

# BD-AE1 – 2147392N (ANDREI-MIHAI NICOLAE)

## 1. DESIGN DECISIONS

The project structure is rather straightforward, the classes having the following reasoning behind:

- `MyReducer.java` – will add to a `HashMap` the number of article revisions. In the end, after the process has finished, it will sort the map and select the top `k` pages in the given interval with the highest number of modifications.
- `MyMapper.java` – splits the current record on space as a delimiter and checks if the date is between the 2 command-line provided dates. If so, it gets the article id and writes it to the context, the value being 1.
- `MyCombiner.java` – this will merge all the revisions with the same article id, summing up the value and passing it towards the reducer.
- `MyInputFormat.java` – it will simply separate the records by using the “`\n\n`” delimiter.

## 2. SCALABILITY

I have run the Driver multiple times with various configurations and I did find out that the runtime is not fully dependent on the value of `k`. Even more, the speed can be faster when `k`'s value is higher. However, the factor that I found the most decisive in the speed of the program was the network as its traffic influenced drastically the overall runtime.

## 3. PERFORMANCE

Command: `$ java-run.sh Driver 2006-01-01T12:00:00Z 2008-01-01T12:00:00Z 100`

	Query processing time	Bytes read from HDFS	Bytes transferred over network
Run 1	10m 51s	31274123655	119017508
Run 2	8m 41s	31274123655	119017508
Run 3	7m 33s	31274123655	119017508
Run 4	7m 20s	31274123655	119017508
Mean	Run 1	31274123655	119017508
Standard Deviation	Run 1	0	0