

Project Phase 2

The Uses of Machine Learning Technique to Improve Software Quality and Testing

SE433
DePaul University

Group 7

Jenney Chang
Amy Aumpansub
Darrell Lane

jdzchang@gmail.com
amy.aump@gmail.com
darrell.lane25@gmail.com

February 25, 2021

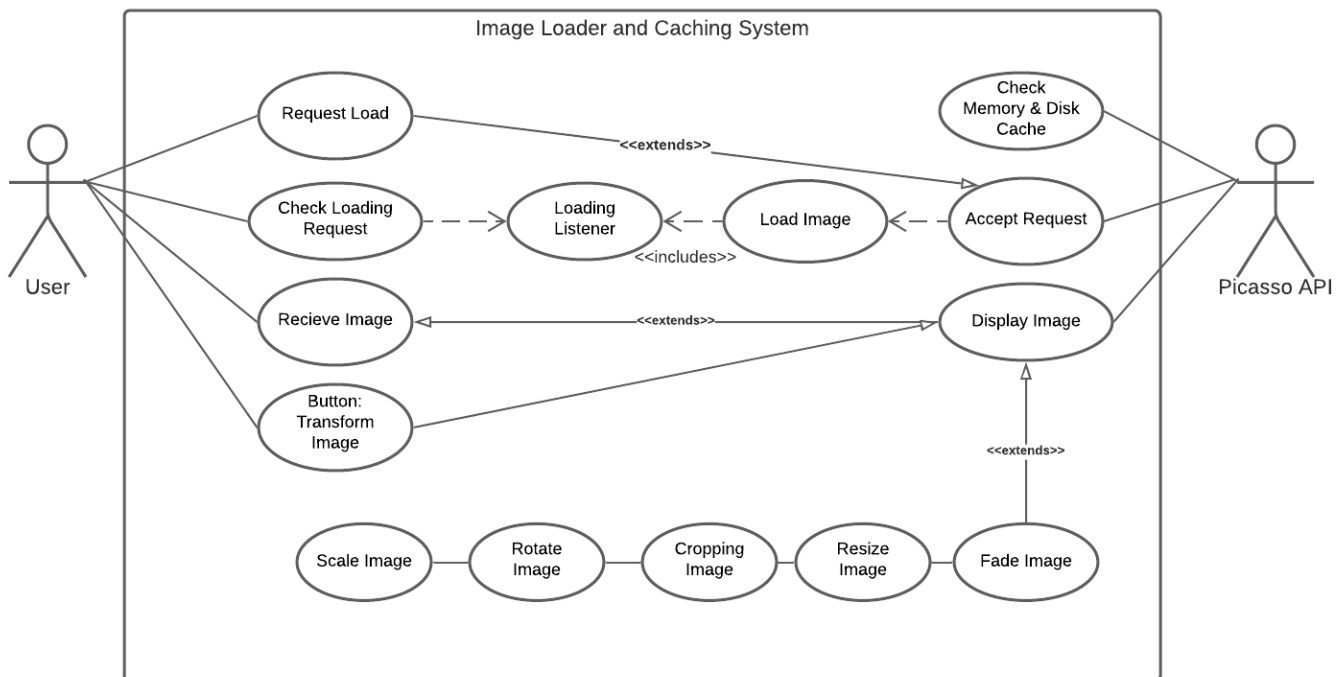
Part A: Functional/Non-functional requirements

Android Image loaders Photo View, Picasso, and Android Universal Image Loader were chosen for quality analysis and improvement. To optimize the quality improvement process, K-mean clustering models were built to predict quality, discover patterns, and partition all classes of each Image Loader. Classes were placed into 3 clusters: low-, medium-, and high-quality based on the historical CK-Metric values extracted from the Understand tool. Using K-mean clustering enables us to focus on low-quality classes in refactoring and testing, which significantly improves speed and effectiveness.

