Summary:

For our project we chose an Android application. As recommended we chose an application which we felt had limited test coverage, but on exploration turned out that it did not have unit test coverage at all. This was something we discovered during the learning process. We added a unit test to an entire class using Mockito, JUnit and test runner, the frameworks recommended by Android on their site <https://d.android.com/testing> . We learnt the way to create and write tests and implemented it for class.

Experiences and Learnings:

In this process we learnt a few things as listed below

* Using Gradle and build dependency for android modules
* Difference between instrumented tests ( can be thought of as UI tests or functional tests for mobile) and unit tests ( running on machine JVM)
* How to create a testing module in Android
* Visualizing code coverage from the test suite
* Writing tests
* Identifying the testability of a class

Challenges we faced were also on similar lines. We realized that code is not always cleanly written. That frameworks like Android have a lot of static methods which when added as dependencies become hard to Mock, thereby hard to create state. We learnt that writing code that is testable, code which follows observability,controllability and smallness is important and how that can help write tests.

With the above experiences we are looking forward to improving the coverage of this Android project during the course of this class.