Test Documentation

ORFAP Organisation for all purposes

All

- **Code Review** Every change and commitment is reviewed due to our PR-Workflow on github from all team members. Combined with an active linter we made sure that all standards are met.

Backend

- **Unit Tests** For every implemented Java Method exists at least one unit test using JUnit4. The test will be run on every commit on the Github Repository through the CI Chain with `travis.ci`. The link to to the results is: `https://travis-ci.org/ORFAP/FAPBackend`
- **Code Coverage** Every test run on travis includes a code coverage check, which will be passed to `codecov.io`. The link to to the results is: `https://codecov.io/gh/ORFAP/FAPBackend`

CRAWLER

- **Unit Tests** JUnit tests are implemented which examine if the crawled data (arlines, markets, etc.) are valid. They furthermore assure if the components of the pipes and filter architecture work correctly. The link to to the results is: `https://travis-ci.org/ORFAP/FAPCrawler`
- **Code Coverage** Every test run on travis includes a code coverage check, which will be passed to `codecov.io`. The link to to the results is: `https://codecov.io/gh/ORFAP/FAPCrawler`

GUI

- **Unit Tests** were written for multiple javascript methods which control the workflow in the GUI. Only if all tests pass a new release is being deployed. `https://travis-ci.org/ORFAP/flight-analyzer`
- **Beta Test** The beta tests revealed some minor design flaws that got fixed immediately.