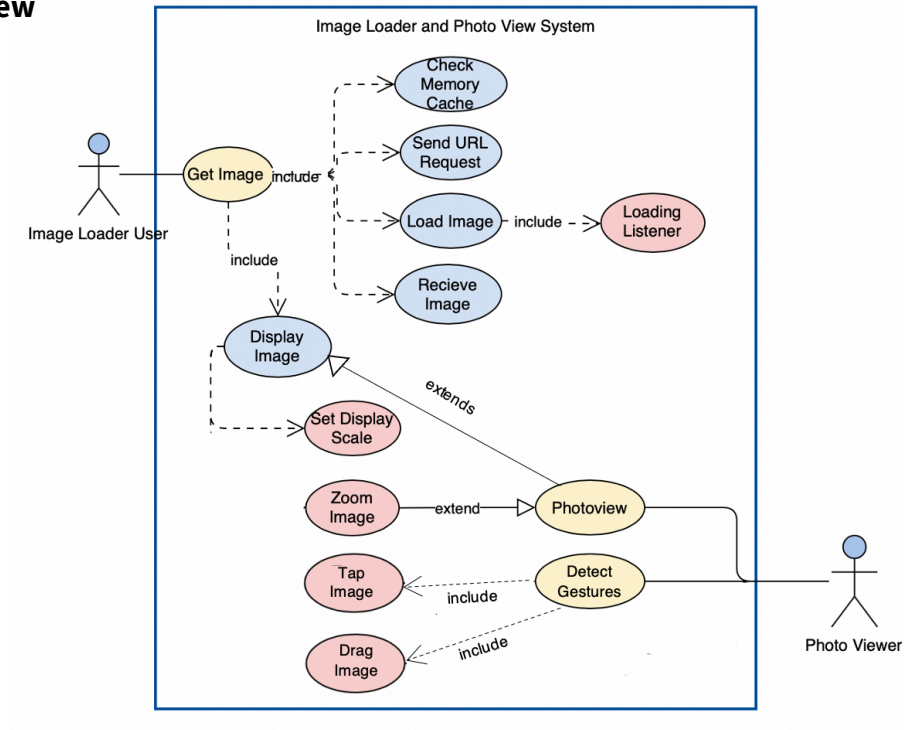
The main focus is to develop a strong predictive model and efficiently improve quality. We need to evaluate the structure and implementation of these image loaders. Examine how classes and functions are related and interact with each other. Image loaders have often been utilized to extend the functionality of Android applications, we will need to explore the user's interaction with the system functionality and examine the sequence of exchange between objects -as shown in the following diagrams. Combining the machine learning approach with a deep understanding of the system will enable us to accurately detect system flaws and improve its quality during refactoring and testing.

Part B: Use case diagrams

1. Picasso



2. Photo View

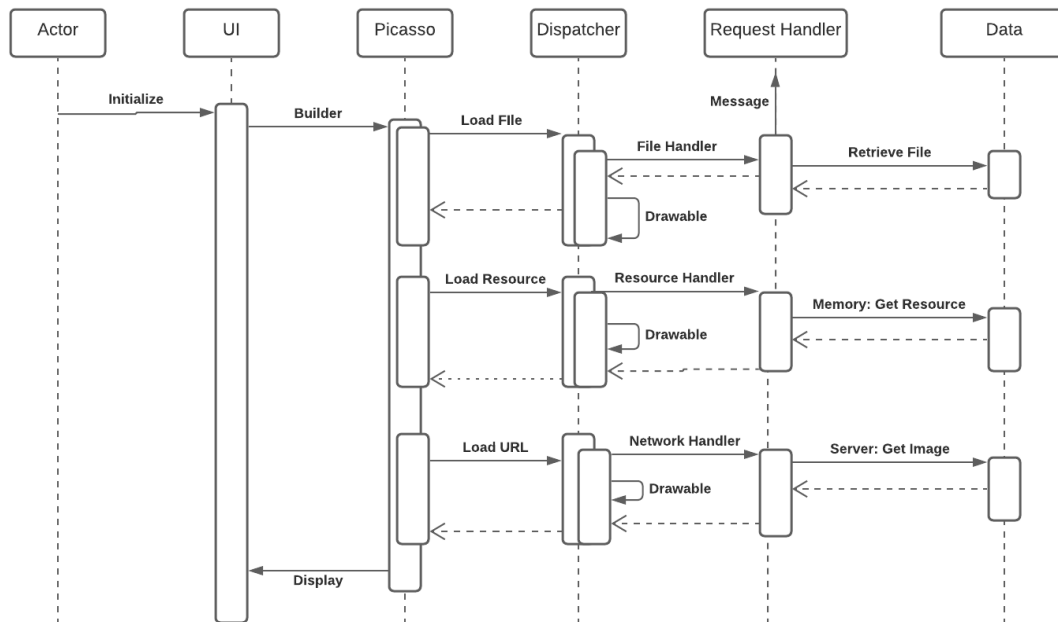


3. Android Universal Image Loader

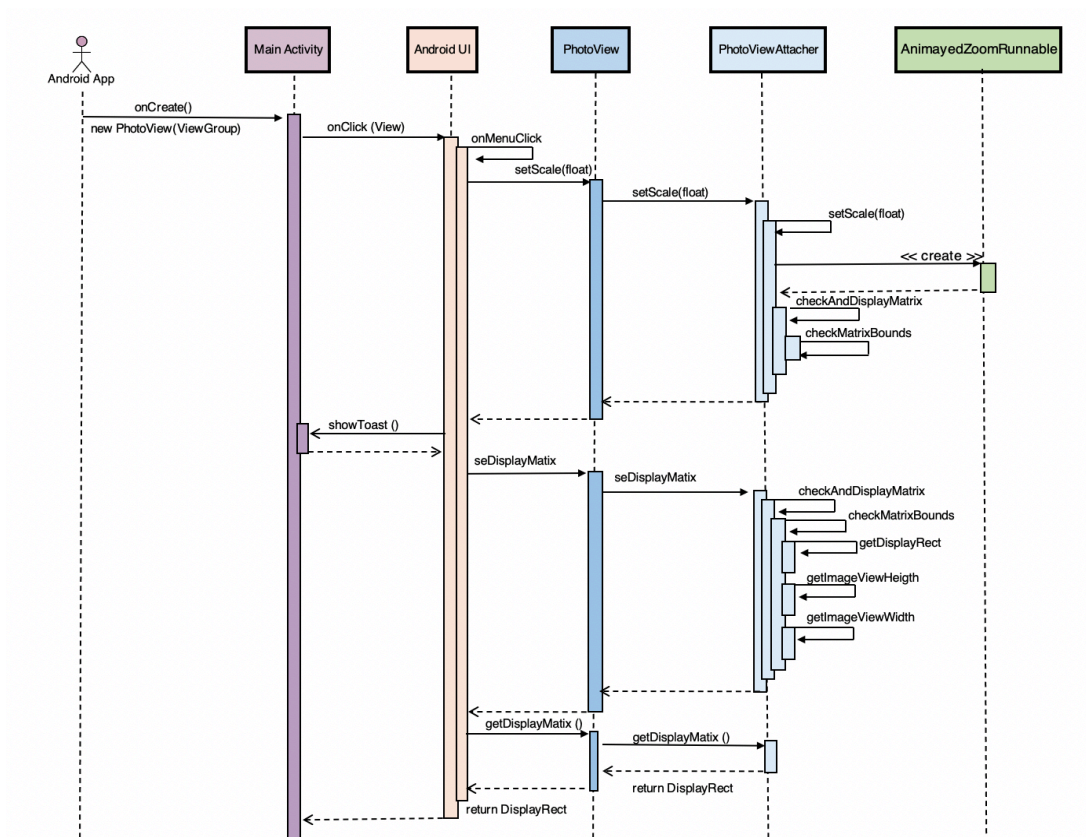


Part C: Sequence diagrams (for only main scenarios)

