CS205 Assignment 3 - Documentation

Team 8: Edwin, Ezekiel, Ke Xin, Jolene Loh, Wei Jie
Github: https://github.com/blankshiro/CS205-Assignment3

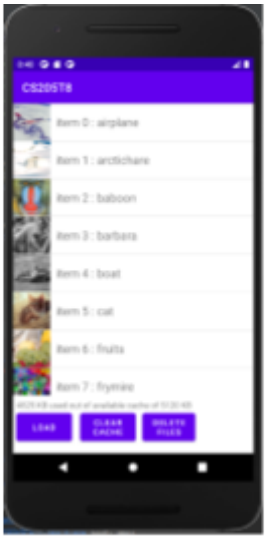

Figure 1: UI of Application

Cache Design

Cache Size: 5mb
Replacement Algorithm: Least Recently Used (LRU)
Number of caches: 1

We have a single cache that is implemented using the *java.util.LinkedHashMap* (sorted hash map). The images are stored as a bitmap using the hashcode of the url as the key. The *put(K key, V value)* is overridden with additional conditions to ensure that the cache size limit is maintained before adding the new image. Since keys are returned from least recently accessed to most recently accessed, the first key will correspond to the eldest entry to be removed

Overview of Implementation

We have also created an additional class - *DiskLoade*r to handle the retrieval of images from the local storage. To handle the deletion of images, we added a *deleteListener* in *MainActivity.java* which will be triggered upon pressing the “Delete Images” button. To handle the clearing of the cache, we have a *clearCacheListener* which will call the *Cacher clear()* method which will remove all entries from the cache.

For the overall implementation of the loading of images, it is handled by the *DownloadTask* in the *Loader* class. It first attempts to load the target image from the cache, then the disk. If both operations return null, it will download the image from the url and save it to the cache and disk using background threads.

We have a responsive UI, with the size of the list view automatically matching the size of the phone screen.

2 UML Class Diagram