Client Brief

Project goal:

Develop an application “SafetyNetAlerts” that manage the information about a population.

Project requirements:

- ✓ The SafetyNet Alerts server starts.
- ✓ SafetyNet Alerts is developed with a Model-View-Controller architecture.
- ✓ All URL endpoints are functional as well as the actuators: health, info, trace, and metrics.
- ✓ All URL endpoints log their requests and their responses. Log successful responses with the info level, log errors or exceptions on the error level, and log informative calculations or steps on the debug level.
- ✓ A working Gradle or Maven build that executes unit tests and code coverage.
- ✓ All URL endpoints and any additional functionality have unit tests.
- ✓ The build generates a SureFire test report of the JUnit test results.
- ✓ The build includes a JaCoCo unit test coverage report and archives a code coverage of 80%.

Tech solution:

- Created five POJO classes: FireStation, Household, MedicalRecord, Person, PersonsListServicedByStation.
-To be able to retrieve data from the compiled data.json file, the JSONReader class was created
-For CRUD and data manipulation, repository classes were created: FireStationRepository, MedicalRecordRepository, PersonRepository
-All business logic of an application, such as calculations, data transformations, data processes were placed in a service package
-To be able to serve all URL endpoints, a controller package with appropriate classes was created.

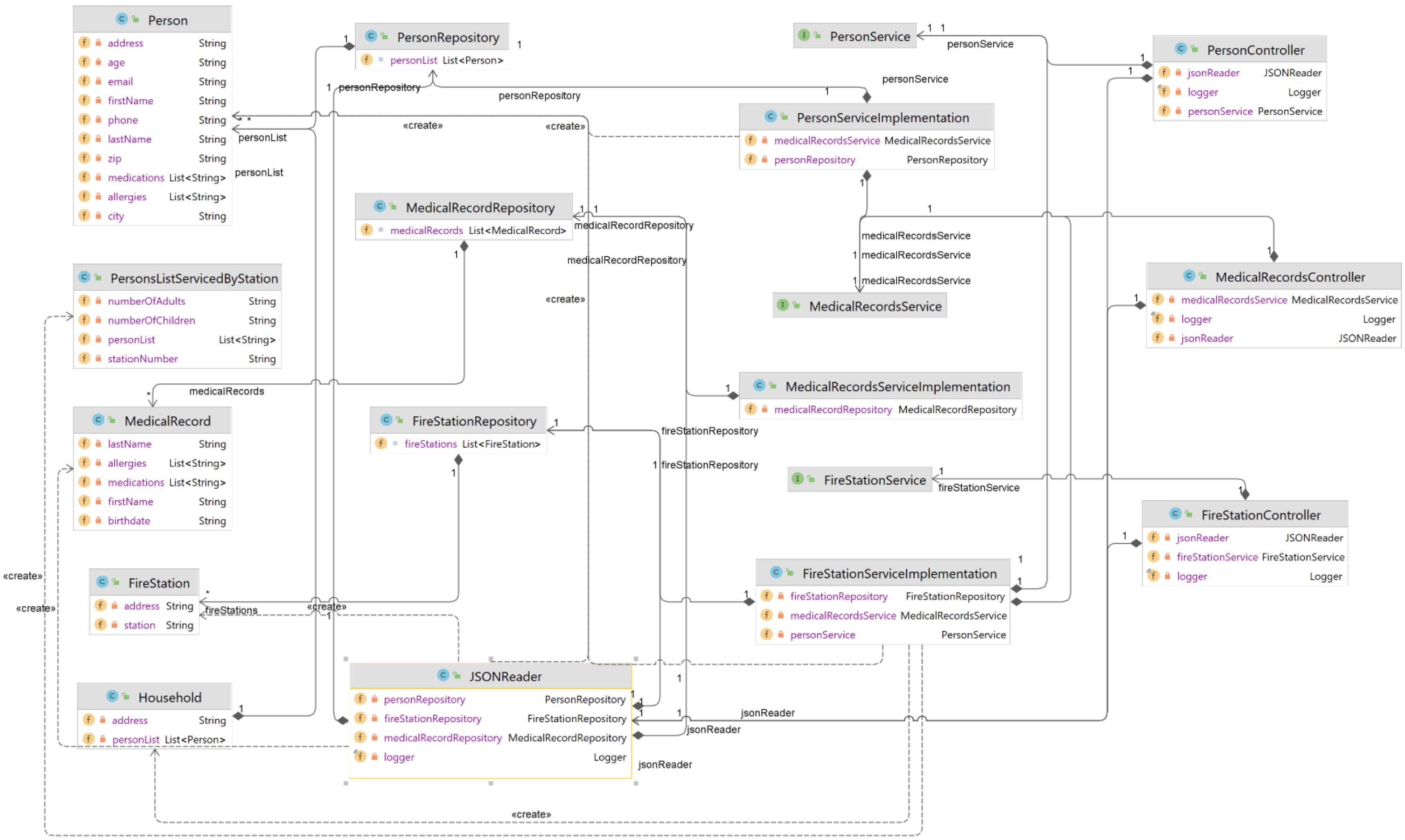
- Make sure that the application works correctly and meets all requirements. I tested my implementation using Postman and with unit and web layer tests.

What I would like to do differently:

-Would create a database with appropriate tables to store all data and do extraction when needed instead of using one compiled file.

- Establish more accurate time estimates for next time to improve development planning.

Domain model:



Person
address String
age String
email String
firstName String
phone String
lastName String
zip String
medications List<String>
allergies List<String>
city String
PersonRepository
personList List<Person>
PersonService
PersonController
jsonReader JSONReader
logger Logger
personService PersonService
PersonServiceImplementation
medicalRecordsService MedicalRecordsService
personRepository PersonRepository
MedicalRecordRepository
medicalRecords List<MedicalRecord>
MedicalRecordsService
MedicalRecordsController
medicalRecordsService MedicalRecordsService
logger Logger
jsonReader JSONReader
PersonsListServicedByStation
numberOfAdults String
numberOfChildren String
personList List<String>
stationNumber String
MedicalRecordsServiceImplementation
medicalRecordRepository MedicalRecordRepository
MedicalRecord
lastName String
allergies List<String>
medications List<String>
firstName String
birthdate String
FireStationRepository
fireStations List<FireStation>
FireStationService
FireStationController
jsonReader JSONReader
fireStationService FireStationService
logger Logger
FireStation
address String
station String
FireStationServiceImplementation
fireStationRepository FireStationRepository
medicalRecordsService MedicalRecordsService
personService PersonService
Household
address String
personList List<Person>
JSONReader
personRepository PersonRepository
fireStationRepository FireStationRepository
medicalRecordRepository MedicalRecordRepository
logger Logger
personRepository
personService
personList
medicalRecordRepository
medicalRecordsService
medicalRecords
fireStationRepository
fireStationService
fireStations
jsonReader
«create»
1
*