Assignment 3 – Mutation testing
Task 1

1. ActivityController class in the Activity microservice

The Activity microservice has only two classes that could be chosen for this assignment and the first one of them is the **ActivityController** class which had the mutation test **coverage** of **81%** as seen in the picture below.

ActivityController.java

Mutations

66 1. negated conditional → KILLED
67 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::getActivityTimeslotById → NO_COVERAGE
69 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::getActivityTimeslotById → KILLED
82 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::getAllActivitiesWithinTimeslot → KILLED
93 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::getAllActivitiesByOwner → KILLED
103 1. negated conditional → KILLED
104 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::createActivity → KILLED
106 1. removed call to nl/tudelft/sem/template/activities/domain/Activity::setOwnerId → SURVIVED
108 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::createActivity → KILLED
119 1. negated conditional → KILLED
120 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::deleteActivity → KILLED
123 1. negated conditional → KILLED
124 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::deleteActivity → KILLED
126 1. negated conditional → KILLED
127 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::deleteActivity → KILLED
129 1. removed call to nl/tudelft/sem/template/activities/domain/ActivityRepository::delete → KILLED
130 1. removed call to nl/tudelft/sem/template/activities/domain/MatchingClient::deleteAllMatches → KILLED
131 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::deleteActivity → KILLED
143 1. negated conditional → KILLED
144 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::updateActivity → SURVIVED
147 1. removed call to nl/tudelft/sem/template/activities/domain/Activity::updateFields → SURVIVED
148 1. negated conditional → KILLED
149 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::updateActivity → NO_COVERAGE
151 1. negated conditional → KILLED
152 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::updateActivity → NO_COVERAGE
155 1. removed call to nl/tudelft/sem/template/activities/domain/MatchingClient::deleteAllMatches → KILLED
156 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::updateActivity → KILLED
169 1. negated conditional → KILLED
170 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::reduceByOne → KILLED
173 1. negated conditional → KILLED
174 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::reduceByOne → KILLED
177 1. replaced return value with null for nl/tudelft/sem/template/activities/controllers/ActivityController::reduceByOne → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

The score after:

nl.tudelft.sem.template.activities.controllers

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
1	100%	49/49	100%	32/32	100%	32/32

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
ActivityController.java	100%	49/49	100%	32/32	100%	32/32

The commit that contains the updates:

https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM30b/-/commit/9adb2babc1c4aaa7c99c0620acbbab9c8058bcb6

2. Activity class in the Activity microservice

The other class that had a lot of room for improvement was the **Activity class**, with the **mutation score** measuring at **81%**. It was mostly this low because of a single method called **updateFields** whose behaviour was not properly verified by the testing suite.

Activity.java

Mutations

85	1. negated conditional → KILLED 2. negated conditional → KILLED
86	1. negated conditional → KILLED 2. negated conditional → KILLED 3. negated conditional → KILLED 4. replaced boolean return with true for nl/tudelft/sem/template/activities/domain/Activity::checkIfValid → KILLED
95	1. negated conditional → KILLED 2. negated conditional → KILLED 3. negated conditional → KILLED 4. negated conditional → KILLED 5. replaced boolean return with true for nl/tudelft/sem/template/activities/domain/Activity::checkIfPositionsAndTimeslotValid → KILLED
104	1. negated conditional → KILLED 2. negated conditional → KILLED 3. negated conditional → KILLED 4. negated conditional → KILLED 5. replaced boolean return with true for nl/tudelft/sem/template/activities/domain/Activity::checkIfCompetitionValid → KILLED
113	1. negated conditional → KILLED
114	1. removed call to nl/tudelft/sem/template/activities/domain/Activity::setPositions → KILLED
116	1. negated conditional → KILLED
117	1. removed call to nl/tudelft/sem/template/activities/domain/Activity::setTimeslot → SURVIVED
119	1. negated conditional → KILLED
120	1. removed call to nl/tudelft/sem/template/activities/domain/Activity::setCertificate → SURVIVED
122	1. negated conditional → KILLED
123	1. removed call to nl/tudelft/sem/template/activities/domain/Activity::setCompetition → SURVIVED
125	1. negated conditional → SURVIVED
126	1. removed call to nl/tudelft/sem/template/activities/domain/Activity::setGender → SURVIVED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

After:

nl.tudelft.sem.template.activities.domain

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
1	97%	35/36	100%	26/26	100%	26/26

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
Activity.java	97%	35/36	100%	26/26	100%	26/26

This is the commit that contains the updates:

https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM30b/-/commit/baac9ce09e4aca08284d48a412b896b4351f0edb

3. UserService class in User microservice

The **UserService class** lies at the very core of the **User microservice domain**, but had only achieved 76% mutation coverage (killing 13/17 mutants) with the original test suite. This was due to failure to test the connections between handlers in the chain of responsibility validation pattern.

Mutations

44	1. removed call to nl/tudelft/sem/template/users/domain/UserService::userValidationHandlerSetUp → SURVIVED
53	1. removed call to nl/tudelft/sem/template/users/domain/handlers/UserValidationHandler::setNext → SURVIVED
55	1. removed call to nl/tudelft/sem/template/users/domain/handlers/UserValidationHandler::setNext → SURVIVED
57	1. removed call to nl/tudelft/sem/template/users/domain/handlers/UserValidationHandler::setNext → SURVIVED
67	1. negated conditional → KILLED
70	1. replaced return value with null for nl/tudelft/sem/template/users/domain/UserService::getByEmail → KILLED
85	1. replaced return value with null for nl/tudelft/sem/template/users/domain/UserService::saveUser → KILLED
98	1. replaced boolean return with false for nl/tudelft/sem/template/users/domain/UserService::userExists → KILLED 2. replaced boolean return with true for nl/tudelft/sem/template/users/domain/UserService::userExists → KILLED
113	1. negated conditional → KILLED 2. removed call to nl/tudelft/sem/template/users/domain/User::setGender → KILLED
114	1. negated conditional → KILLED 2. removed call to nl/tudelft/sem/template/users/domain/User::setCertificate → KILLED
116	1. negated conditional → KILLED 2. removed call to nl/tudelft/sem/template/users/domain/User::setOrganisation → KILLED
118	1. removed call to nl/tudelft/sem/template/users/domain/User::setCompetitiveness → KILLED
121	1. replaced return value with null for nl/tudelft/sem/template/users/domain/UserService::updateUser → KILLED

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

We indeed managed to increase the mutation coverage for this class up to 100%. To achieve this we added methods to the handler classes so that we could verify that they were connected to another handler. After adding this functionality, we added a test to check that the handlers were connected in the right order, and by doing so killed all the surviving mutants. See commit.

UserService.java	100%	35/35	100%	17/17

Mutations

44	1. removed call to nl/tudelft/sem/template/users/domain/UserService::userValidationHandlerSetUp → KILLED
53	1. removed call to nl/tudelft/sem/template/users/domain/handlers/UserValidationHandler::setNext → KILLED
55	1. removed call to nl/tudelft/sem/template/users/domain/handlers/UserValidationHandler::setNext → KILLED
57	1. removed call to nl/tudelft/sem/template/users/domain/handlers/UserValidationHandler::setNext → KILLED
67	1. negated conditional → KILLED
70	1. replaced return value with null for nl/tudelft/sem/template/users/domain/UserService::getByEmail → KILLED
85	1. replaced return value with null for nl/tudelft/sem/template/users/domain/UserService::saveUser → KILLED
98	1. replaced boolean return with false for nl/tudelft/sem/template/users/domain/UserService::userExists → KILLED 2. replaced boolean return with true for nl/tudelft/sem/template/users/domain/UserService::userExists → KILLED
113	1. negated conditional → KILLED 2. removed call to nl/tudelft/sem/template/users/domain/User::setGender → KILLED
114	1. negated conditional → KILLED 2. removed call to nl/tudelft/sem/template/users/domain/User::setCertificate → KILLED
116	1. negated conditional → KILLED 2. removed call to nl/tudelft/sem/template/users/domain/User::setOrganisation → KILLED
118	1. removed call to nl/tudelft/sem/template/users/domain/User::setCompetitiveness → KILLED
121	1. replaced return value with null for nl/tudelft/sem/template/users/domain/UserService::updateUser → KILLED

Note: There were classes in the User microservice with < 70% mutation coverage that were provided by the template project, for example JwtRequestFilter or Application. We decided not to focus on these classes as they aren't the focus of the project/assignment.

4. CreateAccount class in Auth microservice

This is one of two classes in the Auth microservices that had a mutation testing score of 80% or lower according to PitTest. In fact, the score for this class **was exactly 80 percent**, thus the aim was to improve testing to kill all mutants created by Pit (improving to score to 100%).

Below, the report showing the surviving mutants **before** any improvements were made:

CreateAccount.java

Mutations

41	1. negated conditional → KILLED 2. negated conditional → KILLED
45	1. negated conditional → KILLED
49	1. negated conditional → KILLED
50	1. removed call to nl/tudelft/sem/template/auth/application/handlers/AuthHandler::handle → KILLED
84	1. negated conditional → KILLED
85	1. removed call to nl/tudelft/sem/template/auth/application/handlers/ExceptionHandler::handleException → KILLED
87	1. replaced boolean return with false for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::accountExists → SURVIVED
89	1. replaced boolean return with true for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::accountExists → KILLED
99	1. negated conditional → KILLED
100	1. removed call to nl/tudelft/sem/template/auth/application/handlers/ExceptionHandler::handleException → KILLED
102	1. replaced boolean return with true for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::verifySavedAccount → KILLED
105	1. replaced boolean return with false for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::verifySavedAccount → KILLED 2. replaced boolean return with true for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::verifySavedAccount → SURVIVED
116	1. replaced return value with "" for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::hashPassword → SURVIVED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

To kill these three mutants the following **improvements** were made to the test class:

- - To kill the mutant of the **accountExists** method, the mocked repository in the **testAlreadyExists** test was set to return an AccountCredentials object with the password hashed. In the situation before, the password was still un-hashed, causing the handle method to stop execution at the next if-block and thus still completing the test. Now the execution continues correctly and the next Handler is called, thus causing the test to fail when the mutation is present. See Commit.
- - To kill the mutant of the **verifySavedAccount** method, the **testInccorectPassword** test was created. This test tests if the handle method correctly aborts when the hashed password from the database does not correspond with the password that was provided. The test kills the mutant by detecting that the next Handler is still called, even though the hash of the password in the database does not match the original password. See Commit.
- - To kill the mutant of the **hashPassword** method, the **testCorrectHashing** test was added. The test checks if the credentials saved to the repository are indeed Credentials with the correct userId and a hashed version of the password. The test extracts the argument provided to the repository's save method and tests if this corresponds to the original credentials. The test kills the mutant by detecting that the hashed password to be saved (an empty String in this case) is not a correct hash obtained from the original password. See Commit.

