**Tools and components which I have used for this application:**

Android Studio : 3.6.1

Kotlin: 1.3.61

MVVM Clean Architecture

Repository Pattern

Live Data  
Retrofit – 2.6.0

okhttp3 – 4.2.2

Glide – 4.11.0

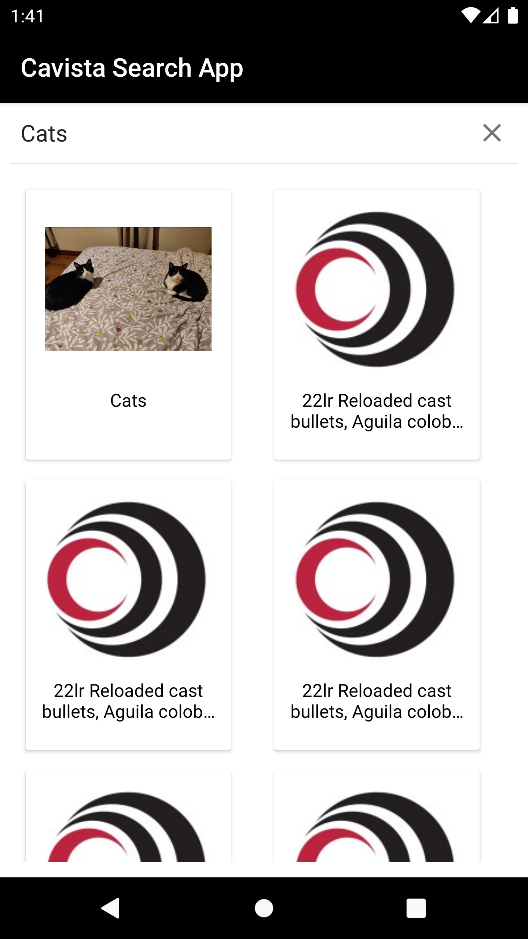
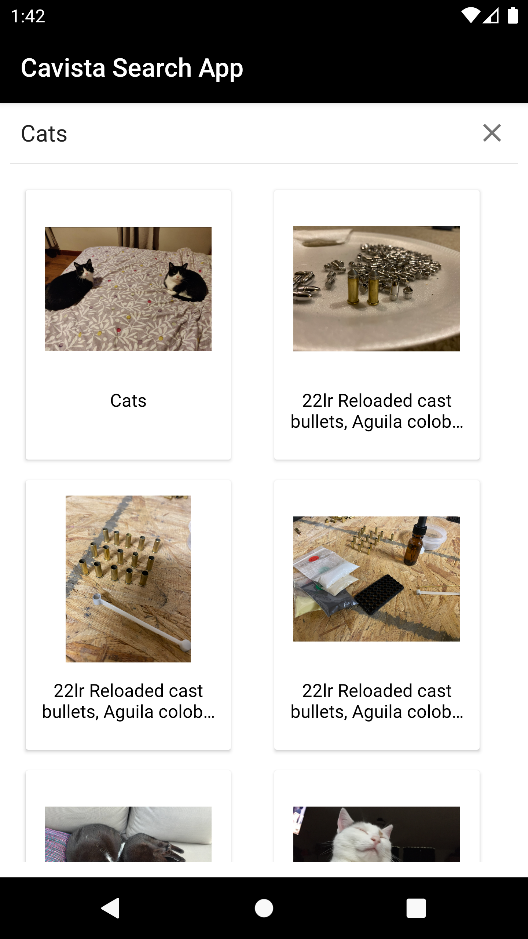
Kotlin Coroutines – 1.3.0

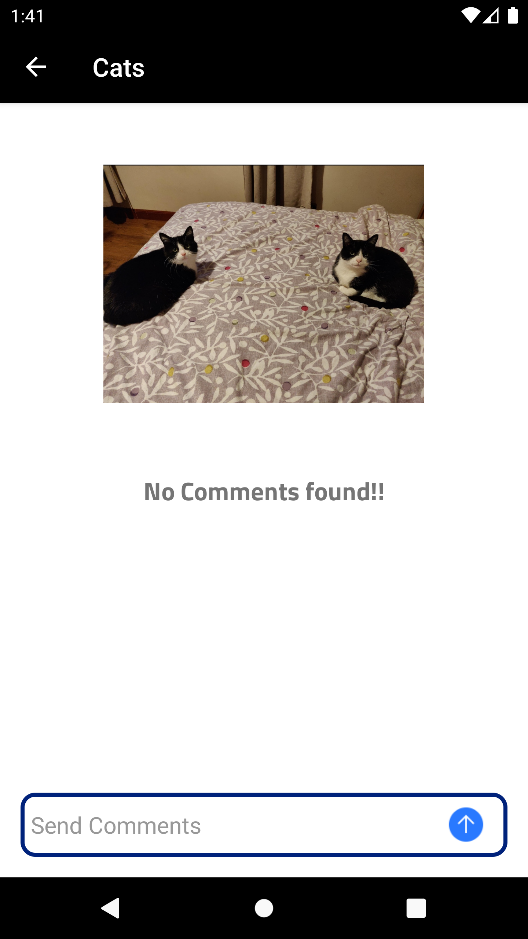
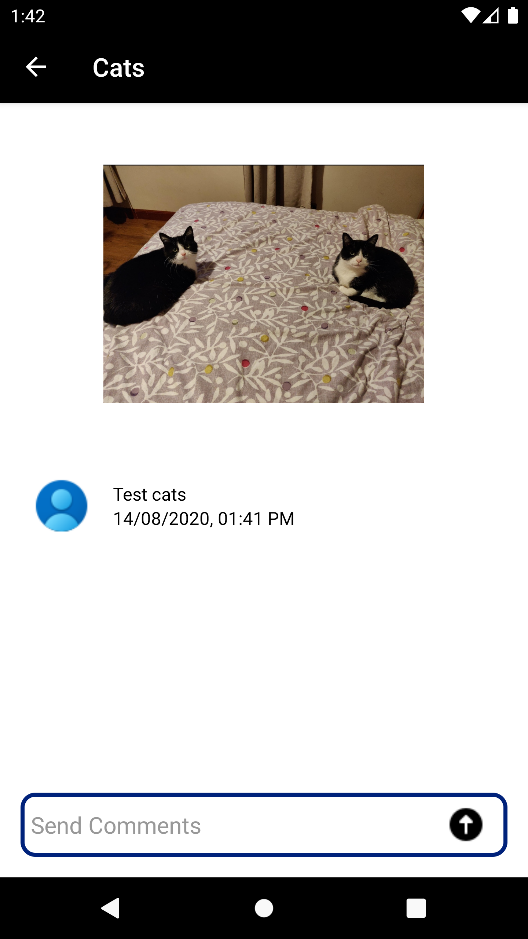
RxJava - 2.2.0

Dagger 2 – 2.24

For more details, please have a look dependencies.gradle and build.gradle

**Search screen:**

**Architecture:**

I have used MVVM Clean design pattern, which will be generic to the application irrespective of how many API calls we have across the app. This pattern will be taking care of everything.

Coroutines and IO dispatcher will be taking care of searching task running tasks.

Completed Task

* 250ms debounce in the search field : Added Custom debounce class ”**DebounceQueryTextListener**”
* Design to accommodate phone and tablet : **Yes**
* Any use of RxAndroid/RxJava/Android architecture components : **Yes**
* Use of dependency injections : **Yes(Dagger 2)**
* Use of Constraint layouts : **Yes**
* Writing some tests: **Yes**