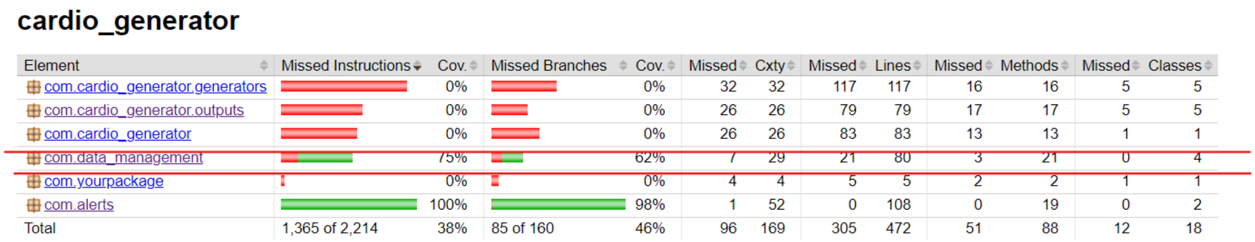
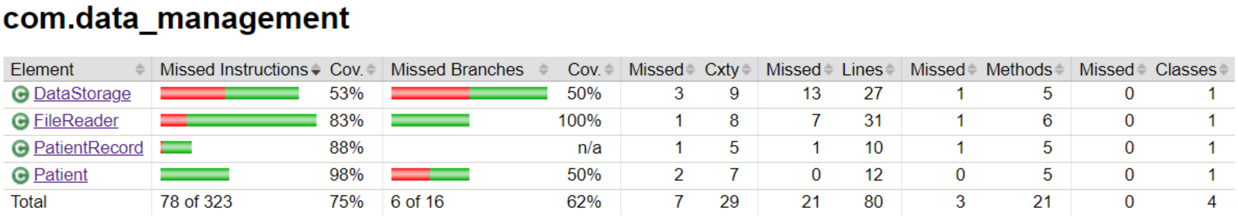
Code Coverage Documentation – Data management



cardio_generator

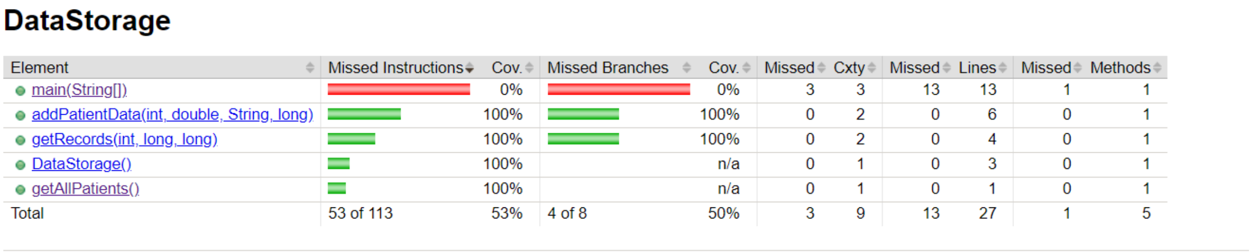
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.cardio_generator.generators		0%		0%	32	32	117	117	16	16	5	5
com.cardio_generator.outputs		0%		0%	26	26	79	79	17	17	5	5
com.cardio_generator		0%		0%	26	26	83	83	13	13	1	1
com.data_management		75%		62%	7	29	21	80	3	21	0	4
com.yourpackage		0%		0%	4	4	5	5	2	2	1	1
com.alerts		100%		98%	1	52	0	108	0	19	0	2
Total	1,365 of 2,214	38%	85 of 160	46%	96	169	305	472	51	88	12	18



com.data_management

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
DataStorage		53%		50%	3	9	13	27	1	5	0	1
FileReader		83%		100%	1	8	7	31	1	6	0	1
PatientRecord		88%		n/a	1	5	1	10	1	5	0	1
Patient		98%		50%	2	7	0	12	0	5	0	1
Total	78 of 323	75%	6 of 16	62%	7	29	21	80	3	21	0	4

- **DataStorage.java**

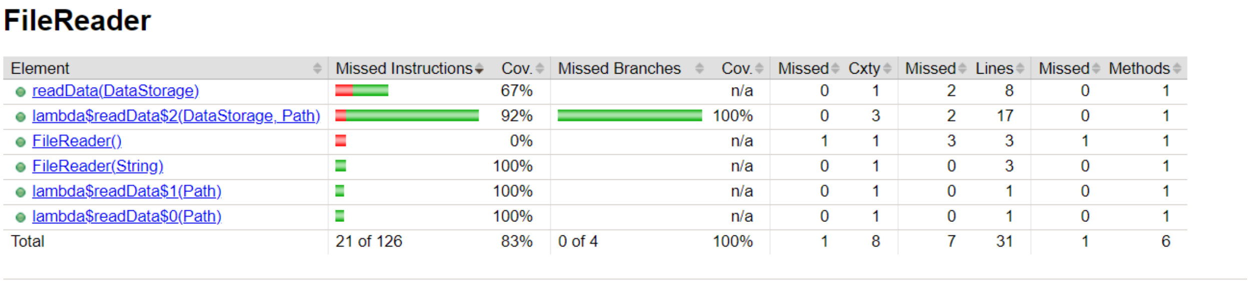


DataStorage

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
main(String[])		0%		0%	3	3	13	13	1	1
addPatientData(int, double, String, long)		100%		100%	0	2	0	6	0	1
getRecords(int, long, long)		100%		100%	0	2	0	4	0	1
DataStorage()		100%		n/a	0	1	0	3	0	1
getAllPatients()		100%		n/a	0	1	0	1	0	1
Total	53 of 113	53%	4 of 8	50%	3	9	13	27	1	5