The report for this class **after** the improvements looks as follows:

CreateAccount.java

Mutations

41 1. negated conditional → KILLED
2. negated conditional → KILLED
45 1. negated conditional → KILLED
49 1. negated conditional → KILLED
50 1. removed call to nl/tudelft/sem/template/auth/application/handlers/AuthHandler::handle → KILLED
84 1. negated conditional → KILLED
85 1. removed call to nl/tudelft/sem/template/auth/application/handlers/ExceptionHandler::handleException → KILLED
87 1. replaced boolean return with false for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::accountExists → KILLED
89 1. replaced boolean return with true for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::accountExists → KILLED
100 1. negated conditional → KILLED
2. negated conditional → KILLED
101 1. removed call to nl/tudelft/sem/template/auth/application/handlers/ExceptionHandler::handleException → KILLED
103 1. replaced boolean return with true for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::verifySavedAccount → KILLED
105 1. replaced boolean return with false for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::verifySavedAccount → KILLED
116 1. replaced return value with "" for nl/tudelft/sem/template/auth/application/handlers/CreateAccount::hashPassword → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

All the mutants have been killed, thus the mutation test score has been **improved to 100%.**

5. CreateToken class in Auth microservice (bonus)

The lowest scoring class in the Auth service was the CreateToken class, scoring **only 75%.** The report **before** the improvements looks as follows:

CreateToken.java

Mutations

43 1. negated conditional → KILLED
51 1. Replaced long addition with subtraction → KILLED
61 1. removed call to nl/tudelft/sem/template/auth/application/handlers/ExceptionHandler::handleException → NO_COVERAGE
92 1. replaced return value with "" for nl/tudelft/sem/template/auth/application/handlers/CreateToken::getToken → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

As can be seen, this class only spawned four mutations of which only one is surviving. The score can thus be easily perfected by applying the following **improvement**:

- To kill the mutant **removing the ExceptionHandler**, the **credentialsNullTest** was added. This test causes an exception in the handle method, because getUserId is called on the provided credentials, which are now set to null. It kills the mutant by detecting the ExceptionHandler has not been called. See Commit.

The report for this class **after** the improvements looks as follows:

CreateToken.java

Mutations

```
43 1. negated conditional → KILLED
51 1. Replaced long addition with subtraction → KILLED
61 1. removed call to nl/tudelft/sem/template/auth/application/handlers/ExceptionHandler::handleException → KILLED
92 1. replaced return value with "" for nl/tudelft/sem/template/auth/application/handlers/CreateToken::getToken → KILLED
```

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

All the mutants have been killed, thus the mutation test score has been **improved to 100%.**

6. NotificationController class in Notification microservice

This class had a mutation testing score of 65% according to PitTest.
These were the generated mutants:

Breakdown by Class

Name	Line Coverage	Mutation Coverage
NotificationController.java	95% 37/39	65% 11/17

Mutations

54	1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::helloWorld → KILLED
73	1. removed call to nl/tudelft/sem/template/notification/domain/Director::makeNotificationForPlayer → SURVIVED
77	1. removed call to nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotification → KILLED
78	1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayer → KILLED
80	1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayer → SURVIVED
100	1. removed call to nl/tudelft/sem/template/notification/domain/Director::makeNotificationForPlayerChanges → SURVIVED
103	1. removed call to nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotification → KILLED
104	1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayerChanges → KILLED
106	1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayerChanges → SURVIVED
125	1. removed call to nl/tudelft/sem/template/notification/domain/Director::makeNotificationForPublisher → SURVIVED
129	1. removed call to nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotification → KILLED
130	1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPublisher → KILLED
132	1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPublisher → NO_COVERAGE
144	1. removed call to org/springframework/mail/SimpleMailMessage::setTo → KILLED
145	1. removed call to org/springframework/mail/SimpleMailMessage::setSubject → KILLED
146	1. removed call to org/springframework/mail/SimpleMailMessage::setText → KILLED
147	1. removed call to org/springframework/mail/javamail/JavaMailSender::send → KILLED

We managed to get it to 100% coverage by changing the tests such that null return values are checked properly and all method calls on class fields are verified. To make this possible, we added the Builder field of the Director class in its constructor: dependency injection helps a lot during the testing process because mocking or spying on those fields allows verification on method calls. Another change to the production code was adding a 'setBuilder()' method inside the Director class and making the following modification in all API's from the NotificationController to be able to keep the Director as a mock in our tests.

Images representing changes in production code:

```
    */
    public void setBuilder(Builder builder) {
        this.builder = builder;
    }
```

```
    @PostMapping(value = "/participant",
                 consumes = {MediaType.APPLICATION_JSON_VALUE,
MediaType.APPLICATION_XML_VALUE},
                 produces = {MediaType.APPLICATION_JSON_VALUE,
MediaType.APPLICATION_XML_VALUE})
    public ResponseEntity<String> sendNotificationToPlayer(
                 @RequestBody NotificationRequestModelParticipant
notificationRequestModelParticipant) {
        try {
            Builder builder = new NotificationBuilder();
-           director = new Director(builder);
+           director.setBuilder(builder);

director.makeNotificationForPlayer(notificationRequestModelParticipant.getParticipan
tId(),
                    notificationRequestModelParticipant.getActivityId(),
                    notificationRequestModelParticipant.getTimeslot(),
                    notificationRequestModelParticipant.isDecision());
            sendNotification(builder.build());
            return ResponseEntity.ok("Email sent successfully.");
        } catch (Exception exception) {
            return ResponseEntity.badRequest().body(exception.getMessage());
        }
    }
```

Images representing changes in tests (only some examples, as they repeat for several test cases):

```
    @Test
    public void sendNotificationToPlayerOkTest() {
        notificationRequestModelParticipant = new
NotificationRequestModelParticipant(
                "test@gmail.com", 999, new Timeslot(
                        LocalDateTime.now(), LocalDateTime.now().plusHours(1)),
true);

-       NotificationController spy = Mockito.spy(notificationController);
-       doNothing().when(spy).sendNotification(any());
+       NotificationController spyCtrl = spy(notificationController);

-       ResponseEntity response =
spy.sendNotificationToPlayer(notificationRequestModelParticipant);
+       ResponseEntity response = notificationController
+               .sendNotificationToPlayer(notificationRequestModelParticipant);
        assertThat(response).isEqualTo(ResponseEntity.ok("Email sent
successfully."));
+
+       verify(spyCtrl.director).setBuilder(any(Builder.class));
+       verify(spyCtrl.director).makeNotificationForPlayer(
+               notificationRequestModelParticipant.getParticipantId(),
+               notificationRequestModelParticipant.getActivityId(),
+               notificationRequestModelParticipant.getTimeslot(),
+               notificationRequestModelParticipant.isDecision());
    }
```

```
    @Test
    public void sendNotificationToPlayerChangesBadRequestTest() {
        notificationRequestModelParticipantChanges = new
NotificationRequestModelParticipantChanges(
                random, 999, new Timeslot(
                LocalDateTime.now(), LocalDateTime.now().plusHours(1)));
        doThrow(new
RuntimeException()).when(javaMailSender).send(isA(SimpleMailMessage.class));
-       try {
-           ResponseEntity response = notificationController
-
.sendNotificationToPlayerChanges(notificationRequestModelParticipantChanges);
-           assertThat(response.getStatusCode()).isEqualTo(BAD_REQUEST);
-       } catch (RuntimeException e) {
-           assertThat(e).isNotNull();
-       }
+
+       ResponseEntity response = notificationController
+
.sendNotificationToPlayerChanges(notificationRequestModelParticipantChanges);
+       assertThat(response.getStatusCode()).isEqualTo(BAD_REQUEST);
+       assertThat(response).isNotNull();
    }
```

Statistics after these changes:

Breakdown by Class

Name	Line Coverage		Mutation Coverage	
NotificationController.java	100%	40/40	100%	20/20

Mutations

```
56  1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::helloWorld → KILLED
74  1. removed call to nl/tudelft/sem/template/notification/domain/Director::setBuilder → KILLED
75  1. removed call to nl/tudelft/sem/template/notification/domain/Director::makeNotificationForPlayer → KILLED
79  1. removed call to nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotification → KILLED
80  1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayer → KILLED
82  1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayer → KILLED
101 1. removed call to nl/tudelft/sem/template/notification/domain/Director::setBuilder → KILLED
102 1. removed call to nl/tudelft/sem/template/notification/domain/Director::makeNotificationForPlayerChanges → KILLED
105 1. removed call to nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotification → KILLED
106 1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayerChanges → KILLED
108 1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPlayerChanges → KILLED
126 1. removed call to nl/tudelft/sem/template/notification/domain/Director::setBuilder → KILLED
127 1. removed call to nl/tudelft/sem/template/notification/domain/Director::makeNotificationForPublisher → KILLED
131 1. removed call to nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotification → KILLED
132 1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPublisher → KILLED
134 1. replaced return value with null for nl/tudelft/sem/template/notification/controllers/NotificationController::sendNotificationToPublisher → KILLED
146 1. removed call to org/springframework/mail/SimpleMailMessage::setTo → KILLED
147 1. removed call to org/springframework/mail/SimpleMailMessage::setSubject → KILLED
148 1. removed call to org/springframework/mail/SimpleMailMessage::setText → KILLED
149 1. removed call to org/springframework/mail/javamail/JavaMailSender::send → KILLED
```

This is the link for the commit containing these changes:

https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM30b/-/commit/448c1c93890b008ea9c05e42b663b9302 f14071d

Assignment 3 – Mutation testing

Task 2

For this task we decided to select the **Matching microservice as core domain** as it holds the main functionality of the software by providing clients a way to select availabilities and get matched to activities based on some constraints.

Consequently, these are the classes we chose to mutate:

1. **TimeConstraintHandler class**

We considered **TimeConstraintHandler class** to be a critical one since it's part of the **Chain of Responsibility pattern** for filtering the activities based on clear time constraints given by the **"Rowing" scenario.**

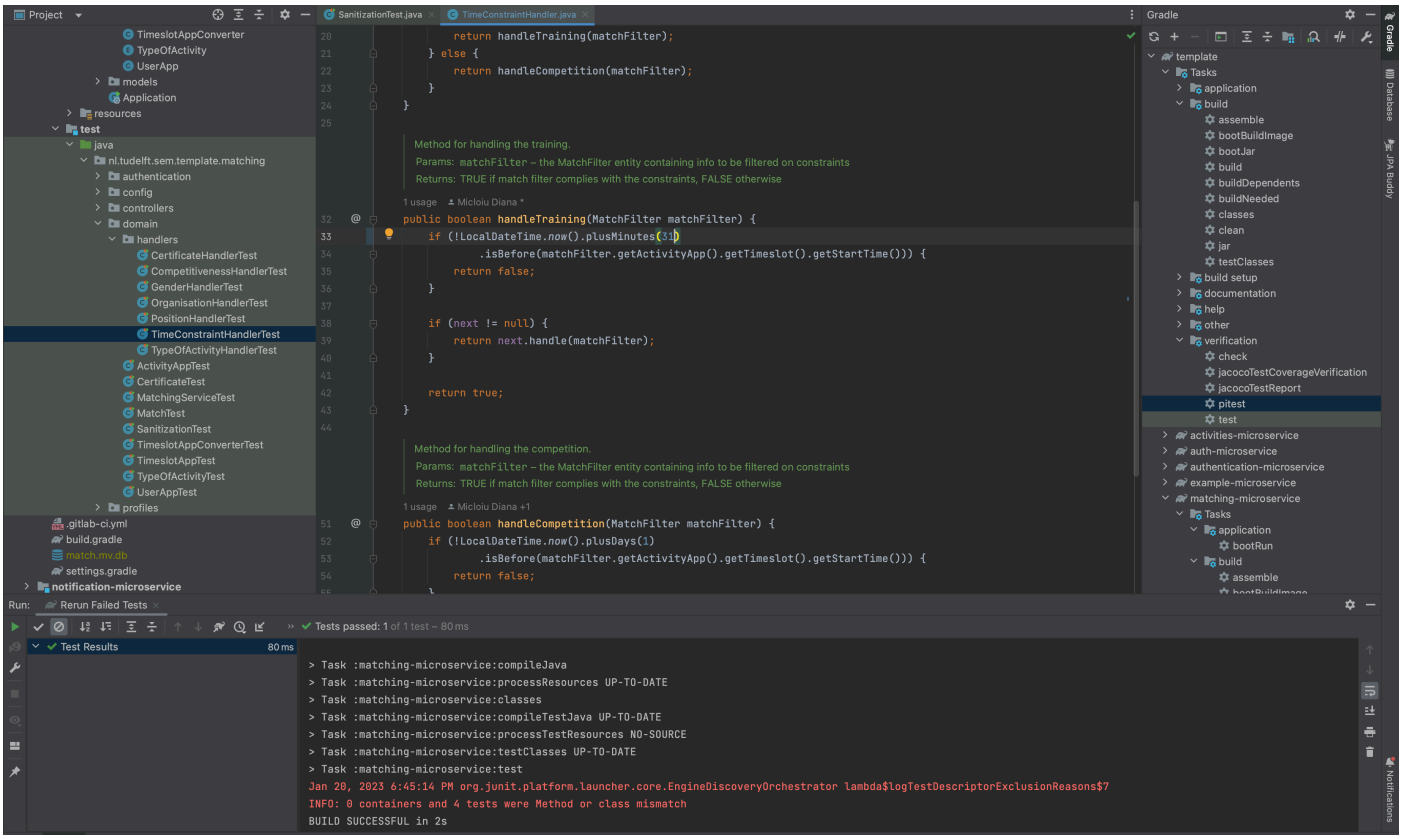
Being a **type of handler** in the layers of the responsibility pattern, it has as critical method the handle one which encapsulates the functionality of class. Moreover, due to refactoring the method is now split in two other ones based in the type of the activities (training or competition). As a consequence, we chose to mutate the **handleTraining** one by injecting a **Inline Constant Mutator type of mutant.**

Before

```
Method for handling the training.
Params: matchFilter – the MatchFilter entity containing info to be filtered on constraints
Returns: TRUE if match filter complies with the constraints, FALSE otherwise
1 usage  Micloiu Diana
public boolean handleTraining(MatchFilter matchFilter) {
    if (!LocalDateTime.now().plusMinutes(30)
            .isBefore(matchFilter.getActivityApp().getTimeslot().getStartTime())) {
        return false;
    }

    if (next != null) {
        return next.handle(matchFilter);
    }

    return true;
}
```

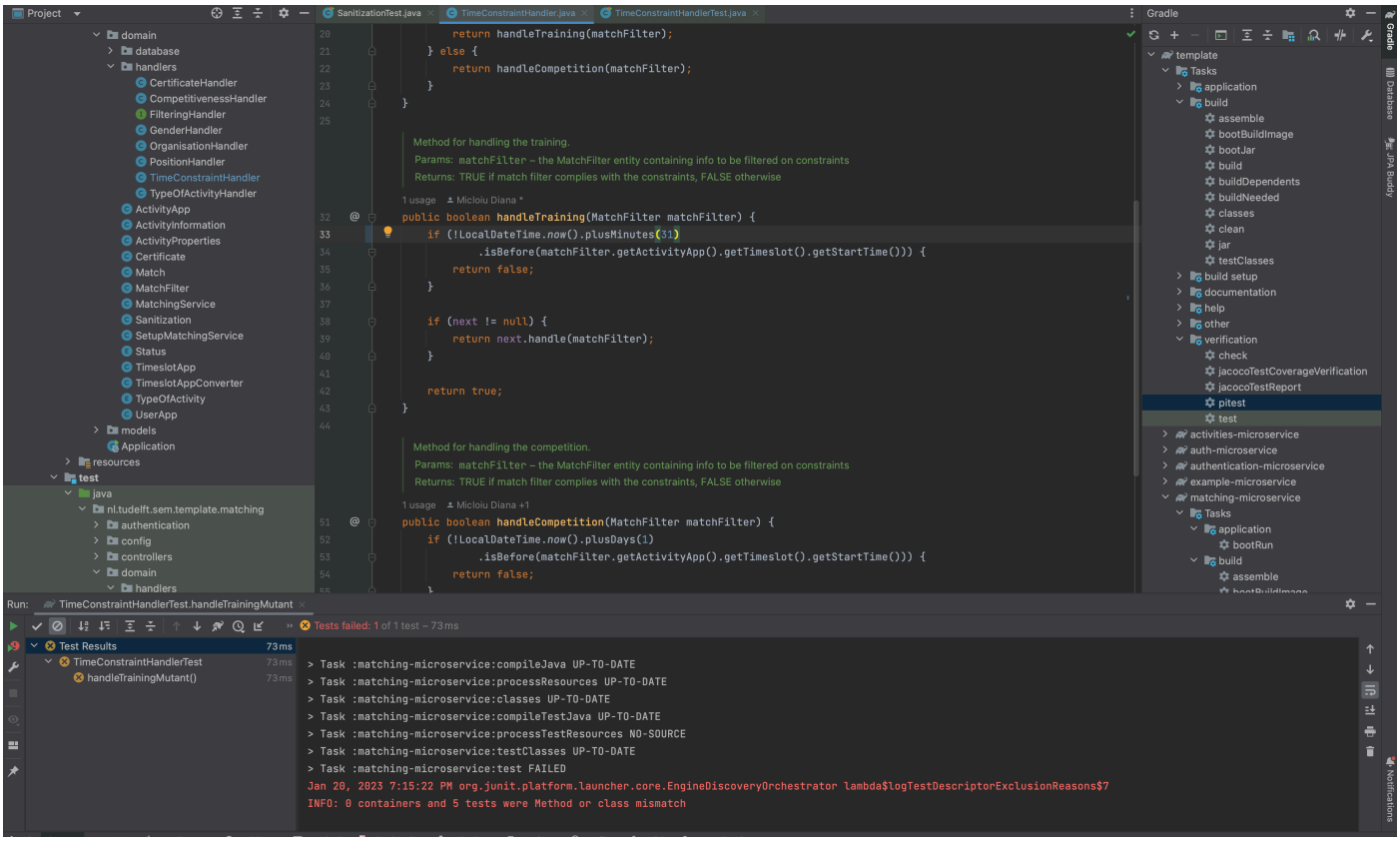
After

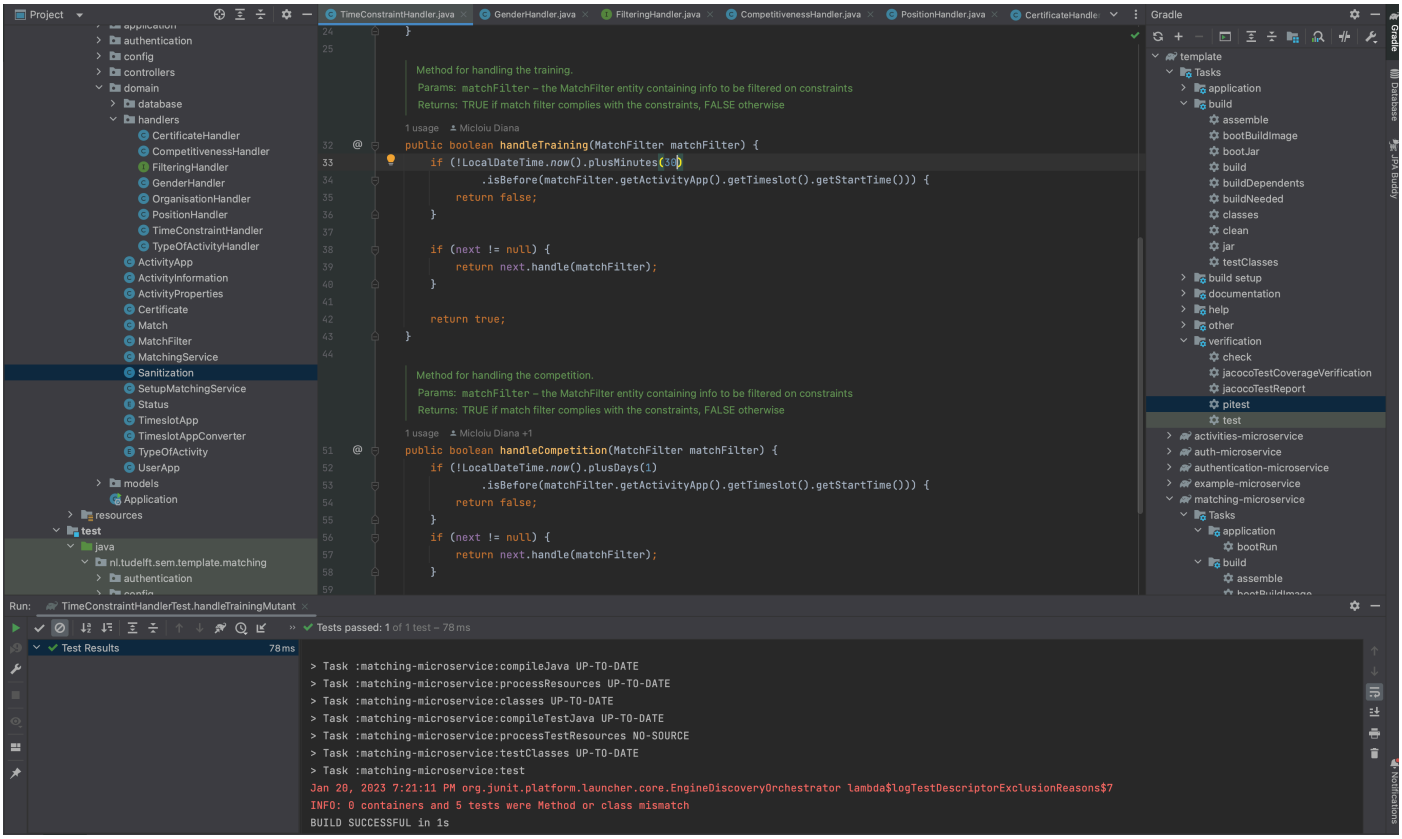
```
Method for handling the training.
Params: matchFilter – the MatchFilter entity containing info to be filtered on constraints
Returns: TRUE if match filter complies with the constraints, FALSE otherwise
1 usage  Micloiu Diana *
public boolean handleTraining(MatchFilter matchFilter) {
    if (!LocalDateTime.now().plusMinutes(31)
            .isBefore(matchFilter.getActivityApp().getTimeslot().getStartTime())) {
        return false;
    }

    if (next != null) {
        return next.handle(matchFilter);
    }

    return true;
}
```



Test added in commit https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM30b/-/commit/7027f1f0d8c29426407079870deb29d19e7bb36c

```
no usages   new *
@Test
void handleTrainingMutant() {
    HashMap<String, Integer> positions = new HashMap<>();
    positions.put("cox", 2);

    activityApp = new ActivityApp( id: 3L,
            ownerId: "l.tosa@tudelft.nl",
            new TimeslotApp(LocalDateTime.now().plusMinutes(31),
                    LocalDateTime.now().plusHours(2)),
            gender: null,  organisation: null, positions,  competition: false,
            TypeOfActivity.TRAINING,  certificate: "4+");
    user = new UserApp( email: "d.micloiu@icloud.com",  certificate: "C4",
            gender: "Female",  organisation: "SEM",  competitive: true);
    matchFilter = new MatchFilter(activityApp, new UserPreferences(new TimeslotApp(LocalDateTime.now(),
            LocalDateTime.now().plusHours(12)), user,  position: "cox"));

    assertThat(filteringHandler.handle(matchFilter)).isTrue();
}
```





2. SetUpMatchingService class

The **SetUpMatchingService class** is a critical one since it instantiates every layer in the **Chain of Responsibility pattern** for filtering the activities based the constraints specified by the **“Rowing” scenario.**

