Unit testing

Due to Firebase being asynchronous, part of unit testing had to be done through other means rather than unit tests.

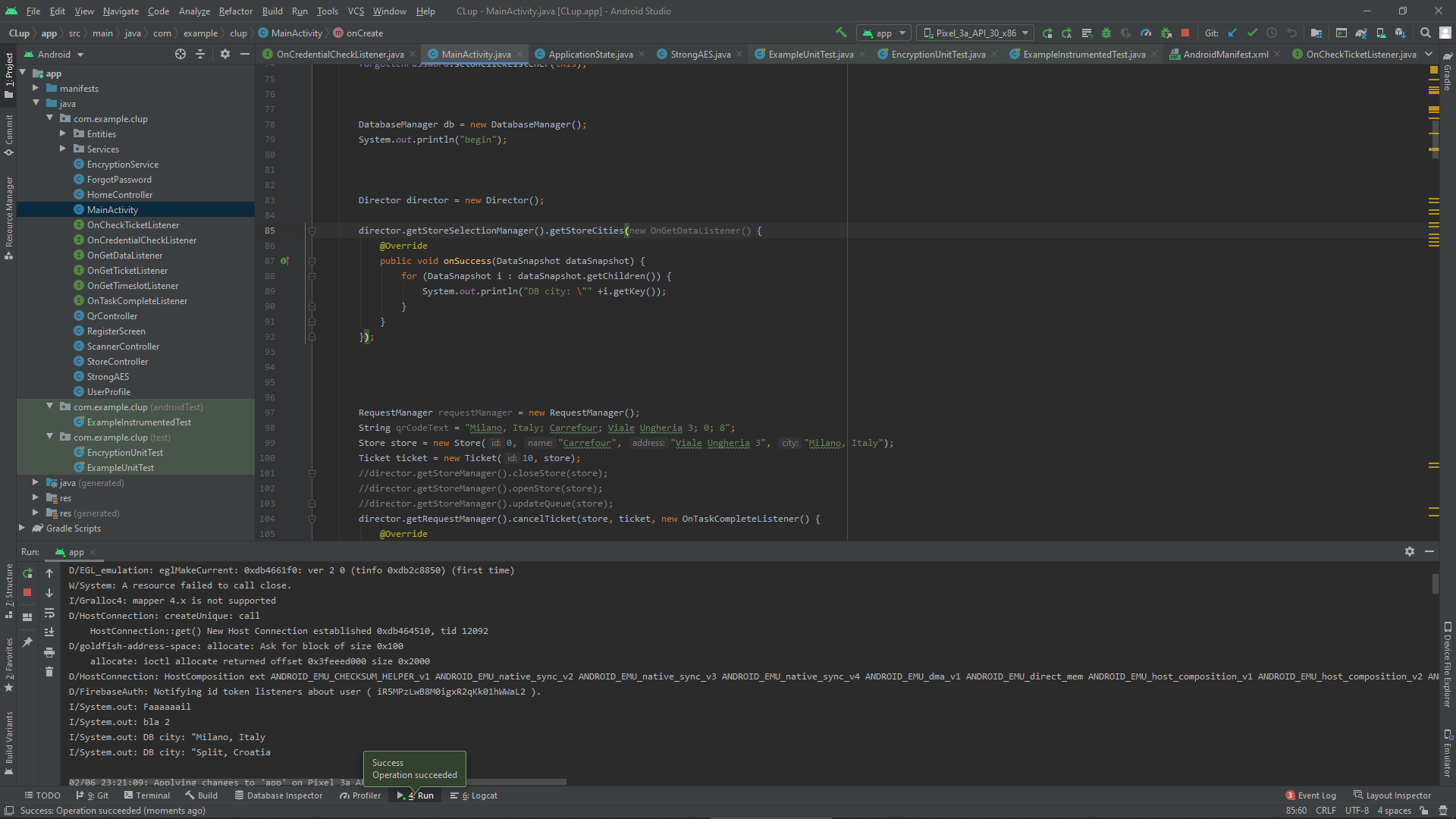
AES encryption, for example, was tested using Android Studio’s native Java Unit Tests

