In the Utils.java class, we create two functions generateData and generateInitialCentroids to create a number of randomized centroids and data points. All these data is stored under particular folders specified by the users and the local files will be cached to HDFS later. The data points are partitioned into multiple chunks assigned with different job id.

In the KmeansMapCollective.java class, job configuration is set with specified parameters. After that each chunk of data will be fed into a map task by a key-value pair <job\_id, file\_directory>. In the KmeansMapper.java, we accomplish the computation of k-mean algorithm. The centroids and masterId are broadcasted to all individual map tasks. Each data point is assigned to a centroid by using computeMin to get the closest Euclidean distance. Then the centroid for each data point will be updated basing on the previous information. The output of each data chunk will be collected and reduced by regroup and allgather functions. The whole algorithm will be executed for a couple of iteration until the centroids converge.