Data lake

General information

Who are we:
Viktor Lindström Söraas
Nonno Rydgren
Kasper Lindström
Jonathan Lindqvist

Task provide by:
Harsha krishna

Supervised by:
Reine Bergström
Jonas Willén

Course:
HI1036 Mjukvarukonstruktion, projektkurs

Video

Functionality from the user interface

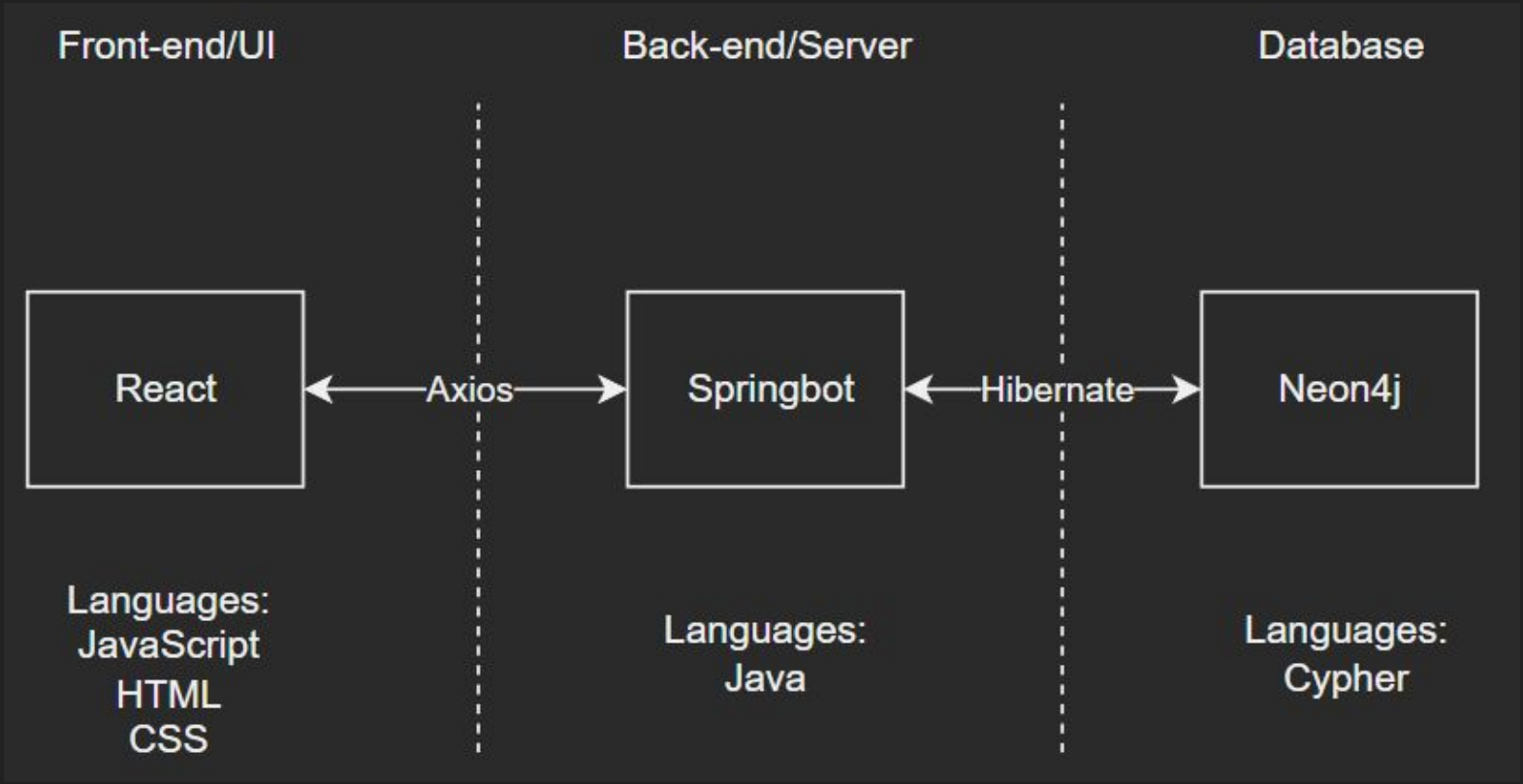
User

- Download the data depending on their authorization
- Change password
- Change first and last name

Admin

- Add new datasets
- See additional information about the dataset
- Add/Change/Remove Users
- Change users access to different datasets

Frame works + architecture



The task, Inconsistency between datasets

Dataset 261

PHATOM_ID	GENDER	AGE	RACE
12727	2	56	5
13700	2	62	5
14768	1	62	5
14795	1	41	5
16718	2	59	5
16790	2	77	5
17712	1	60	5
17726	1	61	5

Dataset 266

SEXCD	SEXCDC	SEX	RACECD	RACE
0	M	Male	1	White
0	M	Male	1	White
0	M	Male	1	White
0	M	Male	1	White
0	M	Male	1	White
0	M	Male	1	White
0	M	Male	1	White
0	M	Male	1	White