It’s only and main method is **filteringHandlerSetUp** and it’s a critical one for the functionality of the software (In the case of not setting up correctly the layers, the whole flow of the application is jeopardized since the rowers are going to be matched to activities that they shouldn’t be added to). Thus, we chose to mutate it by injecting an **Actual Type Change mutant.**

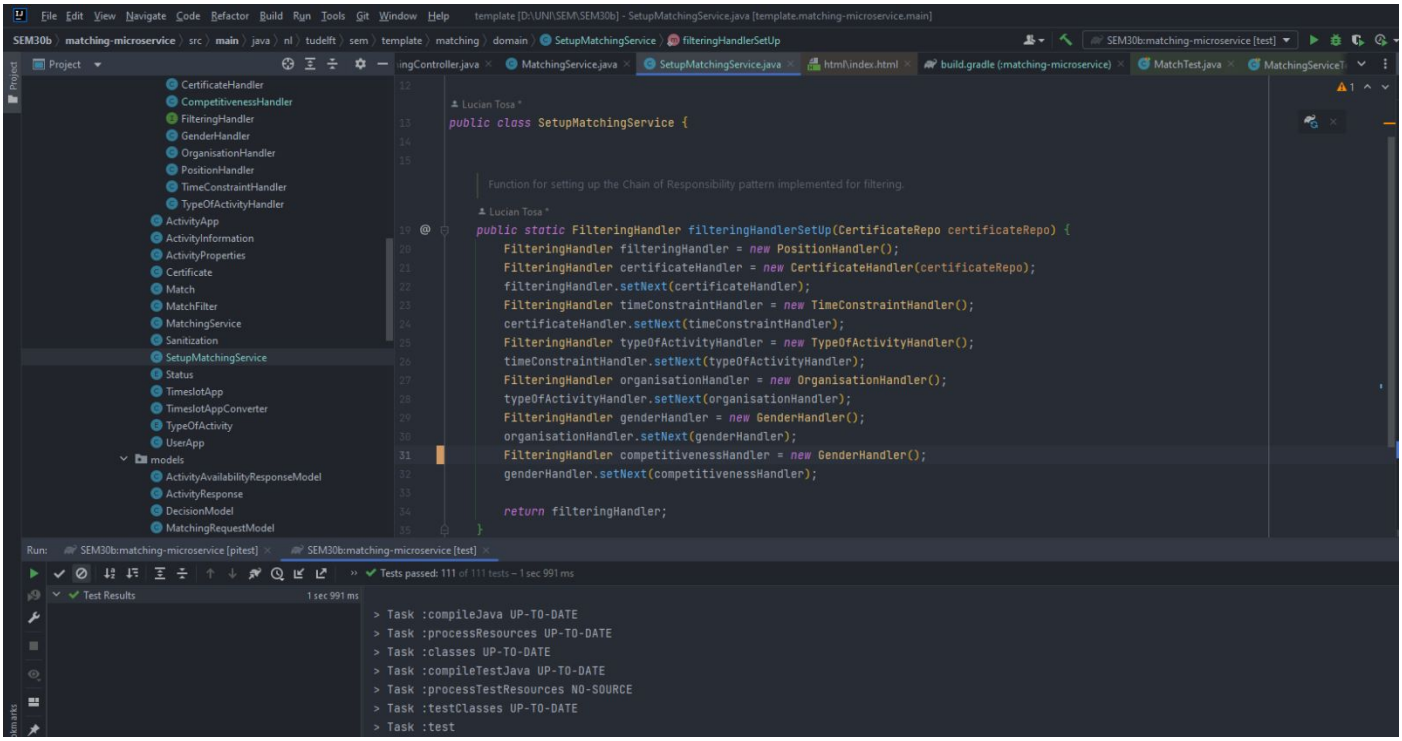
Being a **type of handler** in the layers of the responsibility pattern, it has as critical method the handle one which encapsulates the functionality of class. Moreover, due to refactoring the method is now split in two other ones based in the type of the activities (training or competition). As a consequence, we chose to mutate the **handleTraining** one by injecting a **Inline Constant Mutator type of mutant.**

Before

```
Function for setting up the Chain of Responsibility pattern implemented for filtering.
Lucian Tosa
public static FilteringHandler filteringHandlerSetUp(CertificateRepo certificateRepo) {
    FilteringHandler filteringHandler = new PositionHandler();
    FilteringHandler certificateHandler = new CertificateHandler(certificateRepo);
    filteringHandler.setNext(certificateHandler);
    FilteringHandler timeConstraintHandler = new TimeConstraintHandler();
    certificateHandler.setNext(timeConstraintHandler);
    FilteringHandler typeOfActivityHandler = new TypeOfActivityHandler();
    timeConstraintHandler.setNext(typeOfActivityHandler);
    FilteringHandler organisationHandler = new OrganisationHandler();
    typeOfActivityHandler.setNext(organisationHandler);
    FilteringHandler genderHandler = new GenderHandler();
    organisationHandler.setNext(genderHandler);
    FilteringHandler competitivenessHandler = new CompetitivenessHandler();
    genderHandler.setNext(competitivenessHandler);

    return filteringHandler;
}
```

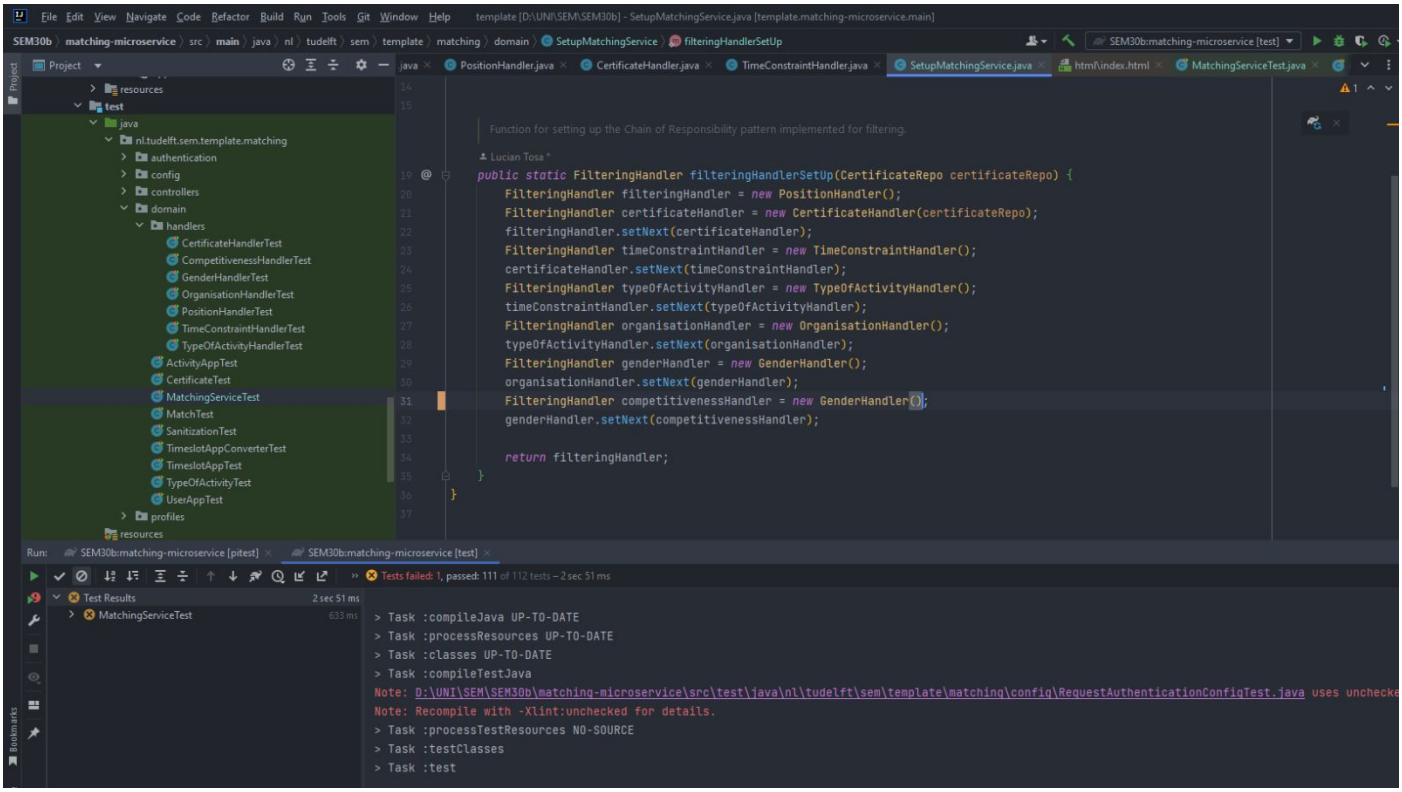
After

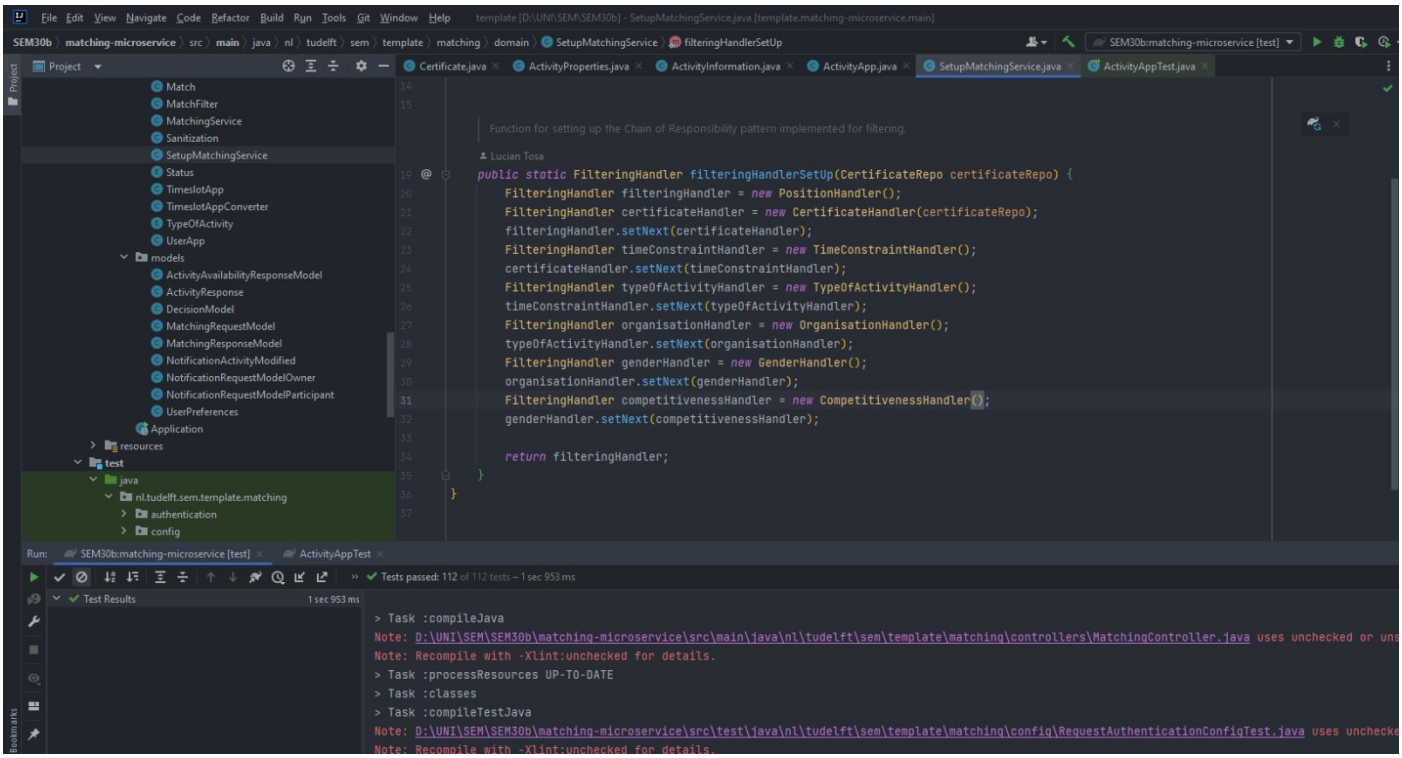
```
Function for setting up the Chain of Responsibility pattern implemented for filtering.
Lucian Tosa *
public static FilteringHandler filteringHandlerSetUp(CertificateRepo certificateRepo) {
    FilteringHandler filteringHandler = new PositionHandler();
    FilteringHandler certificateHandler = new CertificateHandler(certificateRepo);
    filteringHandler.setNext(certificateHandler);
    FilteringHandler timeConstraintHandler = new TimeConstraintHandler();
    certificateHandler.setNext(timeConstraintHandler);
    FilteringHandler typeOfActivityHandler = new TypeOfActivityHandler();
    timeConstraintHandler.setNext(typeOfActivityHandler);
    FilteringHandler organisationHandler = new OrganisationHandler();
    typeOfActivityHandler.setNext(organisationHandler);
    FilteringHandler genderHandler = new GenderHandler();
    organisationHandler.setNext(genderHandler);
    FilteringHandler competitivenessHandler = new GenderHandler();
    genderHandler.setNext(competitivenessHandler);

    return filteringHandler;
}
```



Test added in commit https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM30b/-/commit/657550a5dc9f7415eace284e1092952229b03360

```
@Test
void filterActivityCompetitivenessFail() {
    HashMap<String, Integer> positions = new HashMap<>();
    positions.put("cox", 2);

    user = new UserApp( email: "d.micloiu@icloud.com",  certificate: "C4",
            gender: "Male",  organisation: "SEM",  competitive: false);
    timeslot = new TimeslotApp(LocalDateTime.now(),
            LocalDateTime.now().plusDays(1).plusHours(4));

    ArrayList<ActivityApp> activities = new ArrayList<>();

    activities.add(new ActivityApp( id: 4L,
            ownerId: "l.tosa@tudelft.nl",
            new TimeslotApp(LocalDateTime.now().plusDays(1).plusHours(2),
                    LocalDateTime.now().plusDays(1).plusHours(3)),
            gender: "Male",  organisation: "SEM", positions,  competition: true, TypeOfActivity.COMPETITION,  certificate: "C4"));


    when(certificateRepo.getCertificateByName("C4")).thenReturn(Optional.of(new Certificate( id: 1L,  name: "C4+")));
    List<ActivityResponse> result = service.filterActivities(activities, new UserPreferences(timeslot, user,  position: "cox"));
    assertThat(result.size()).isEqualTo( expected: 0);

    verify(matchingRepo, times( wantedNumberOfInvocations: 0)).save(any());
}
```


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
template [D:\UNI\SEM\SEM30b] - SetupMatchingService.java [template.matching-microservice.main]
SEM30b matching-microservice src main java nl tudelft sem template matching domain SetupMatchingService filteringHandlerSetUp
SEM30b:matching-microservice [test]
Project
resources
test
java
nl.tudelft.sem.template.matching
authentication
config
controllers
domain
handlers
CertificateHandlerTest
CompetitivenessHandlerTest
GenderHandlerTest
OrganisationHandlerTest
PositionHandlerTest
TimeConstraintHandlerTest
TypeOfActivityHandlerTest
ActivityAppTest
CertificateTest
MatchingServiceTest
MatchTest
SanitizationTest
TimeslotAppConverterTest
TimeslotAppTest
TypeOfActivityTest
UserAppTest
profiles
resources
java
PositionHandler.java
CertificateHandler.java
TimeConstraintHandler.java
SetupMatchingService.java
html\index.html
MatchingServiceTest.java
Function for setting up the Chain of Responsibility pattern implemented for filtering.
Lucian Tosa *
public static FilteringHandler filteringHandlerSetUp(CertificateRepo certificateRepo) {
FilteringHandler filteringHandler = new PositionHandler();
FilteringHandler certificateHandler = new CertificateHandler(certificateRepo);
filteringHandler.setNext(certificateHandler);
FilteringHandler timeConstraintHandler = new TimeConstraintHandler();
certificateHandler.setNext(timeConstraintHandler);
FilteringHandler typeOfActivityHandler = new TypeOfActivityHandler();
timeConstraintHandler.setNext(typeOfActivityHandler);
FilteringHandler organisationHandler = new OrganisationHandler();
typeOfActivityHandler.setNext(organisationHandler);
FilteringHandler genderHandler = new GenderHandler();
organisationHandler.setNext(genderHandler);
FilteringHandler competitivenessHandler = new GenderHandler();
genderHandler.setNext(competitivenessHandler);
return filteringHandler;
}
}
Run: SEM30b:matching-microservice [pitest] SEM30b:matching-microservice [test]
Tests failed: 1, passed: 111 of 112 tests – 2 sec 51 ms
Test Results 2 sec 51 ms
MatchingServiceTest 633 ms
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava
Note: D:\UNI\SEM\SEM30b\matching-microservice\src\test\java\nl\tudelft\sem\template\matching\config\RequestAuthenticationConfigTest.java uses unchecke
Note: Recompile with -Xlint:unchecked for details.
> Task :processTestResources NO-SOURCE
> Task :testClasses
> Task :test


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
template [D:\UNI\SEM\SEM30b] - SetupMatchingService.java [template.matching-microservice.main]
SEM30b matching-microservice src main java nl tudelft sem template matching domain SetupMatchingService filteringHandlerSetUp
SEM30b:matching-microservice [test]
Project
Match
MatchFilter
MatchingService
Sanitization
SetupMatchingService
Status
TimeslotApp
TimeslotAppConverter
TypeOfActivity
UserApp
models
ActivityAvailabilityResponseModel
ActivityResponse
DecisionModel
MatchingRequestModel
MatchingResponseModel
NotificationActivityModified
NotificationRequestModelOwner
NotificationRequestModelParticipant
UserPreferences
Application
resources
test
java
nl.tudelft.sem.template.matching
authentication
config
Certificate.java
ActivityProperties.java
ActivityInformation.java
ActivityApp.java
SetupMatchingService.java
ActivityAppTest.java
Function for setting up the Chain of Responsibility pattern implemented for filtering.
Lucian Tosa
public static FilteringHandler filteringHandlerSetUp(CertificateRepo certificateRepo) {
FilteringHandler filteringHandler = new PositionHandler();
FilteringHandler certificateHandler = new CertificateHandler(certificateRepo);
filteringHandler.setNext(certificateHandler);
FilteringHandler timeConstraintHandler = new TimeConstraintHandler();
certificateHandler.setNext(timeConstraintHandler);
FilteringHandler typeOfActivityHandler = new TypeOfActivityHandler();
timeConstraintHandler.setNext(typeOfActivityHandler);
FilteringHandler organisationHandler = new OrganisationHandler();
typeOfActivityHandler.setNext(organisationHandler);
FilteringHandler genderHandler = new GenderHandler();
organisationHandler.setNext(genderHandler);
FilteringHandler competitivenessHandler = new CompetitivenessHandler();
genderHandler.setNext(competitivenessHandler);
return filteringHandler;
}
}
Run: SEM30b:matching-microservice [test] ActivityAppTest
Tests passed: 112 of 112 tests – 1 sec 953 ms
Test Results 1 sec 953 ms
> Task :compileJava
Note: D:\UNI\SEM\SEM30b\matching-microservice\src\main\java\nl\tudelft\sem\template\matching\controllers\MatchingController.java uses unchecked or uns
Note: Recompile with -Xlint:unchecked for details.
> Task :processResources UP-TO-DATE
> Task :classes
> Task :compileTestJava
Note: D:\UNI\SEM\SEM30b\matching-microservice\src\test\java\nl\tudelft\sem\template\matching\config\RequestAuthenticationConfigTest.java uses unchecke
Note: Recompile with -Xlint:unchecked for details.

3. MatchingService class

The **MatchingService class** encapsulates the main functionality of the **matching domain** because it takes care of the interaction of matching the user to some specific activities. Moreover, all the methods in the class are called within the **MatchingController** and make calls to the classes handling the communication with the ither microservices.

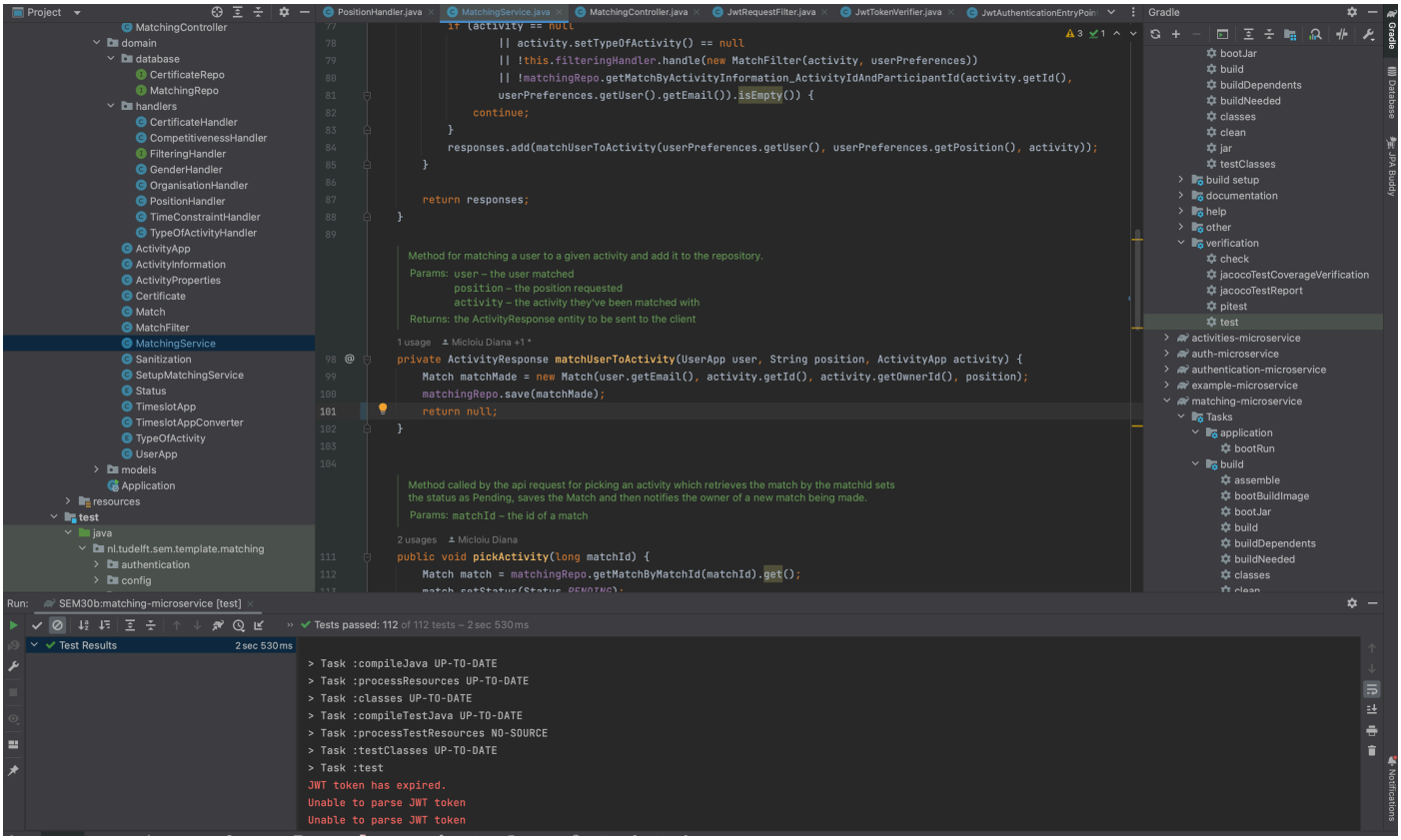
The method we chose to mutate is the **matchUserToActivity** one which provides the subsystem a way to create and save a new match to the repository when an user is matched to a certain activity and it also returns an **ActivityResponse** which should be visible for the client when making a request specifying the availability. Thus, we chose to in inject a **Null returns Mutator type of mutant.**

