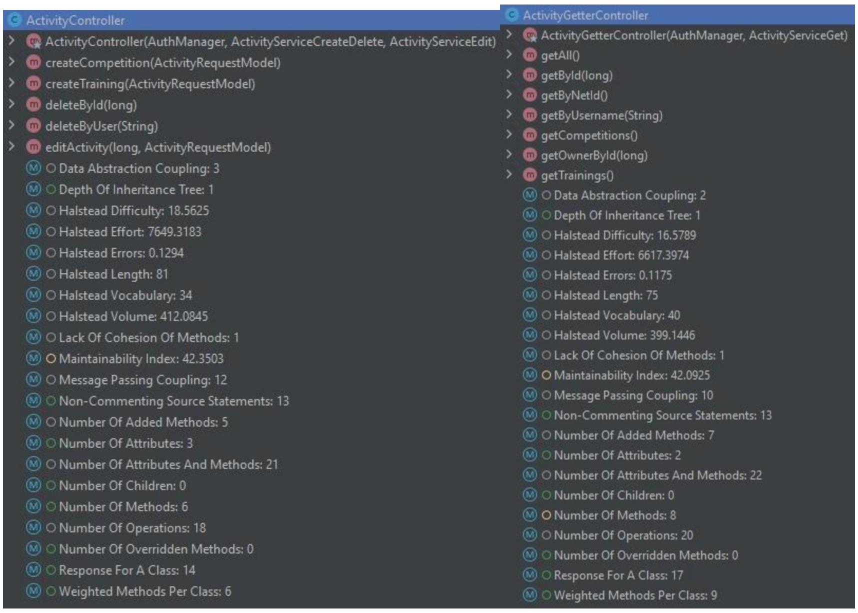
4. ActivityController class

Output of the metrics tool for the ActivityControllerService class before the refactoring:



```
ActivityController
  ActivityController(AuthManager, ActivityService)
  createCompetition(ActivityRequestModel)
  createTraining(ActivityRequestModel)
  deleteById(long)
  deleteByUser(String)
  editActivity(long, ActivityRequestModel)
  getAll()
  getById(long)
  getByNetId()
  getByUsername(String)
  getCompetitions()
  getOwnerById(long)
  getTrainings()
  Data Abstraction Coupling: 2
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 30.1538
  Halstead Effort: 23253.287
  Halstead Errors: 0.2716
  Halstead Length: 134
  Halstead Vocabulary: 54
  Halstead Volume: 771.1549
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 34.0747
  Message Passing Coupling: 22
  Non-Commenting Source Statements: 23
  Number Of Added Methods: 12
  Number Of Attributes: 2
  Number Of Attributes And Methods: 27
  Number Of Children: 0
  Number Of Methods: 13
  Number Of Operations: 25
  Number Of Overridden Methods: 0
  Response For A Class: 29
  Weighted Methods Per Class: 14
```

Output of the metrics tool for the split classes ActivityGetterController and ActivityController after the refactoring:



```
ActivityController
  ActivityController(AuthManager, ActivityServiceCreateDelete, ActivityServiceEdit)
  createCompetition(ActivityRequestModel)
  createTraining(ActivityRequestModel)
  deleteById(long)
  deleteByUser(String)
  editActivity(long, ActivityRequestModel)
  Data Abstraction Coupling: 3
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 18.5625
  Halstead Effort: 7649.3183
  Halstead Errors: 0.1294
  Halstead Length: 81
  Halstead Vocabulary: 34
  Halstead Volume: 412.0845
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 42.3503
  Message Passing Coupling: 12
  Non-Commenting Source Statements: 13
  Number Of Added Methods: 5
  Number Of Attributes: 3
  Number Of Attributes And Methods: 21
  Number Of Children: 0
  Number Of Methods: 6
  Number Of Operations: 18
  Number Of Overridden Methods: 0
  Response For A Class: 14
  Weighted Methods Per Class: 6
```

```
ActivityGetterController
  ActivityGetterController(AuthManager, ActivityServiceGet)
  getAll()
  getById(long)
  getByNetId()
  getByUsername(String)
  getCompetitions()
  getOwnerById(long)
  getTrainings()
  Data Abstraction Coupling: 2
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 16.5789
  Halstead Effort: 6617.3974
  Halstead Errors: 0.1175
  Halstead Length: 75
  Halstead Vocabulary: 40
  Halstead Volume: 399.1446
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 42.0925
  Message Passing Coupling: 10
  Non-Commenting Source Statements: 13
  Number Of Added Methods: 7
  Number Of Attributes: 2
  Number Of Attributes And Methods: 22
  Number Of Children: 0
  Number Of Methods: 8
  Number Of Operations: 20
  Number Of Overridden Methods: 0
  Response For A Class: 17
  Weighted Methods Per Class: 9
```