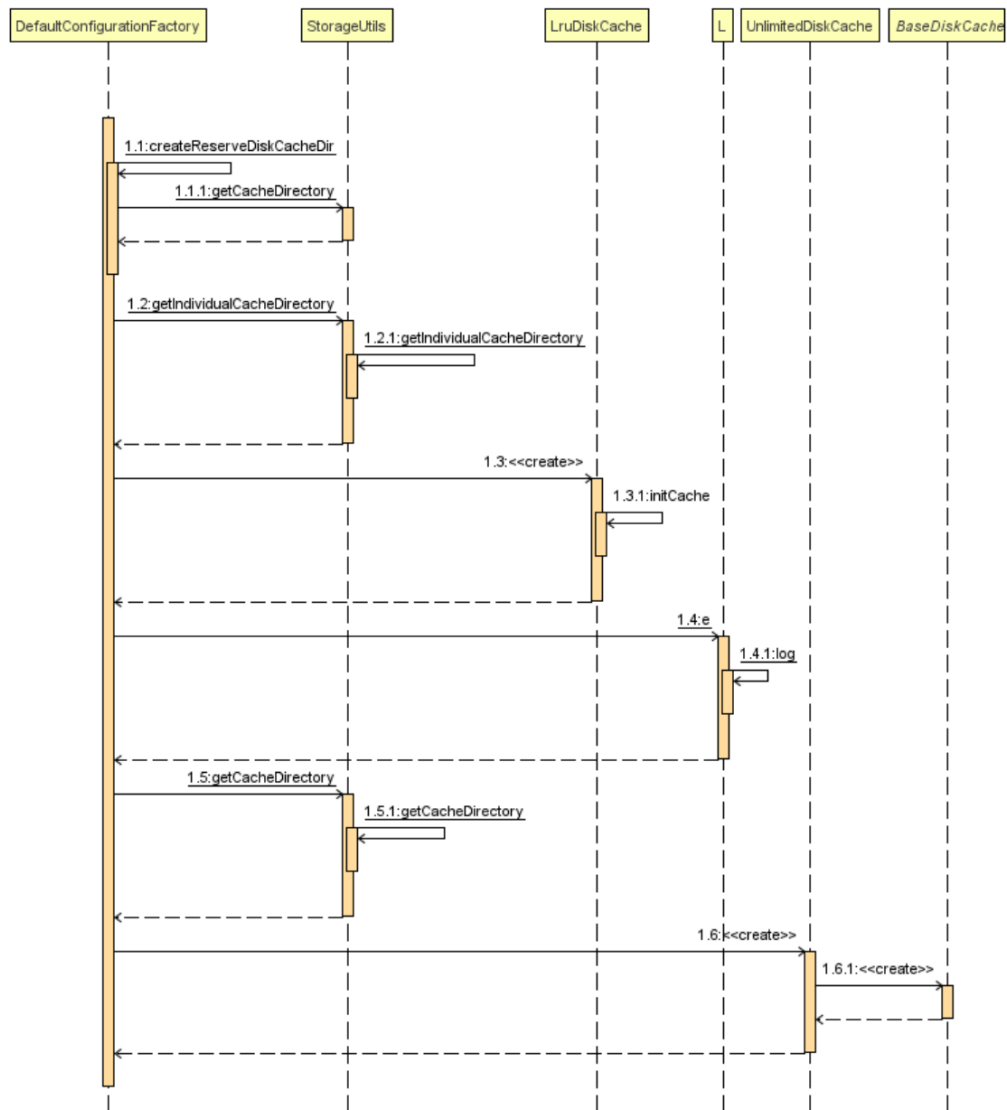
1. Picasso



2. Photo View



3. Android Universal Image Loader



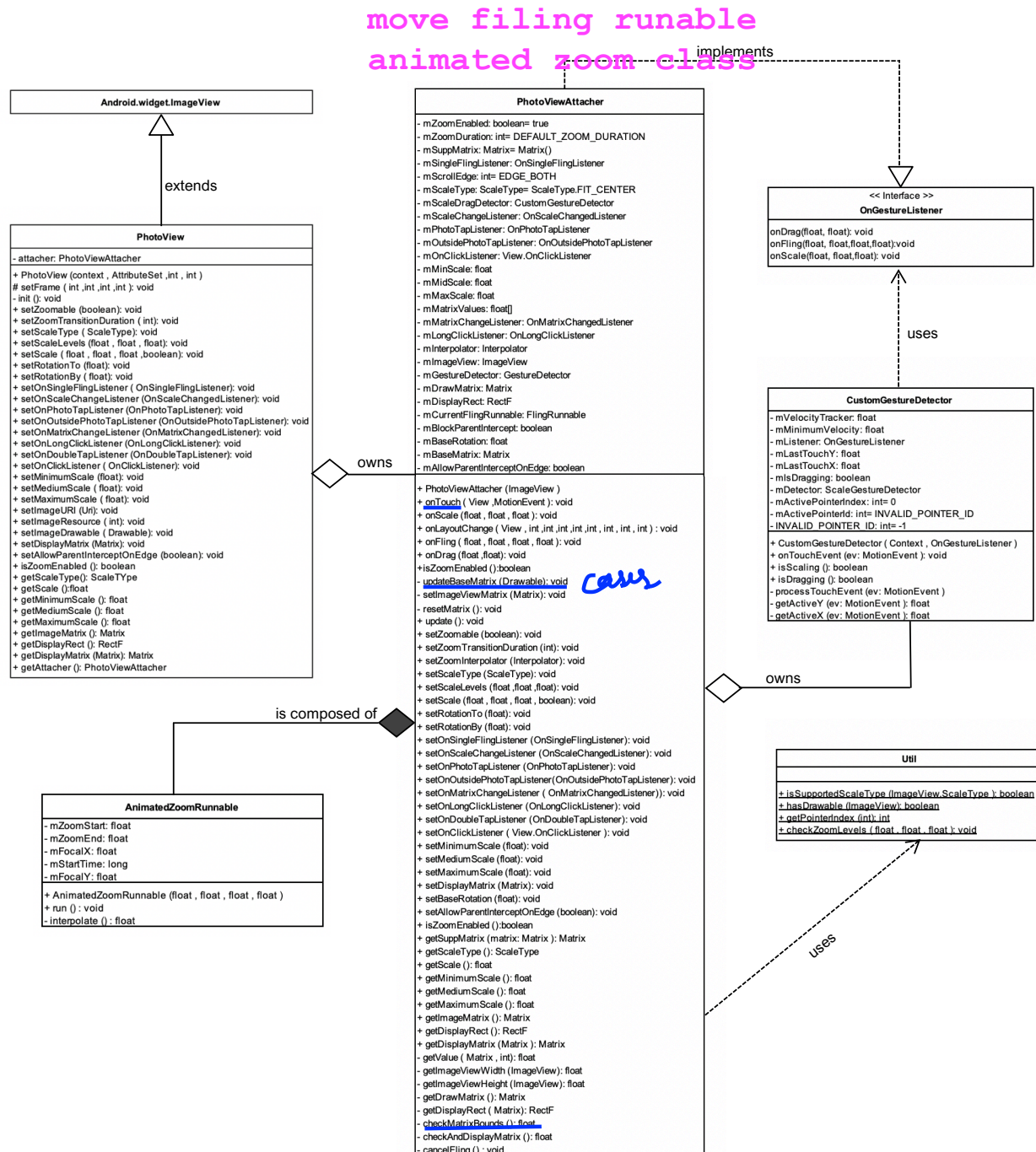
Part D: UML Class diagrams

1. Picasso

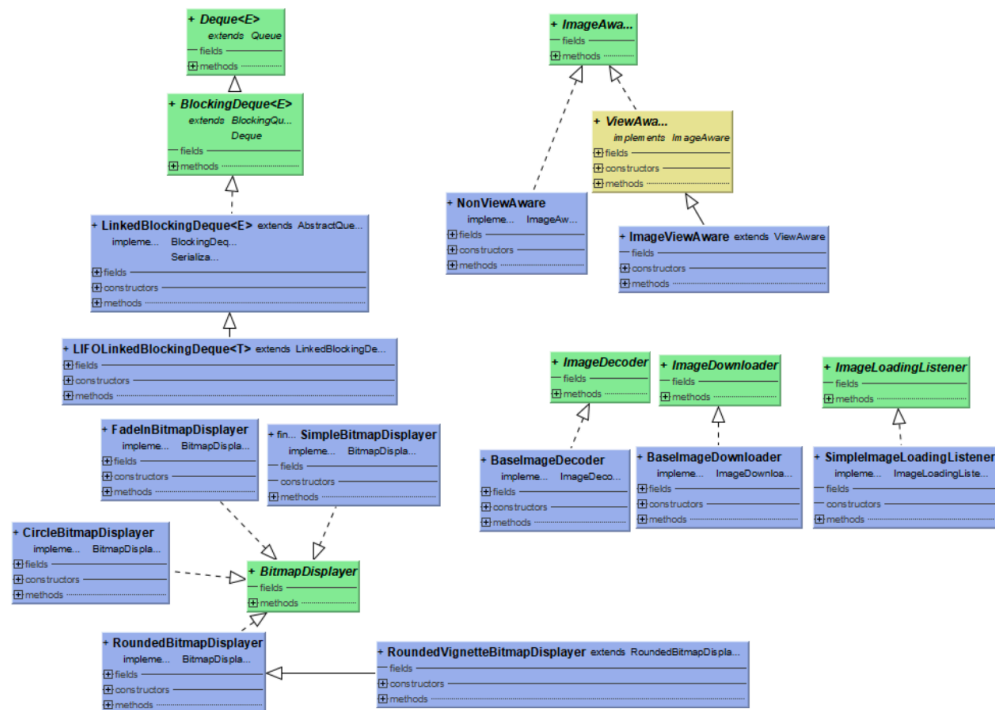


2. Photo View

Photo View is aimed to extend the implementation of Image View for Android that supports zooming, by various touch gestures, so this UML class diagram focuses on the classes related to the main functionalities and on low-quality classes predicted by our K-mean clustering model.



3. Android Universal Image Loader



Part E: Tasks done and Task remaining

Our project contains 3-step method:

1. Software quality assessment using CK metrics
2. Quality Prediction and Clustering using K-Means Clustering (Machine Learning)
3. Software Quality Improvement on low-quality classes using refactoring and testing

Tasks done: We have successfully finished all tasks for step 1 and step 2 as follows:

- Extracted the CK metric of Image Loaders using Understand Tools
- Created graphs for comparing each metric across 3 image loader and conducted a quality assessment based on metric values
- Preprocessed metric values data by applying z-score normalization and principal component analysis to reduce the dimensional space of features
- Built 3 K-means clustering models for each image loader to predict quality of each class and use models to partition all classes of each Image Loader into 3 clusters: low-, medium-, high-quality clusters based on the historical metric values

Task remaining: We will implement the remaining tasks for step 3 as follows:

- Utilize UML, sequences, and use-case diagrams to detect flaws of low-quality classes
- Refactor low-quality classes to improve their quality
- Write test programs for the refactored classes to ensure their functions implement as expected