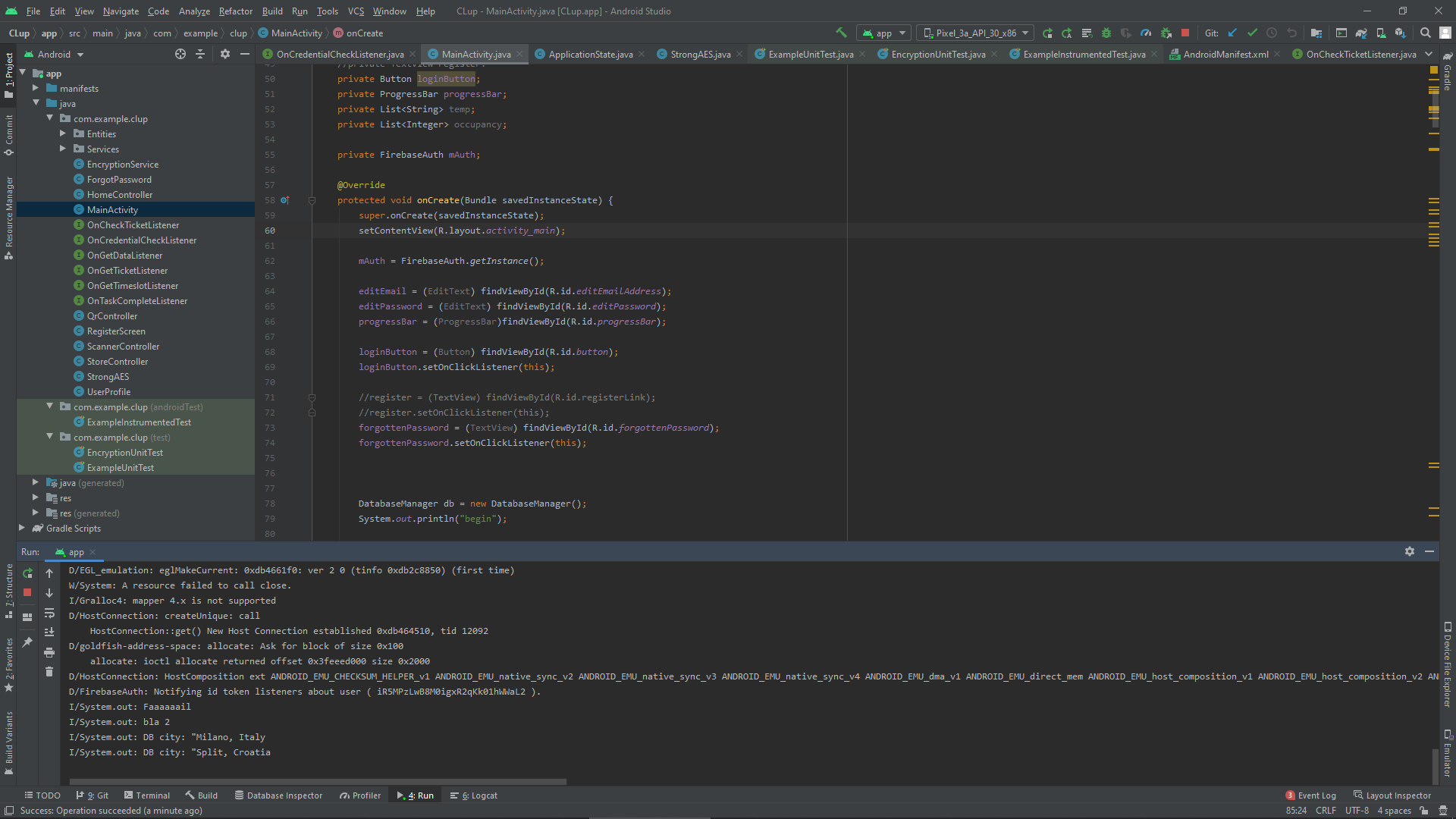
­Text

Description automatically generated

Firebase data fetching tests:

Fetching store cities:

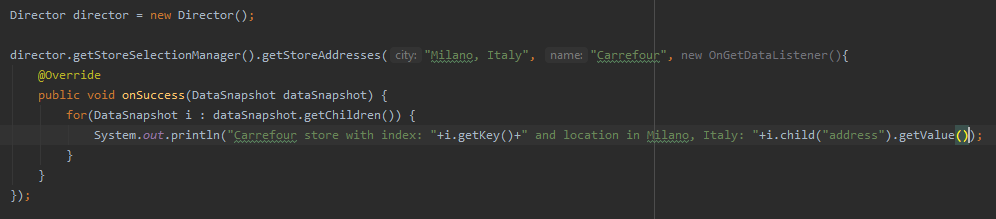




Fetching store chains in a city:



Fetching store chain store addresses:

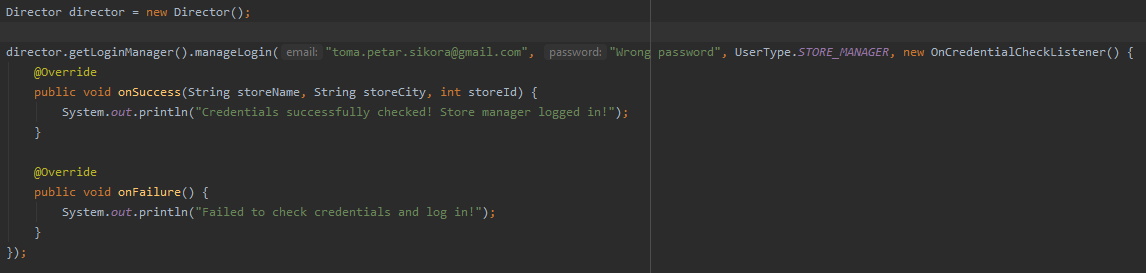


Checking credentials for a store manager login:

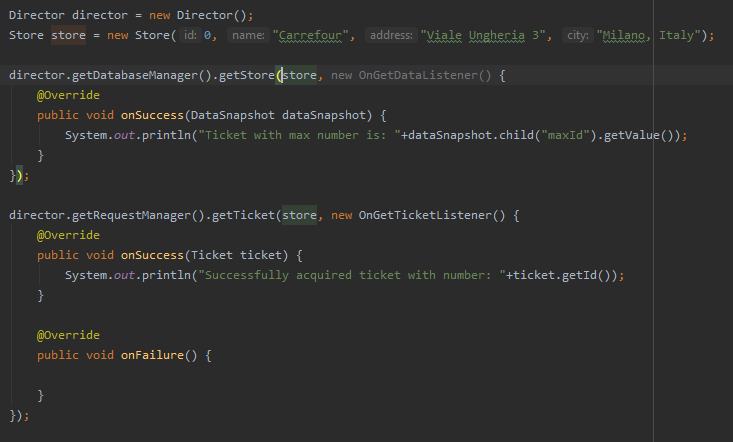
Successful login:



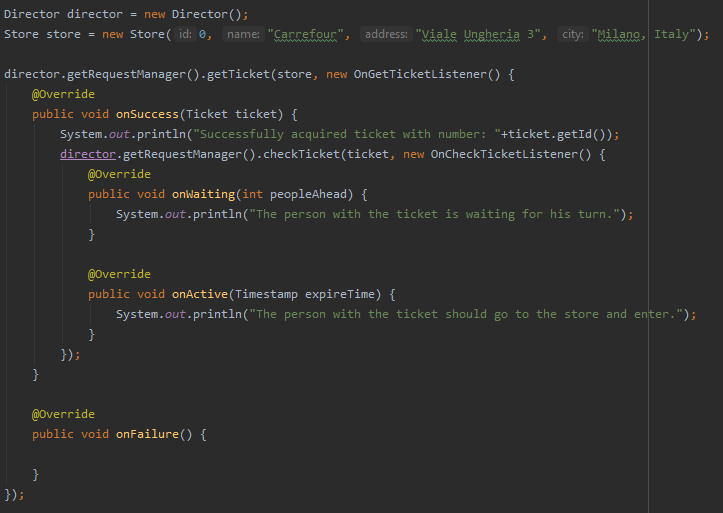
Failed login:



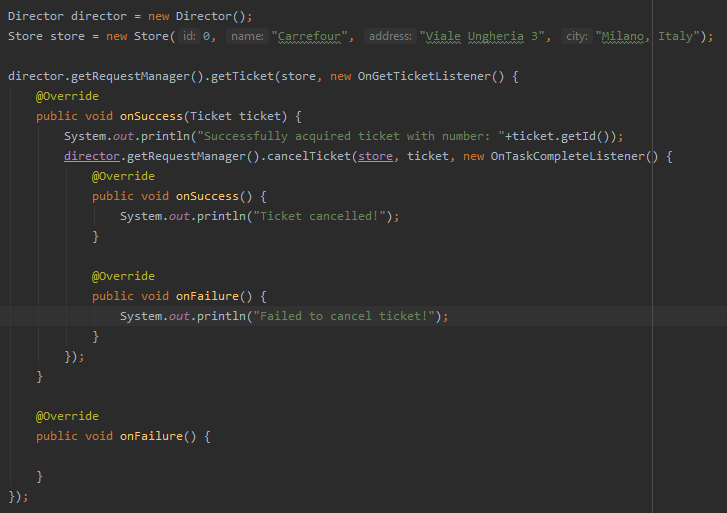
Acquiring ticket to get in the virtual line:



Checking the state of the ticket:



Cancelling the ticket:



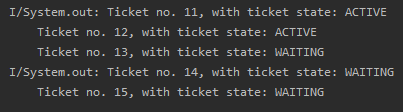
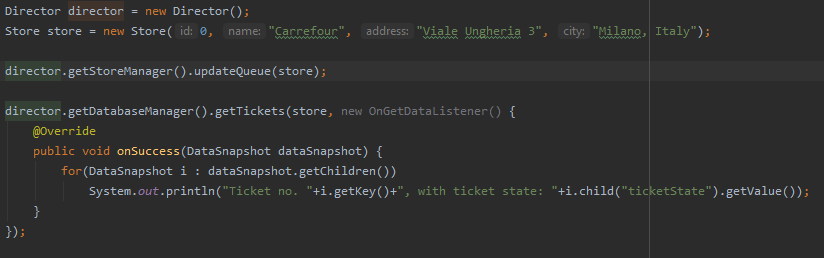
Checking ticket logic:

**Test 1**

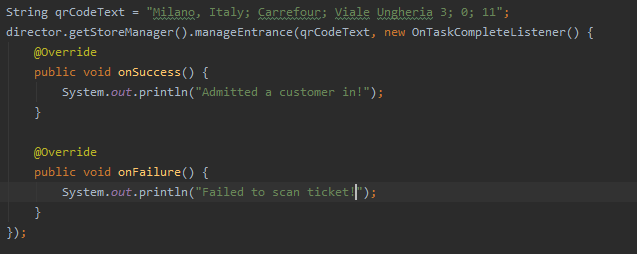
Beginning state:

Maximum store occupancy is 2, current occupancy 0.

2 active tickets with numbers 11 and 12, with expire time 11:01 (5 mins after activation), 3 waiting tickets with numbers 13, 14, and 15

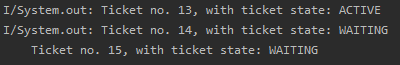
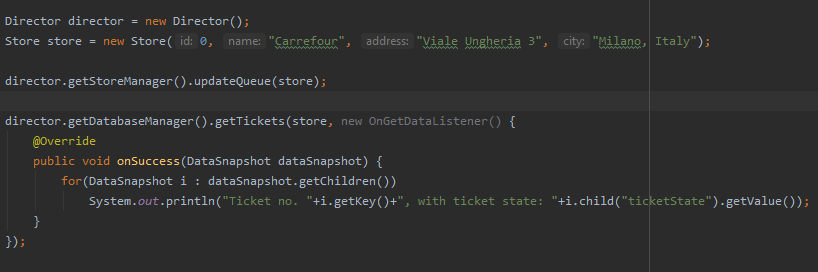


Customer with ticket number 11 scans his ticket and the store manager lets him in.

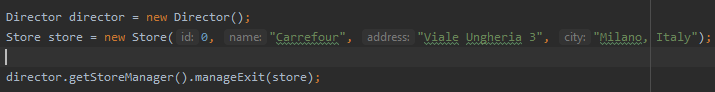


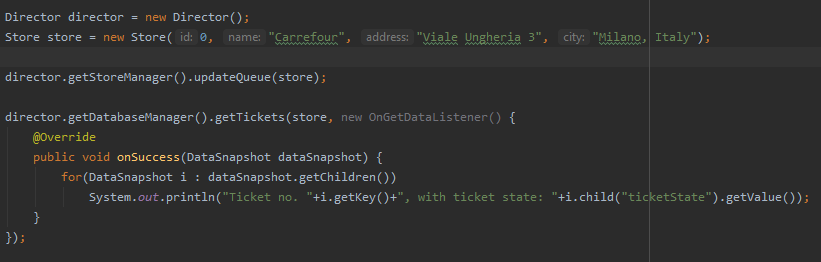
After a couple of minutes, store manager automatically updates the queue:

In the meantime, ticket with number 12 has expired, and the ticket with number 13 is activated.



When a customer exits, the occupancy changes and another customers ticket is activated:







**Test 2**

Beginning state:

Maximum store occupancy is 2, tickets with number 1 and 2 are activated, ticket with number 3 is waiting.

Person with ticket number 3 tries to enter the store before his ticket is activated:

