We used JUNIT tests for testing the functionality of the backend (calculating, fetching and changing of data), black box testing with giving input and looking for the expected output of the application, Selenium was used to test the overall behavior of the application.

Account Test

The account functionality was tested by first signing in an existing user (CvdB) and fetching his account data and asserting the returned user object against the username, first name, last name, total steps, total kilometers, height and weight.

The second test does a change of the user's account of the first name, last name, weight, height and then saving the this data into the database and the fetching of the information again to assert with the new values.

Authentication Test

In the first test we test the functionality to login with incorrect pass and right username and any combination of that for CvdB. For the register we use a username that is not in the database to register the user and also we test it as existing username which is not allowed.

In the second test we test the hashing of password we are giving the correct hashing of the password against the correct plain pass. Then we are asserting incorrect hashing with the correct plain pass and incorrect password asserted with correct hashing.

Run Test

For the run test first we use run 1 and 8 of CvdB to check if they return the right average pace, total steps and duration. We also test if the count functionality returns 8 runs as initially given by TRIMM.

For the next test we test the step data of a run by taking the 10th step and asserting the values returned by the StepData object e.g breaking off power right, time, to_right and etc. against the values from the database.

For the last test we assert the meta data of run 1 BodyPack object which has the distance, time, description and date. We assert them against the actual values in the database.

Shoe test

For the shoe test we check for given sid if the right data of the shoe is fetched e.g. model, brand, heel(mm),forefoot(mm), drop(mm),weight(gr) against the actual values from the database.

Verification Test

For the verification test we check if it returns true for existing email in the database and false for new email. Then we generate token which we map to the new email and assert it. We do any combination like wrong token with right email, correct token with wrong email and etc.

Session Token Test

The SessionTokenTest tests all functionality that involves a SessionToken. Before every test user CvdB is logged in and the reply is saved. The first test tests if the login is successful and has a Session Token appended to it. The second test tests whether the assigned session token will return the user. After that the next test tests if the expire function works. This is done by lowering the expire value, logging in the user, waiting and finally the sessiontoken should not return the user. The final test tests if the function that adds time to a date works.

Testing the requests for the REST endpoints

We tested every request manually with Postman to see if it is working and behaving as expected.

Application testing

We tested our application as a whole thoroughly on our local Tomcat servers. We also used Selenium to create automated tests where we test the functionality as whole-accessing the login and logging in and then logging out, accessing the compare page and clicking the buttons to compare run data, accessing the run overview page clicking button to remove data from the graph and scrolling down to click buttons on the second graph,editing the account information and saving the new information and etc.