Before refactoring, the metric statistics were the following :

- Number of Methods = 13
- Weighted Methods Per Class = 14

To decrease the Number of Methods (13) and Weighted Methods per Class (14) for the ActivityControllerService class, it was split into two classes: ActivityController and ActivityGetterController based on their responsibilities. ActivityController manages all the creation, deletion, and editing of activities. Extracted class ActivityGetterController includes endpoints that retrieves the desired Activities.

ActivityController :

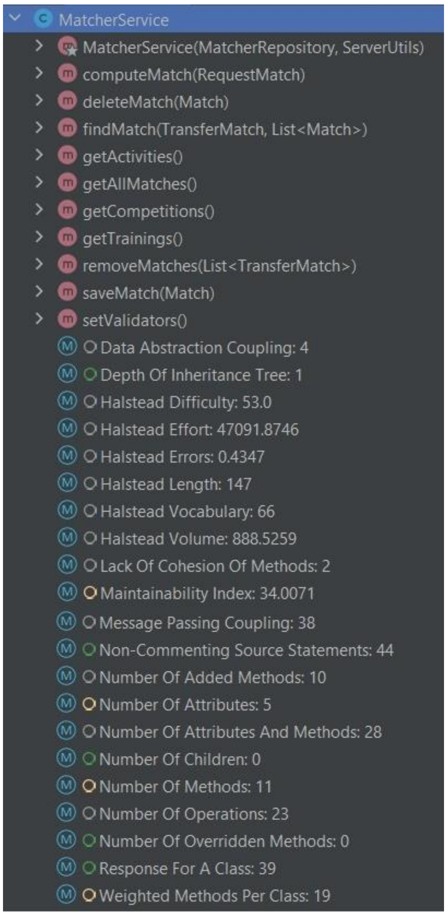
- Number of Methods = 6
- Weighted Methods Per Class = 6

ActivityGetterController :

