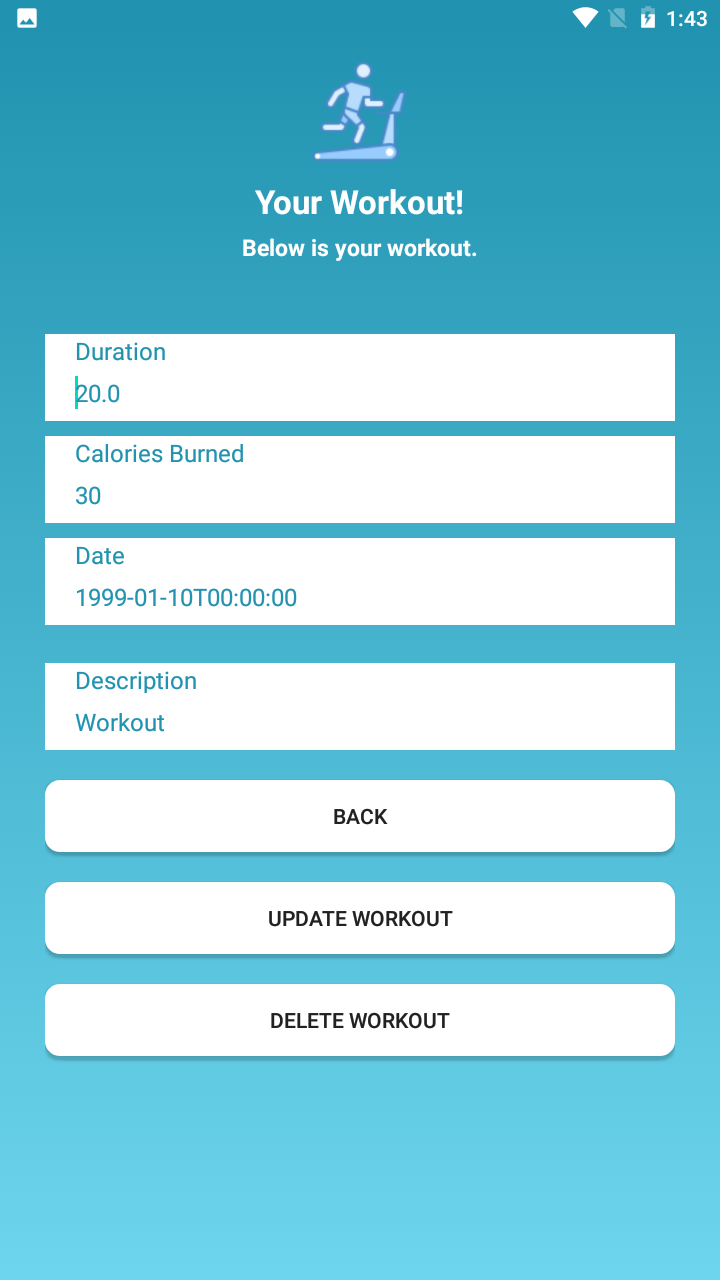
|  |  |
| --- | --- |
|  | Fitness Tracker |
| Fitness TrackerAndroid App | |

|  |
| --- |
| Fitness Tracker Android App  What is Fitness Tracker?  Fitness Tracker is an to view your BMI, view your BMR, add a workout, view previous workouts and get a breakdown of stats  Technologies used:  Okhttp3  OkHTTP3 is an open source library designed for android to be an efficient HTTP client.  Okhttp3 is used for our get, post, put and delete methods entirely.  Java.  Used instead of Kotlin as JAVA has been thought.  Android Utilities used:   * Card view – Used to display items in a nice fashion * Recycler View – Create instances of workouts and populate a array list in which the recycler view can display all our workouts. * Shared Preferences – Used to Verify that a user does not exist or exist. Store strings variables e.g. user id locally on android so it can be retrieved even when application is closed. Used to check if it’s a users first time using the app.   Testing – Espresso.  Espresso is used to end to end test our app. From view BMI,BMR, View User Stats, add workouts and view workouts.  Screenshots of app from start to finish  :  Adding in a users details:  First Screen if userid does not exist, this is the screen you will see.  Contains Validation if wrong information is added.      Main page after adding a user:  Main Page utilizing image views, icons and text so a user can move throughout the app    View BMR:  View BMR page represents a get request based on the user ID and populate the field below.      View BMR:  View BMI page represents a get request based on the user ID and populate the field below.    User Details:  User details : on this page we can update or delete our user  User Details:  User details : on this page we can update or delete our user.    User Stats:  User stats populates the textview with get request using okhttp for the userId it has.    Add a workout page:  Page that will add a workout with editexts that will send this information to the service. Using put in okhttp  Added Workout details:  Added workout with details populated  Workout List, populated with query.  Query being workouts for that userId. Workouts populated via arraylist and viewed with RecyclerView. |

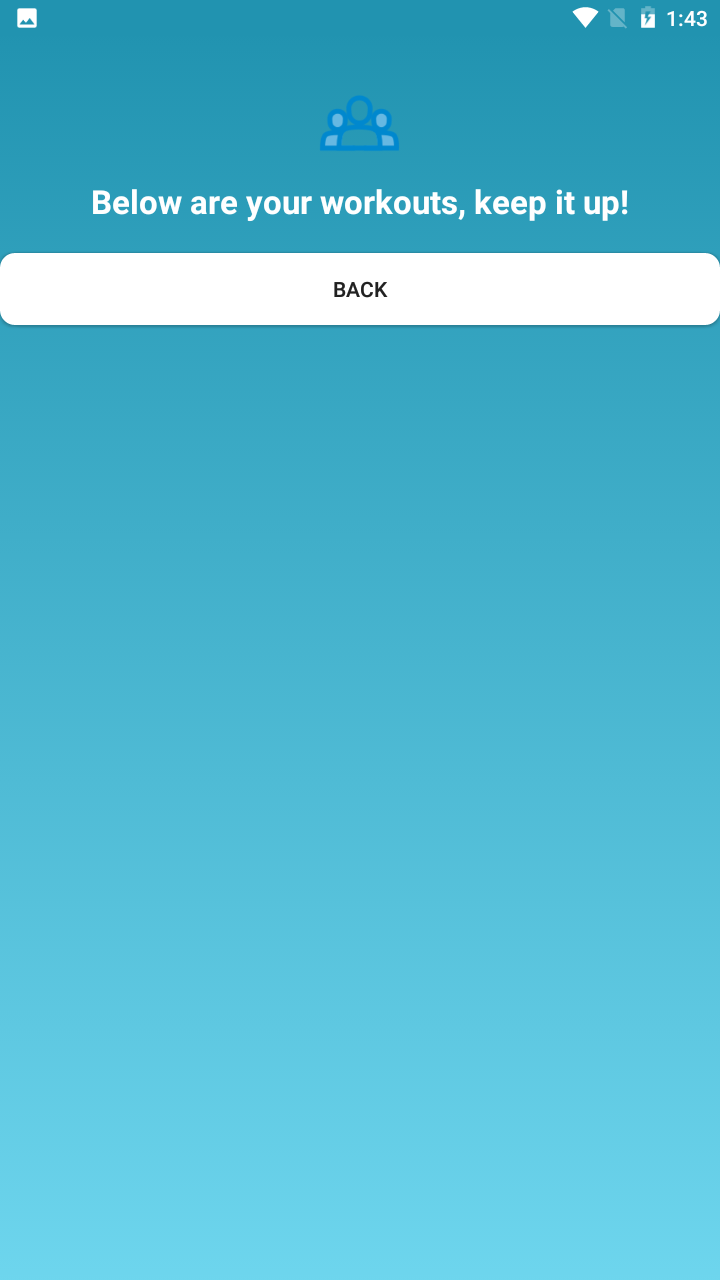
View Workout, can update and delete from here:

Can now view edit and delete this workout and show the information now exists



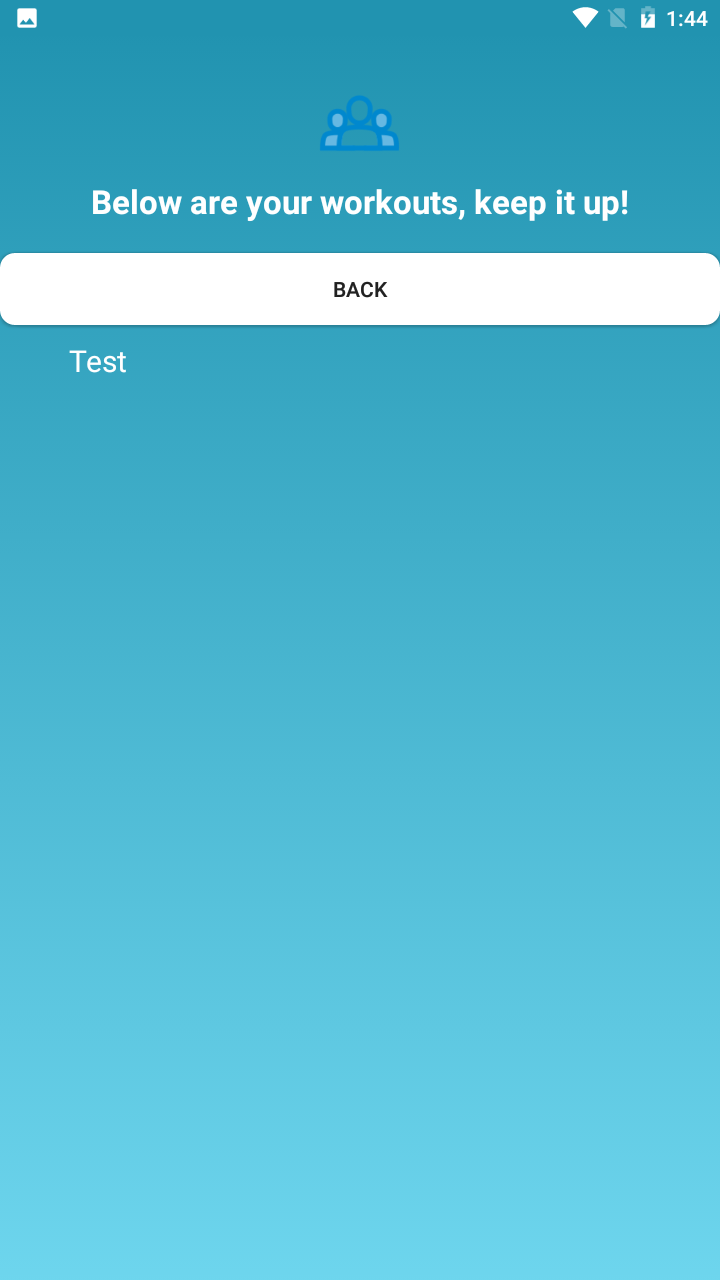
After Deleting this workout:

Shows that the workout has been deleted and is nowhere available on the app anymore.



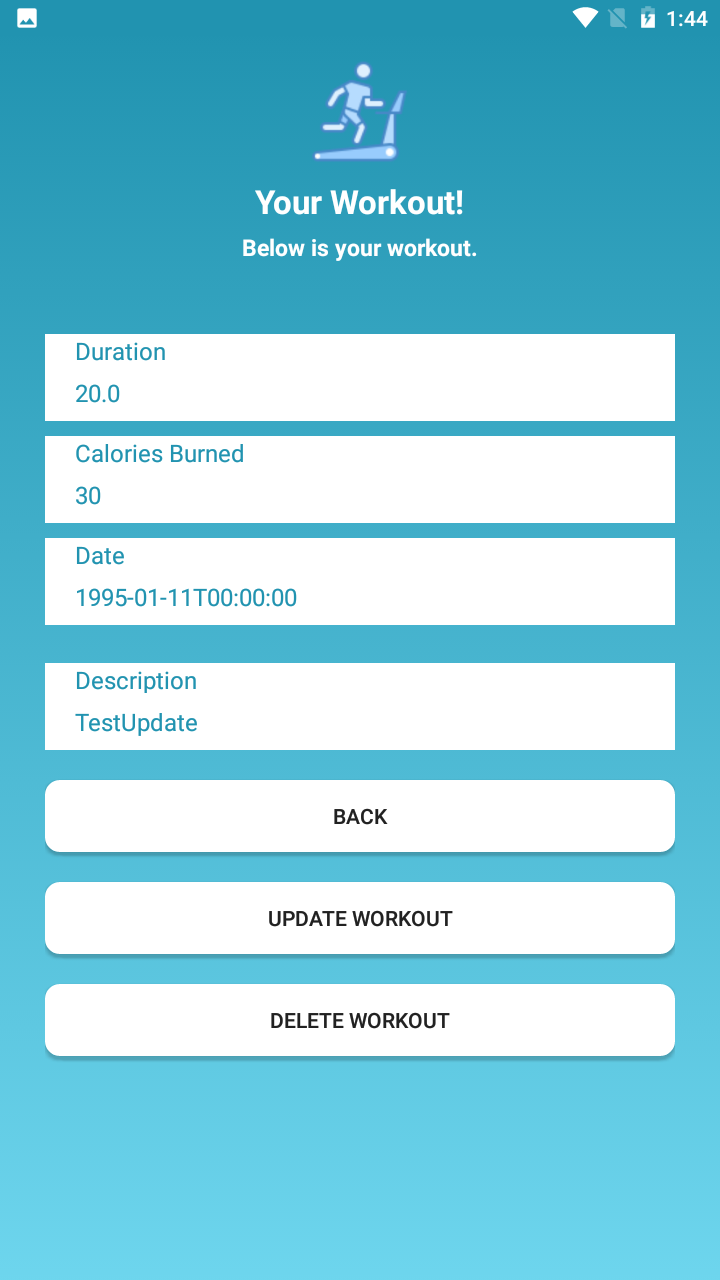
Add new workout test:

Shows that a new workout with the desc Test has been added. Using RecyclerView to display



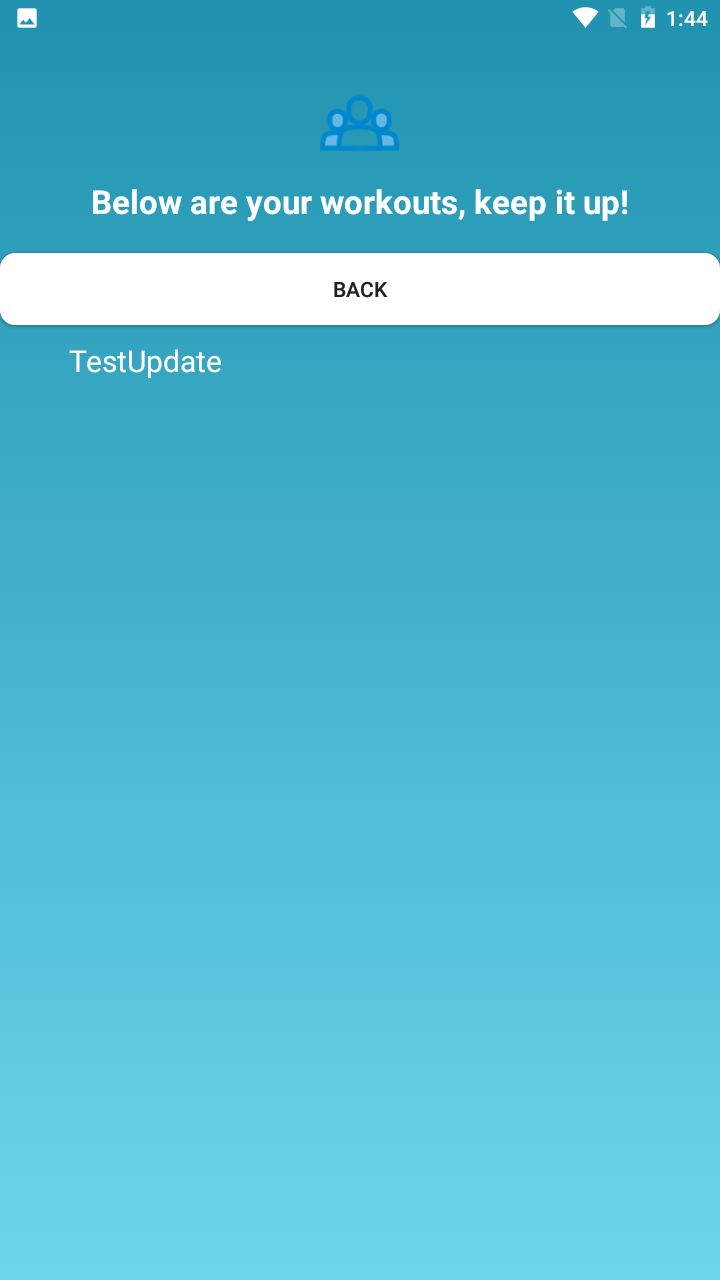
Edit this workout:

Fields are editable to allow the user to update their workout.

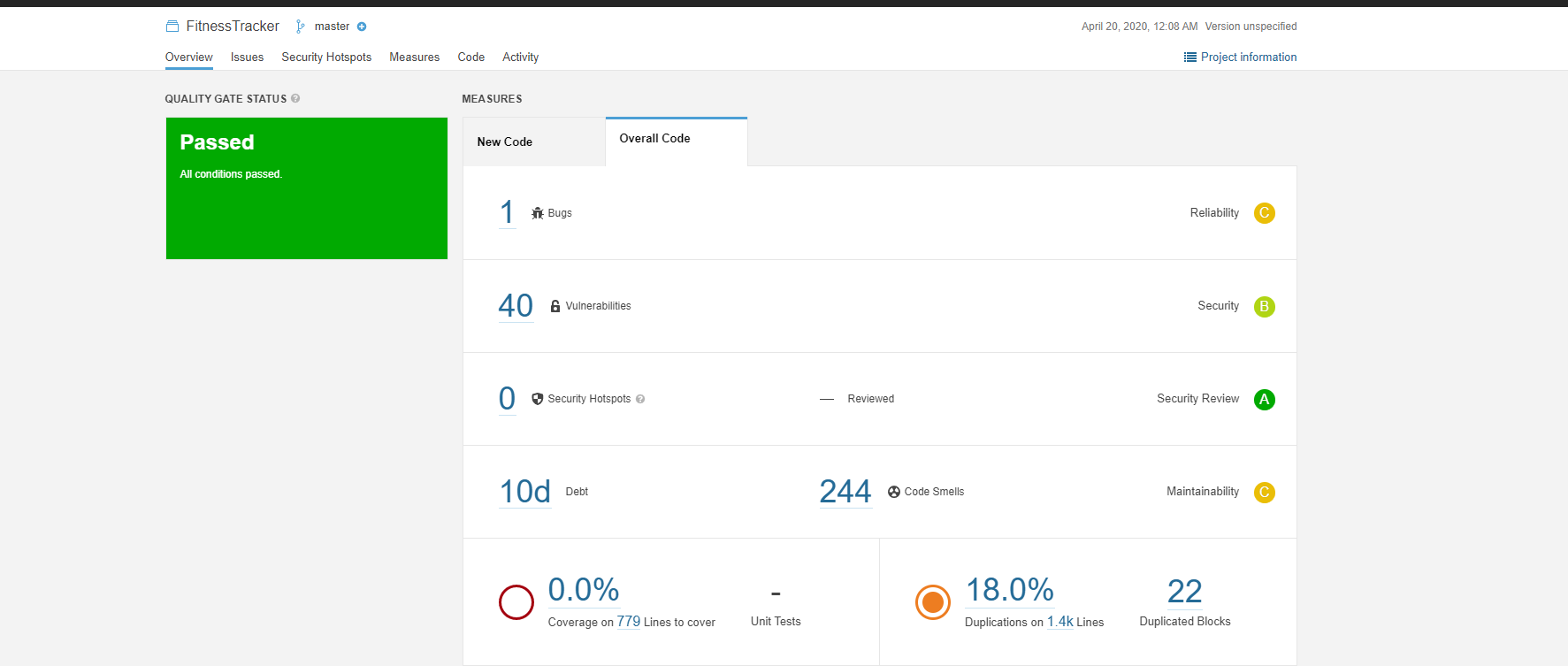


Updated details added:

Verification that this information has been updated: Test has became TestUpdate. - .put method



Testing – SonarQube



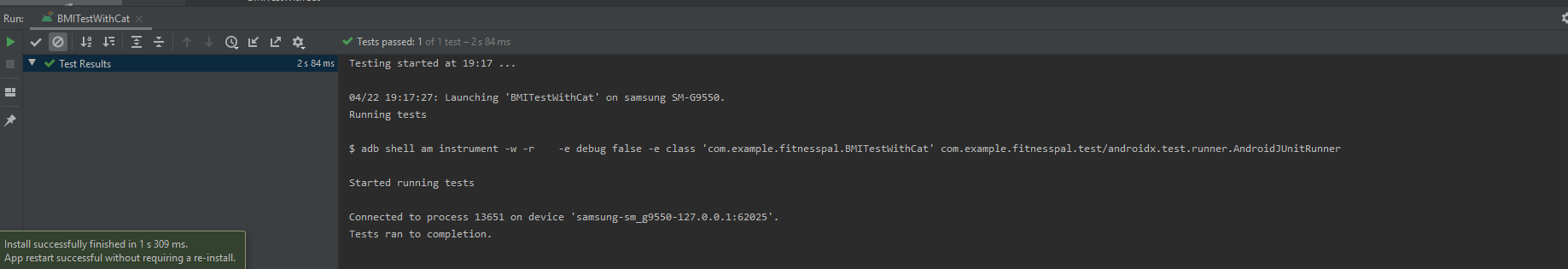
Testing – Espresso

BMR Test – Create User and View BMR is correct Value.



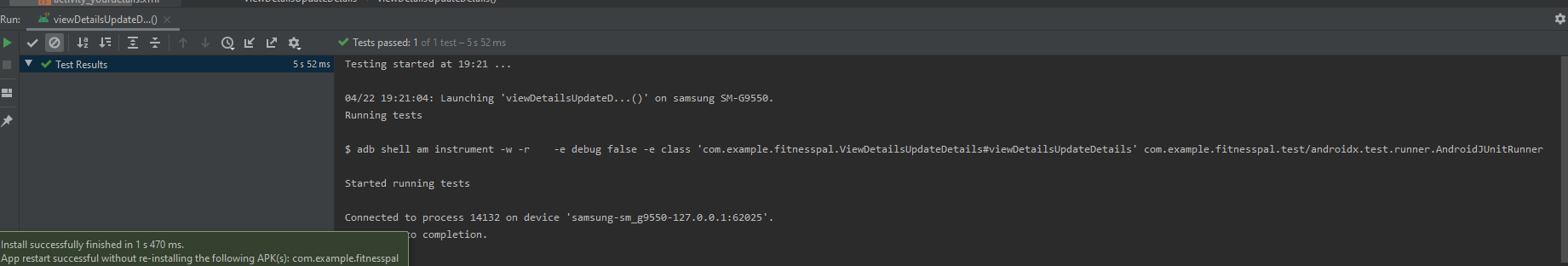
BMI

Test – Create User and View BMI is correct Value and correct category.



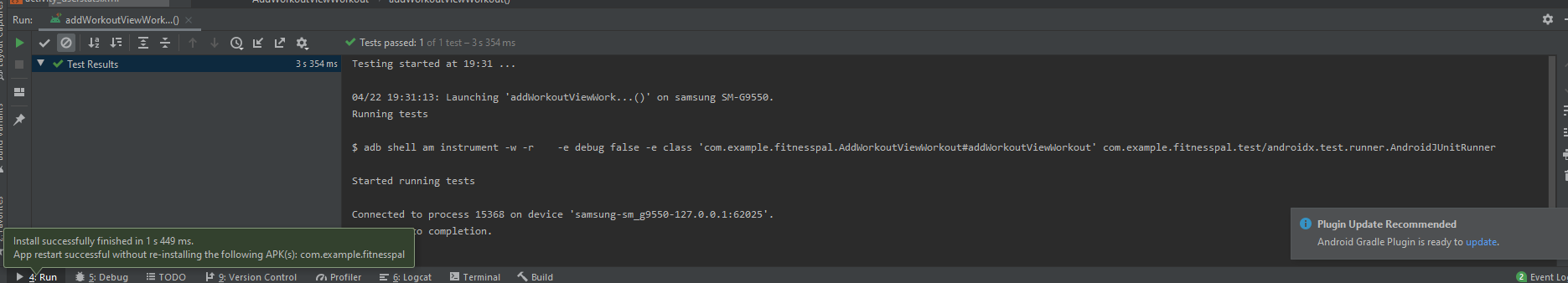
Details

Test – Create a user, view First Name, update First Name assert new name is set.



Workout

Test – Add Workout, View Newly Created Workout



View Workout

Test – Add Workout, View Longest workout in userstats