Before

```
Method for matching a user to a given activity and add it to the repository.
Params: user – the user matched
        position – the position requested
        activity – the activity they've been matched with
Returns: the ActivityResponse entity to be sent to the client
1 usage  Micloiu Diana +1
private ActivityResponse matchUserToActivity(UserApp user, String position, ActivityApp activity) {
    Match matchMade = new Match(user.getEmail(), activity.getId(), activity.getOwnerId(), position);
    matchingRepo.save(matchMade);
    return new ActivityResponse(matchMade.getMatchId(), activity.getType(), activity.getTimeslot());
}
```

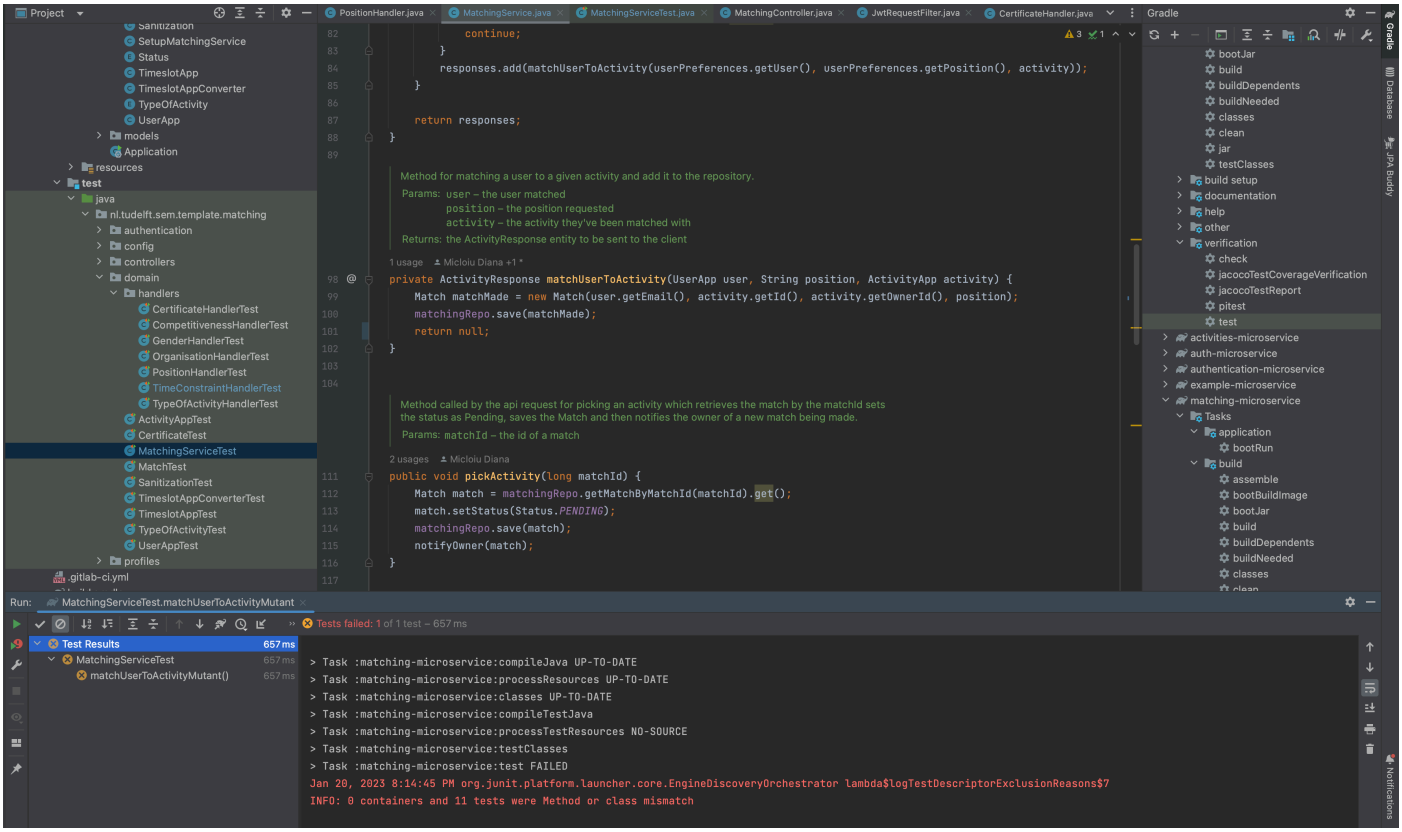
After

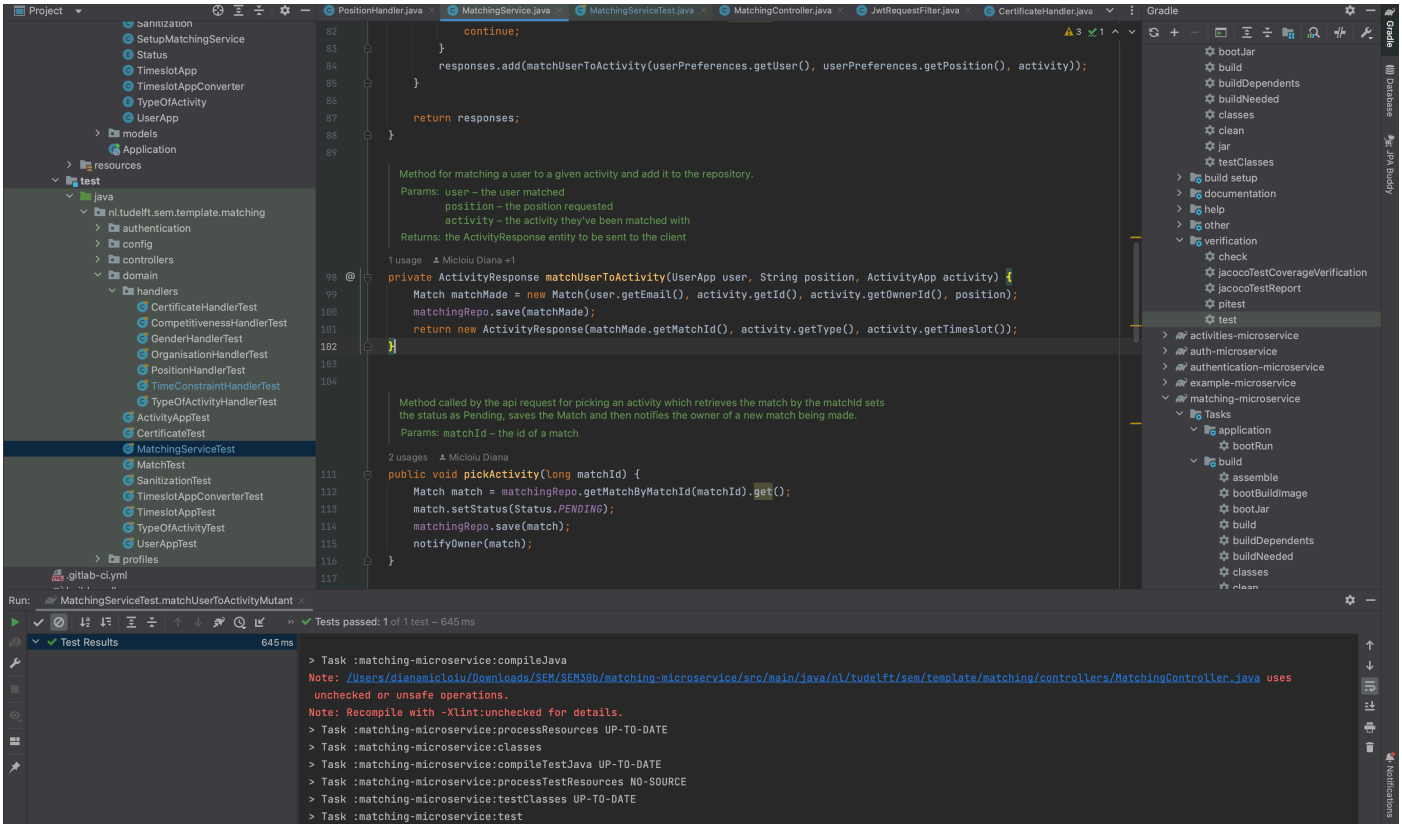
```
Method for matching a user to a given activity and add it to the repository.
Params: user – the user matched
        position – the position requested
        activity – the activity they've been matched with
Returns: the ActivityResponse entity to be sent to the client
1 usage  Micloiu Diana +1 *
private ActivityResponse matchUserToActivity(UserApp user, String position, ActivityApp activity) {
    Match matchMade = new Match(user.getEmail(), activity.getId(), activity.getOwnerId(), position);
    matchingRepo.save(matchMade);
    return null;
}
```



Test added in commit https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM30b/-/commit/bb3d35f5b9880312a0d5241ade02077ba85f6624

```
@Test
void matchUserToActivityMutant() {
    HashMap<String, Integer> positions = new HashMap<>();
    positions.put("cox", 2);

    ArrayList<ActivityApp> activities = new ArrayList<>();
    activities.add(null);
    activities.add(new ActivityApp( id: 1L,
            ownerId: "l.tosa@tudelft.nl",
            new TimeslotApp(LocalDateTime.now().plusMinutes(45),
                    LocalDateTime.now().plusHours(3)),
            gender: null, organisation: null, positions, competition: false, TypeOfActivity.TRAINING, certificate: "C4"));
    activities.add(new ActivityApp( id: 2L,
            ownerId: "l.tosa@tudelft.nl",
            new TimeslotApp(LocalDateTime.now(),
                    LocalDateTime.now().plusMinutes(45)),
            gender: null, organisation: null, positions, competition: false, TypeOfActivity.TRAINING, certificate: "C4"));

    activities.add(new ActivityApp( id: 3L,
            ownerId: "l.tosa@tudelft.nl",
            new TimeslotApp(LocalDateTime.now().plusDays(1),
                    LocalDateTime.now().plusDays(1).plusHours(1)),
            gender: "Female", organisation: "SEM", positions, competition: false, TypeOfActivity.COMPETITION, certificate: "4+");


    when(certificateRepo.getCertificateByName("C4")).thenReturn(Optional.of(new Certificate( id: 1L, name: "C4")));
    when(certificateRepo.getCertificateByName("4+")).thenReturn(Optional.of(new Certificate( id: 2L, name: "4+")));
    // one because one of the activities is 30 min after the timeslot given by the user
    List<ActivityResponse> result = service.filterActivities(activities, new UserPreferences(timeslot, user, position:
    assertThat(result.size()).isEqualTo( expected: 1);
    assertThat(result.get(0)).isNotNull();
}
```


Project
PositionHandler.java
MatchingService.java
MatchingServiceTest.java
MatchingController.java
JwtRequestFilter.java
CertificateHandler.java
Gradle
SetupMatchingService
Status
TimeslotApp
TimeslotAppConverter
TypeOfActivity
UserApp
models
Application
resources
test
java
nl.tudelft.sem.template.matching
authentication
config
controllers
domain
handlers
CertificateHandlerTest
CompetitivenessHandlerTest
GenderHandlerTest
OrganisationHandlerTest
PositionHandlerTest
TimeConstraintHandlerTest
TypeOfActivityHandlerTest
ActivityAppTest
CertificateTest
MatchingServiceTest
MatchTest
SanitizationTest
TimeslotAppConverterTest
TimeslotAppTest
TypeOfActivityTest
UserAppTest
profiles
.gitlab-ci.yml
continue;
}
responses.add(matchUserToActivity(userPreferences.getUser(), userPreferences.getPosition(), activity));
}
return responses;
}
Method for matching a user to a given activity and add it to the repository.
Params: user – the user matched
position – the position requested
activity – the activity they've been matched with
Returns: the ActivityResponse entity to be sent to the client
1 usage Micloiu Diana +1 *
private ActivityResponse matchUserToActivity(UserApp user, String position, ActivityApp activity) {
Match matchMade = new Match(user.getEmail(), activity.getId(), activity.getOwnerId(), position);
matchingRepo.save(matchMade);
return null;
}
Method called by the api request for picking an activity which retrieves the match by the matchId sets
the status as Pending, saves the Match and then notifies the owner of a new match being made.
Params: matchId – the id of a match
2 usages Micloiu Diana
public void pickActivity(long matchId) {
Match match = matchingRepo.getMatchByMatchId(matchId).get();
match.setStatus(Status.PENDING);
matchingRepo.save(match);
notifyOwner(match);
}
bootJar
build
buildDependents
buildNeeded
classes
clean
jar
testClasses
build setup
documentation
help
other
verification
check
jacocoTestCoverageVerification
jacocoTestReport
pitest
test
activities-microservice
auth-microservice
authentication-microservice
example-microservice
matching-microservice
Tasks
application
bootRun
build
assemble
bootBuildImage
bootJar
build
buildDependents
buildNeeded
classes
Run:
MatchingServiceTest.matchUserToActivityMutant
Tests failed: 1 of 1 test – 657 ms
Test Results 657 ms
MatchingServiceTest 657 ms
matchUserToActivityMutant() 657 ms
> Task :matching-microservice:compileJava UP-TO-DATE
> Task :matching-microservice:processResources UP-TO-DATE
> Task :matching-microservice:classes UP-TO-DATE
> Task :matching-microservice:compileTestJava
> Task :matching-microservice:processTestResources NO-SOURCE
> Task :matching-microservice:testClasses
> Task :matching-microservice:test FAILED
Jan 20, 2023 8:14:45 PM org.junit.platform.launcher.core.EngineDiscoveryOrchestrator lambda$logTestDescriptorExclusionReasons$7
INFO: 0 containers and 11 tests were Method or class mismatch


Project
PositionHandler.java
MatchingService.java
MatchingServiceTest.java
MatchingController.java
JwtRequestFilter.java
CertificateHandler.java
Gradle
SetupMatchingService
Status
TimeslotApp
TimeslotAppConverter
TypeOfActivity
UserApp
models
Application
resources
test
java
nl.tudelft.sem.template.matching
authentication
config
controllers
domain
handlers
CertificateHandlerTest
CompetitivenessHandlerTest
GenderHandlerTest
OrganisationHandlerTest
PositionHandlerTest
TimeConstraintHandlerTest
TypeOfActivityHandlerTest
ActivityAppTest
CertificateTest
MatchingServiceTest
MatchTest
SanitizationTest
TimeslotAppConverterTest
TimeslotAppTest
TypeOfActivityTest
UserAppTest
profiles
.gitlab-ci.yml
continue;
}
responses.add(matchUserToActivity(userPreferences.getUser(), userPreferences.getPosition(), activity));
}
return responses;
}
Method for matching a user to a given activity and add it to the repository.
Params: user – the user matched
position – the position requested
activity – the activity they've been matched with
Returns: the ActivityResponse entity to be sent to the client
1 usage Micloiu Diana +1
private ActivityResponse matchUserToActivity(UserApp user, String position, ActivityApp activity) {
Match matchMade = new Match(user.getEmail(), activity.getId(), activity.getOwnerId(), position);
matchingRepo.save(matchMade);
return new ActivityResponse(matchMade.getMatchId(), activity.getType(), activity.getTimeslot());
}
Method called by the api request for picking an activity which retrieves the match by the matchId sets
the status as Pending, saves the Match and then notifies the owner of a new match being made.
Params: matchId – the id of a match
2 usages Micloiu Diana
public void pickActivity(long matchId) {
Match match = matchingRepo.getMatchByMatchId(matchId).get();
match.setStatus(Status.PENDING);
matchingRepo.save(match);
notifyOwner(match);
}
bootJar
build
buildDependents
buildNeeded
classes
clean
jar
testClasses
build setup
documentation
help
other
verification
check
jacocoTestCoverageVerification
jacocoTestReport
pitest
test
activities-microservice
auth-microservice
authentication-microservice
example-microservice
matching-microservice
Tasks
application
bootRun
build
assemble
bootBuildImage
bootJar
build
buildDependents
buildNeeded
classes
Run:
MatchingServiceTest.matchUserToActivityMutant
Tests passed: 1 of 1 test – 645 ms
Test Results 645 ms
> Task :matching-microservice:compileJava
Note: /Users/dianamicloiu/Downloads/SEM/SEM30b/matching-microservice/src/main/java/nl/tudelft/sem/template/matching/controllers/MatchingController.java uses
unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
> Task :matching-microservice:processResources UP-TO-DATE
> Task :matching-microservice:classes
> Task :matching-microservice:compileTestJava UP-TO-DATE
> Task :matching-microservice:processTestResources NO-SOURCE
> Task :matching-microservice:testClasses UP-TO-DATE
> Task :matching-microservice:test

4. OrganisationHandler class

This class is similar to the **TimeConstraintHandler class** as it is also a critical one since it's part of the **Chain of Responsibility pattern** for filtering the activities based on organisation constraints given by the **"Rowing" scenario.**

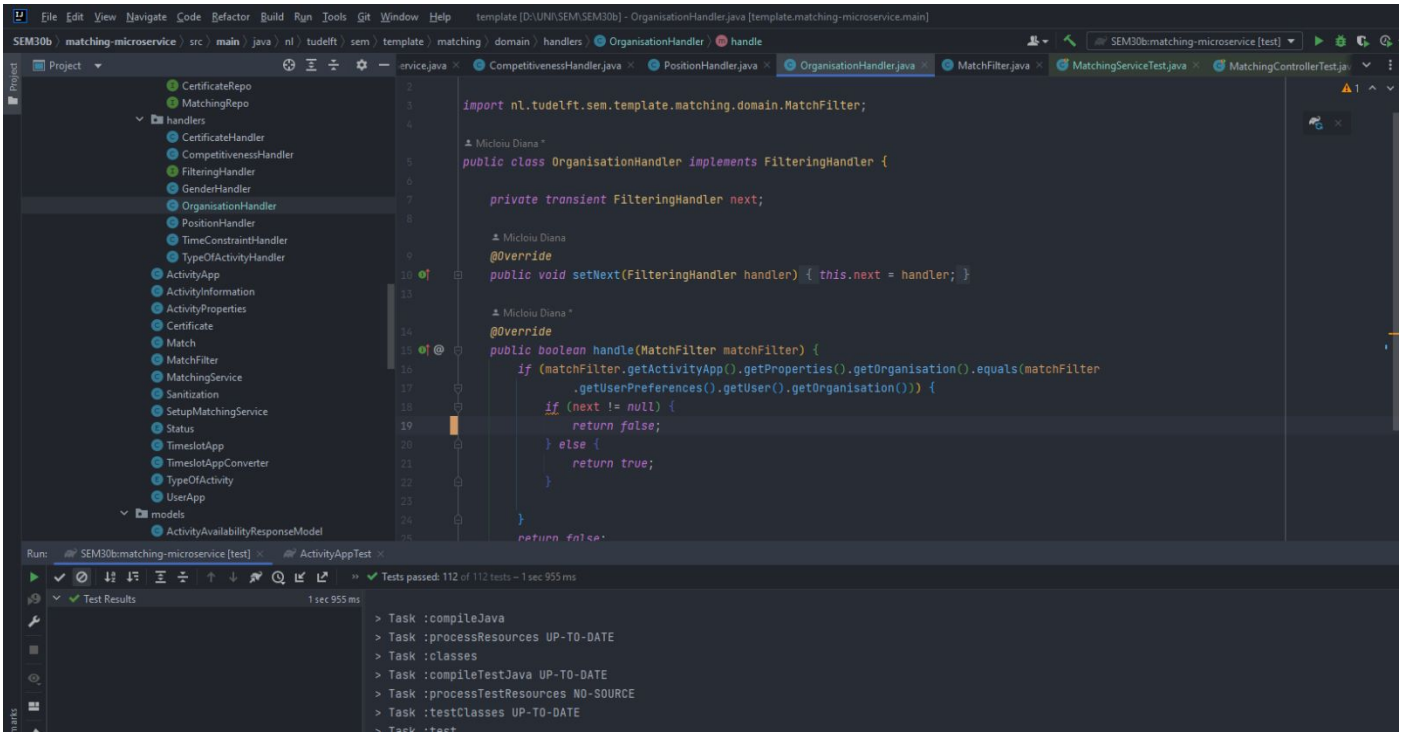
Being a **type of handler** in the layers of the responsibility pattern, it has as critical method the handle one which encapsulates the functionality of class. In contrast with what we have done for the other class, in the **OrganisationHandler class** we introduced a different type fo mutant namely **False returns Mutator.** By doing this, we ensured that the chain works properly when handling the next layer of filtering.

Before

```
@Override
public boolean handle(MatchFilter matchFilter) {
    if (matchFilter.getActivityApp().getProperties().getOrganisation().equals(matchFilter
            .getUserPreferences().getUser().getOrganisation())) {
        if (next != null) {
            return next.handle(matchFilter);
        } else {
            return true;
        }

    }
    return false;
}
```

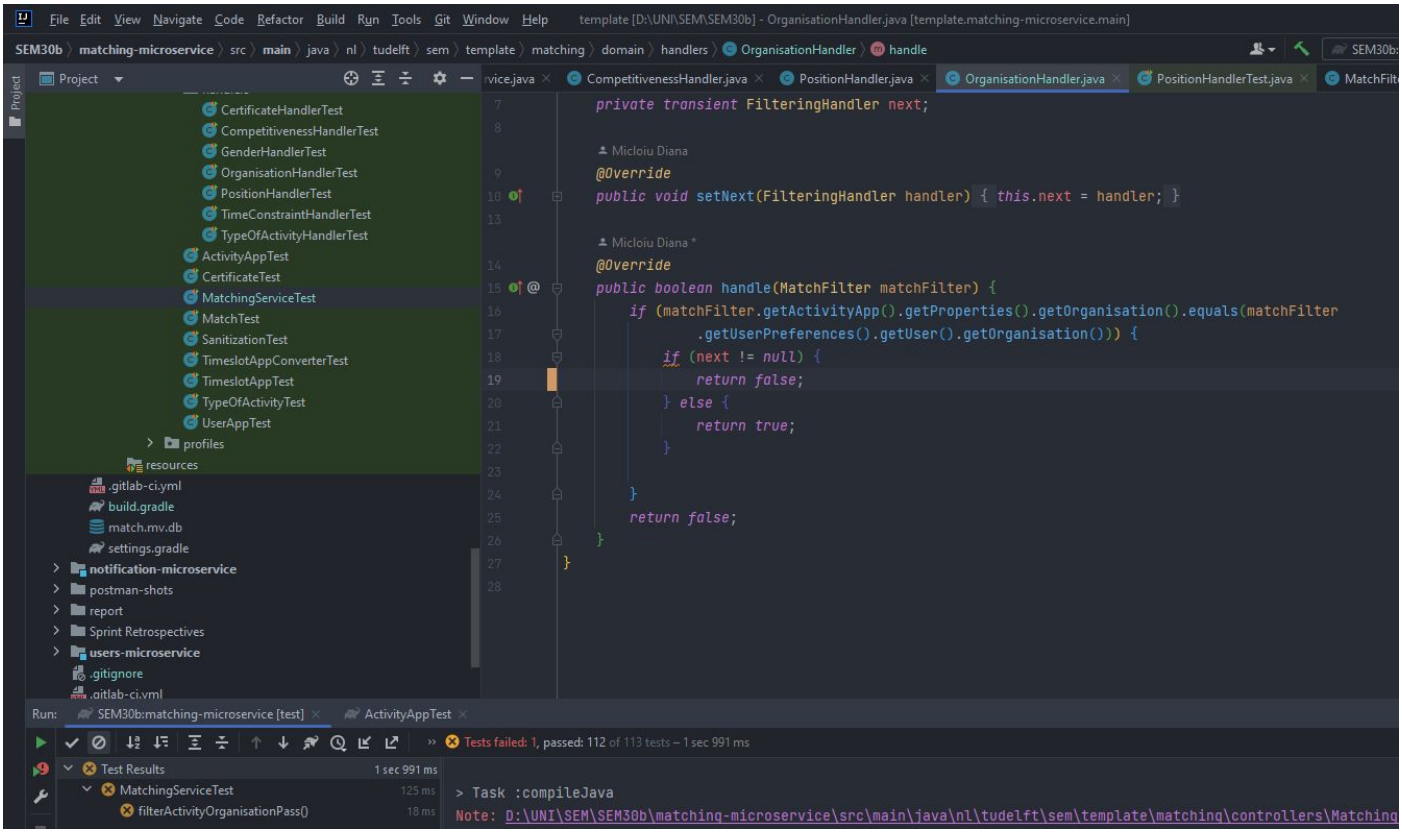
After

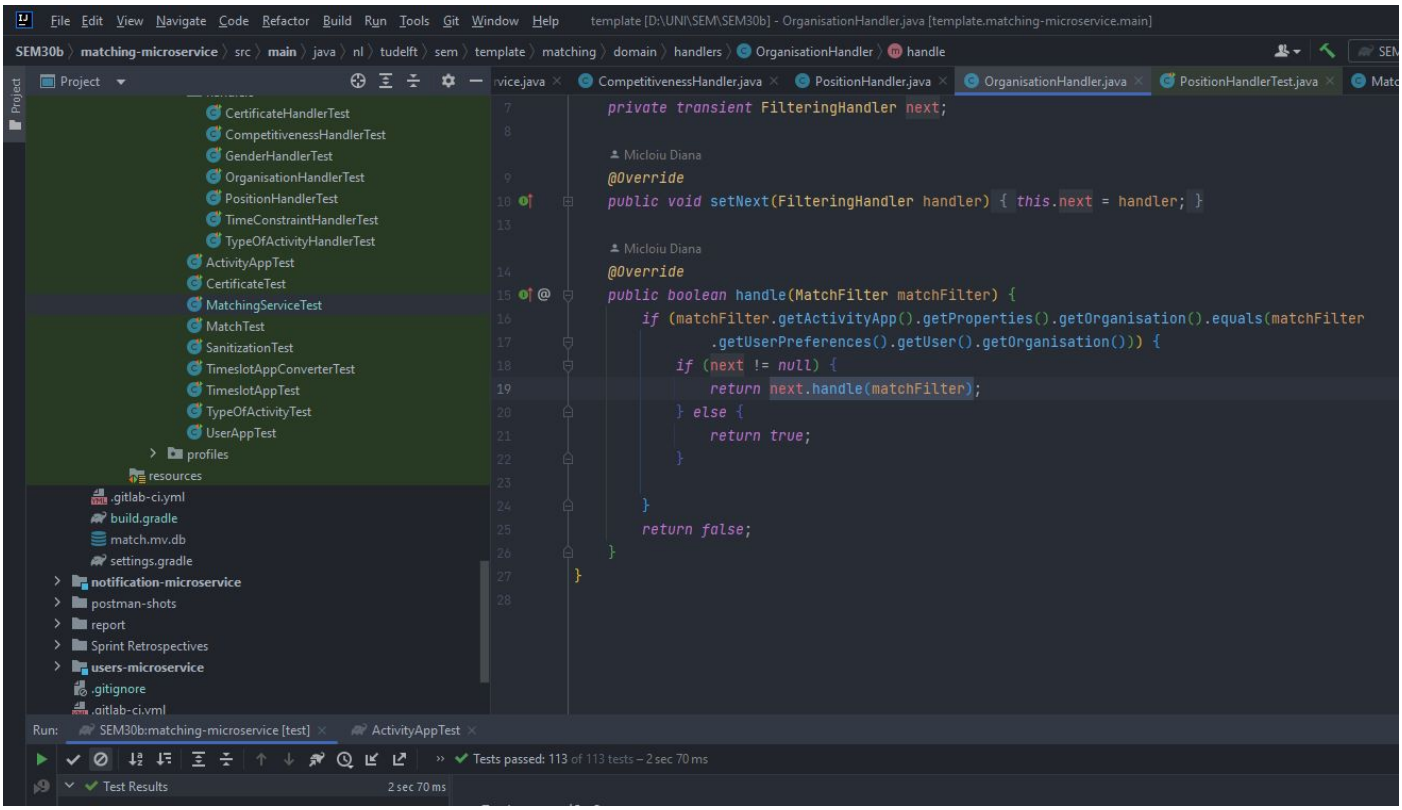
```
@Override
public boolean handle(MatchFilter matchFilter) {
    if (matchFilter.getActivityApp().getProperties().getOrganisation().equals(matchFilter
            .getUserPreferences().getUser().getOrganisation())) {
        if (next != null) {
            return false;
        } else {
            return true;
        }

    }
    return false;
}
```



Test added in commit https://gitlab.ewi.tudelft.nl/cse2115/2022-2023/SEM30b/-/commit/46fdfb4abd37340b5367b4203ad287af3c51d947

```
@Test
void filterActivityOrganisationPass() {
    HashMap<String, Integer> positions = new HashMap<>();
    positions.put("cox", 2);

    user = new UserApp( email: "d.micloiu@icloud.com", certificate: "C4",
            gender: "Male", organisation: "SEM", competitive: true);
    timeslot = new TimeslotApp(LocalDateTime.now(),
            LocalDateTime.now().plusDays(1).plusHours(4));

    ArrayList<ActivityApp> activities = new ArrayList<>();

    activities.add(new ActivityApp( id: 4L,
            ownerId: "l.tosa@tudelft.nl",
            new TimeslotApp(LocalDateTime.now().plusDays(1).plusHours(2),
                    LocalDateTime.now().plusDays(1).plusHours(3)),
            gender: "Male", organisation: "SEM", positions, competition: true, TypeOfActivity.COMPETITION, certificate: "C4"));


    when(certificateRepo.getCertificateByName("C4")).thenReturn(Optional.of(new Certificate( id: 1L, name: "C4+")));
    List<ActivityResponse> result = service.filterActivities(activities, new UserPreferences(timeslot, user, position: "cox"));
    assertThat(result.size()).isEqualTo( expected: 1);


    Match matchMade = new Match( participantId: "d.micloiu@tudelft.nl",
            activityId: 2L,
            ownerId: "l.tosa@tudelft.nl",
            position: "cox");
    verify(matchingRepo, times( wantedNumberOfInvocations: 1)).save(matchMade);
}
```


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
template [D:\UNI\SEM\SEM30b] - OrganisationHandler.java [template.matching-microservice.main]
SEM30b matching-microservice src main java nl tudelft sem template matching domain handlers OrganisationHandler handle
Project
CertificateHandlerTest
CompetitivenessHandlerTest
GenderHandlerTest
OrganisationHandlerTest
PositionHandlerTest
TimeConstraintHandlerTest
TypeOfActivityHandlerTest
ActivityAppTest
CertificateTest
MatchingServiceTest
MatchTest
SanitizationTest
TimeslotAppConverterTest
TimeslotAppTest
TypeOfActivityTest
UserAppTest
profiles
resources
.gitlab-ci.yml
build.gradle
match.mv.db
settings.gradle
notification-microservice
postman-shots
report
Sprint Retrospectives
users-microservice
.gitignore
.gitlab-ci.yml
CompetitivenessHandler.java
PositionHandler.java
OrganisationHandler.java
PositionHandlerTest.java
private transient FilteringHandler next;
Micloiu Diana
@Override
public void setNext(FilteringHandler handler) { this.next = handler; }
Micloiu Diana *
@Override
public boolean handle(MatchFilter matchFilter) {
if (matchFilter.getActivityApp().getProperties().getOrganisation().equals(matchFilter
.getUserPreferences().getUser().getOrganisation())) {
if (next != null) {
return false;
} else {
return true;
}
}
return false;
}
}
Run: SEM30b:matching-microservice [test] ActivityAppTest
Tests failed: 1, passed: 112 of 113 tests – 1 sec 991 ms
Test Results 1 sec 991 ms
MatchingServiceTest 125 ms
filterActivityOrganisationPass() 18 ms
> Task :compileJava
Note: D:\UNI\SEM\SEM30b\matching-microservice\src\main\java\nl\tudelft\sem\template\matching\controllers\Matching


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
template [D:\UNI\SEM\SEM30b] - OrganisationHandler.java [template.matching-microservice.main]
SEM30b matching-microservice src main java nl tudelft sem template matching domain handlers OrganisationHandler handle
Project
CertificateHandlerTest
CompetitivenessHandlerTest
GenderHandlerTest
OrganisationHandlerTest
PositionHandlerTest
TimeConstraintHandlerTest
TypeOfActivityHandlerTest
ActivityAppTest
CertificateTest
MatchingServiceTest
MatchTest
SanitizationTest
TimeslotAppConverterTest
TimeslotAppTest
TypeOfActivityTest
UserAppTest
profiles
resources
.gitlab-ci.yml
build.gradle
match.mv.db
settings.gradle
notification-microservice
postman-shots
report
Sprint Retrospectives
users-microservice
.gitignore
.gitlab-ci.yml
CompetitivenessHandler.java
PositionHandler.java
OrganisationHandler.java
PositionHandlerTest.java
private transient FilteringHandler next;
Micloiu Diana
@Override
public void setNext(FilteringHandler handler) { this.next = handler; }
Micloiu Diana
@Override
public boolean handle(MatchFilter matchFilter) {
if (matchFilter.getActivityApp().getProperties().getOrganisation().equals(matchFilter
.getUserPreferences().getUser().getOrganisation())) {
if (next != null) {
return next.handle(matchFilter);
} else {
return true;
}
}
return false;
}
}
Run: SEM30b:matching-microservice [test] ActivityAppTest
Tests passed: 113 of 113 tests – 2 sec 70 ms
Test Results 2 sec 70 ms