Log Aggregation in Scala Work Sample Question

Process a log file to compute average time spent by each user on an app.

Details

You are given a large log file which stores user interactions with an application. The entries in the log file follow the following schema:

```
{userId, timestamp, actionType}
```

where actionType is one of two possible values: `[open, close]`.

Constraints:

1. The solution must be implemented in Scala
2. The log file is too big to fit in memory on one machine. Also assume that the aggregated data doesn't fit into memory.
3. Your code has to be able to run on a single machine.
4. Do not use an out-of-the box implementation of mapreduce or 3rd party database; don't assume you have a Hadoop or Spark or other distributed computing framework.
5. There can be multiple entries of each actionType for each user, and there might be missing entries in the log file. So a user might be missing a `close` record between two `open` records or vice versa.
6. Timestamps will come in strictly ascending order.

```
u1, t1, open
u2, t2, open
u3, t3, open
u1, t4, close
u4, t5, open
u1, t6, open
u2, t7, close
u1, t9, open
u3, t10, close
u4, t11, close
u1, t12, close
```

There is a sample log file at the end of this document.

For this problem, you need to implement a class/classes that computes the average time spent by each user between `open` and `close`.
Keep in mind that there are missing entries for some users, so you will have to make a choice about how to handle these entries when making your calculations. Your code should follow a consistent policy with regards to how you make that choice.

The desired output for the solution should be `[{userId, timeSpent}, …. ]` for all the users in the log file.

1. Design the interface and implement your solution.
2. Write tests for your code.
3. Write a readme with instructions on how to setup, compile and run your code and tests.
4. Submit the package as zip file.

- **Bonus:** Implement multiple policies for handling missing entries and let the caller specify which policy to use.

This problem should take roughly **4 hours** to produce a tested, working code.

If there is anything unclear, please state your assumptions in comments and go from there. For any other questions, please email your recruiter.

Good luck!

Sample log file (comma-separated, text file)

```
1,1435456566,open
2,1435457643,open
3,1435458912,open
1,1435459567,close
4,1435460345,open
1,1435461234,open
2,1435462567,close
1,1435463456,open
3,1435464398,close
4,1435465122,close
1,1435466775,close
```