- Number of Methods = 8
- Weighted Methods Per Class = 9

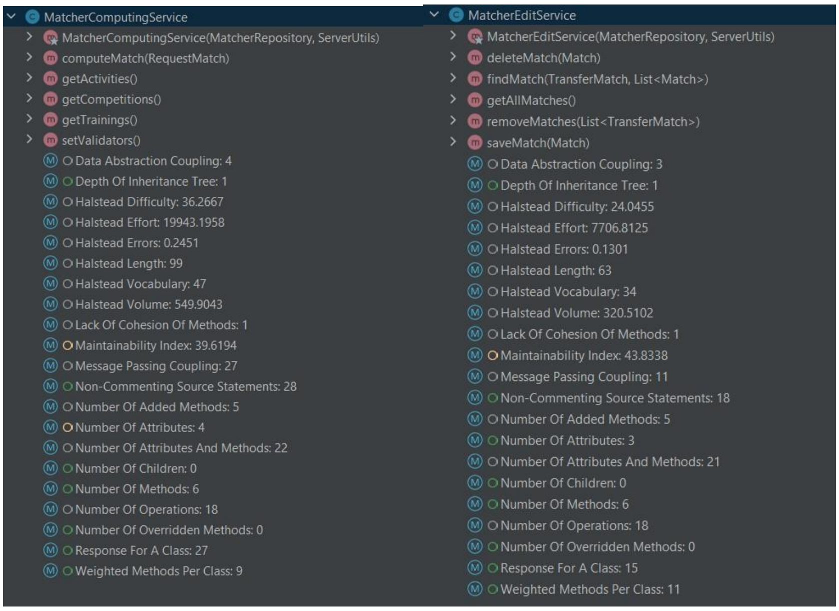
5. MatcherService class

Output of the metrics tool for the MatcherService class before the refactoring:



```
MatcherService
  MatcherService(MatcherRepository, ServerUtils)
  computeMatch(RequestMatch)
  deleteMatch(Match)
  findMatch(TransferMatch, List<Match>)
  getActivities()
  getAllMatches()
  getCompetitions()
  getTrainings()
  removeMatches(List<TransferMatch>)
  saveMatch(Match)
  setValidators()
  Data Abstraction Coupling: 4
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 53.0
  Halstead Effort: 47091.8746
  Halstead Errors: 0.4347
  Halstead Length: 147
  Halstead Vocabulary: 66
  Halstead Volume: 888.5259
  Lack Of Cohesion Of Methods: 2
  Maintainability Index: 34.0071
  Message Passing Coupling: 38
  Non-Commenting Source Statements: 44
  Number Of Added Methods: 10
  Number Of Attributes: 5
  Number Of Attributes And Methods: 28
  Number Of Children: 0
  Number Of Methods: 11
  Number Of Operations: 23
  Number Of Overridden Methods: 0
  Response For A Class: 39
  Weighted Methods Per Class: 19
```

Output of the metrics tool for the split classes MatcherComputingService and MatcherEditService after the refactoring:



```
MatcherComputingService
  MatcherComputingService(MatcherRepository, ServerUtils)
  computeMatch(RequestMatch)
  getActivities()
  getCompetitions()
  getTrainings()
  setValidators()
  Data Abstraction Coupling: 4
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 36.2667
  Halstead Effort: 19943.1958
  Halstead Errors: 0.2451
  Halstead Length: 99
  Halstead Vocabulary: 47
  Halstead Volume: 549.9043
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 39.6194
  Message Passing Coupling: 27
  Non-Commenting Source Statements: 28
  Number Of Added Methods: 5
  Number Of Attributes: 4
  Number Of Attributes And Methods: 22
  Number Of Children: 0
  Number Of Methods: 6
  Number Of Operations: 18
  Number Of Overridden Methods: 0
  Response For A Class: 27
  Weighted Methods Per Class: 9
```

```
MatcherEditService
  MatcherEditService(MatcherRepository, ServerUtils)
  deleteMatch(Match)
  findMatch(TransferMatch, List<Match>)
  getAllMatches()
  removeMatches(List<TransferMatch>)
  saveMatch(Match)
  Data Abstraction Coupling: 3
  Depth Of Inheritance Tree: 1
  Halstead Difficulty: 24.0455
  Halstead Effort: 7706.8125
  Halstead Errors: 0.1301
  Halstead Length: 63
  Halstead Vocabulary: 34
  Halstead Volume: 320.5102
  Lack Of Cohesion Of Methods: 1
  Maintainability Index: 43.8338
  Message Passing Coupling: 11
  Non-Commenting Source Statements: 18
  Number Of Added Methods: 5
  Number Of Attributes: 3
  Number Of Attributes And Methods: 21
  Number Of Children: 0
  Number Of Methods: 6
  Number Of Operations: 18
  Number Of Overridden Methods: 0
  Response For A Class: 15
  Weighted Methods Per Class: 11
```

Before refactoring, the metric statistics were the following :

- Number of Methods = 11
- Weighted Methods Per Class = 19

To decrease the Number of Methods and Weighted Methods per Class for the MatcherService class, it was split into two classes: MatcherComputingService and MatcherEditService. The former class contains the core functionality for computing a match and the latter wraps methods corcerning operations on the MatcherDatabase .

MatcherComputingService :

- Number of Methods = 6
- Weighted Methods Per Class = 9

MatcherEditService :

- Number of Methods = 6
- Weighted Methods Per Class = 11