The DataStorage.java class has been fully tested except for the main function, because it performs the role of starting the file rather than its methods

- **FileReader.java**

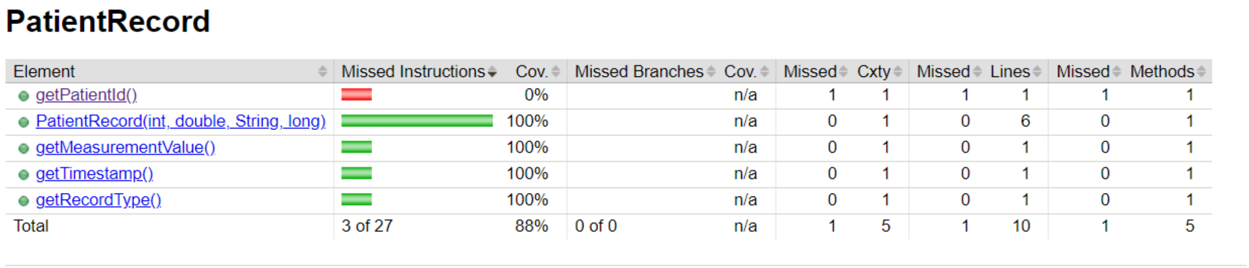


FileReader

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
readData(DataStorage)		67%		n/a	0	1	2	8	0	1
lambda$readData$2(DataStorage, Path)		92%		100%	0	3	2	17	0	1
FileReader()		0%		n/a	1	1	3	3	1	1
FileReader(String)		100%		n/a	0	1	0	3	0	1
lambda$readData$1(Path)		100%		n/a	0	1	0	1	0	1
lambda$readData$0(Path)		100%		n/a	0	1	0	1	0	1
Total	21 of 126	83%	0 of 4	100%	1	8	7	31	1	6

A new file that reads values from a file and writes this to DataStorage. Everything was tested successfully, except for the errors in the readData() method, because they are in the try-catch block, which throws an error when an error occurs.

- **PatientRecords.java**

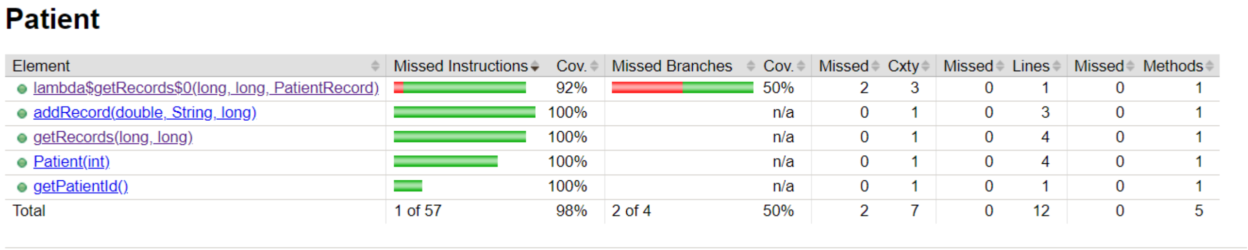


PatientRecord

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
getPatientId()	🟥	0%		n/a	1	1	1	1	1	1
PatientRecord(int, double, String, long)	🟩	100%		n/a	0	1	0	6	0	1
getMeasurementValue()	🟩	100%		n/a	0	1	0	1	0	1
getTimestamp()	🟩	100%		n/a	0	1	0	1	0	1
getRecordType()	🟩	100%		n/a	0	1	0	1	0	1
Total	3 of 27	88%	0 of 0	n/a	1	5	1	10	1	5

A file for storing patient records, everything was successfully tested except the getPatientId() method, as this function is implicitly tested in another test

- **Patient.java**



Patient

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
lambda$getRecords$0(long, long, PatientRecord)	🟥🟩	92%	🟥🟩	50%	2	3	0	1	0	1
addRecord(double, String, long)	🟩	100%		n/a	0	1	0	3	0	1
getRecords(long, long)	🟩	100%		n/a	0	1	0	4	0	1
Patient(int)	🟩	100%		n/a	0	1	0	4	0	1
getPatientId()	🟩	100%		n/a	0	1	0	1	0	1
Total	1 of 57	98%	2 of 4	50%	2	7	0	12	0	5

Everything is tested, except for one element, which is not very clear how to test, but the function in which it participates was successfully tested