Identify two inconsistencies:

- Different headers on cells
- Different values on cells

Where the explanation exists

Identify heards:

DTH
1
1
1
1

STATUS
2
2
2

PDF **DATA DICTIONARY**
Amgen 20010145 DDT.pdf

PDF **CRF**
C9732_CRFs.pdf

Identify Values:

Overall Survival Death - Taken from 3 sources : if RAW.EOS.eoscd = '11' then dth = 1 otherwise if deathdt from the STATUS dataset is populated dth=1.
If still no death date then we check the AE dataset, if RAW.ADVERSE.SEVRCD=05, then dth=1. If the date from these source is greater than 22nd february then we set DTH =0, and for all other subjects who have not died DTH=0.

☐ **Survival status**
1-Alive
2-Dead
3-Lost to follow-up

Our solution to connecting headers

```
HashMap<String, Integer> rowNumbers = new HashMap<>();
XSSFRow row = worksheet.getRow(index);
//Searches for the headers and connect those index number to the hashmap location
for (Cell r : row) {
    //String rowValue = r.toString().toUpperCase();
    switch (r.toString()) {
        case "PHATOM_ID", "SUBJID" -> rowNumbers.put("id", r.getColumnIndex());//id
        case "GENDER", "SEX" -> rowNumbers.put("gender", r.getColumnIndex());//gender
        case "AGE" -> rowNumbers.put("age", r.getColumnIndex());//age (years)
        case "RACE" -> rowNumbers.put("ethnicity", r.getColumnIndex());//race
        case "PD" -> rowNumbers.put("relapse", r.getColumnIndex());//relapse time
        case "OS_TIME" -> rowNumbers.put("overall survival time", r.getColumnIndex());//overall survival time (months)
        case "PD_TIME" -> rowNumbers.put("relapse time", r.getColumnIndex());//relapse time (months)
        case "PFS_STATUS" -> rowNumbers.put("failure free survival", r.getColumnIndex());//failure free survival
        case "PFS_TIME" -> rowNumbers.put("failure free survival time", r.getColumnIndex());//failure free survival time (months)
        case "TRT_ARM_LABEL", "CHPTERM" -> rowNumbers.put("treatment drug", r.getColumnIndex());//treatment drug
        case "STATUS", "DTH" -> rowNumbers.put("overall survival status", r.getColumnIndex());//overall survival status
        case "CAUSEDTH" -> rowNumbers.put("cause of death", r.getColumnIndex());//cause of death
        case "NEW_MALIG" -> rowNumbers.put("new malignancy", r.getColumnIndex());//new malignancy
    }
}
return rowNumbers;
```

Our solution to connecting values

```java
public void setGender(String gender) {
    switch (gender) {
        case "1.0", "1", "Male" -> this.gender = Gender.MALE;
        case "2.0", "2", "Female" -> this.gender = Gender.FEMALE;
        default -> this.gender = Gender.UNKNOWN;
    }
}
```

```java
public void setOverAllSurvivalStatus(String overAllSurvivalStatus, String name) {
    if (name.equals("266")) overAllSurvivalStatus = String.valueOf(Double.parseDouble(overAllSurvivalStatus) + 1);
    switch (overAllSurvivalStatus) {
        case "1.0", "1" -> this.overAllSurvivalStatus = "Alive";
        case "2.0", "2" -> this.overAllSurvivalStatus = "Death";
        case "3.0", "3" -> this.overAllSurvivalStatus = "Lost to follow-up";
        default -> this.overAllSurvivalStatus = "Unknown";
    }
}
```

Creation of nodes and their relationships

Load in existing nodes from the database:

```
ArrayList<Treatment> treatmentList = treatmentRepository.findAll();
ArrayList<String> treatmentName = new ArrayList<>();
for (Treatment t : treatmentList) treatmentName.add(t.getTreatment());
```

Get treatment from the dataset if the header was identify:

```
if (rowNumbers.containsKey("treatment drug") && row.getCell(rowNumbers.get("treatment drug")) != null)
    addToMap(Patientmap, Integer.parseInt(row.getCell(rowNumbers.get("id")).toString().replace(".0", "")), row.getCell(rowNumbers.get("treatment drug")).toString());
```

Check if treatment already exist, if not create it and then create the relationship to the patient

```
    if (!treatmentName.contains(treatment.getTreatment())) {
        treatmentRepository.save(treatment);
        treatmentName.add(treatment.getTreatment());
        treatmentList.add(treatment);
        patient.setTreatment(treatment);
    } else
        for (Treatment a : treatmentList)
            if (a.getTreatment().equals(treatment.getTreatment()))
                patient.setTreatment(a);
```

Node representation in the server

- CauseOfDeath.java
- Gender.java
- NewMalignancy.java
- OverAllSurvivalStatus.java
- Patient.java
- Symptoms.java
- Treatment.java

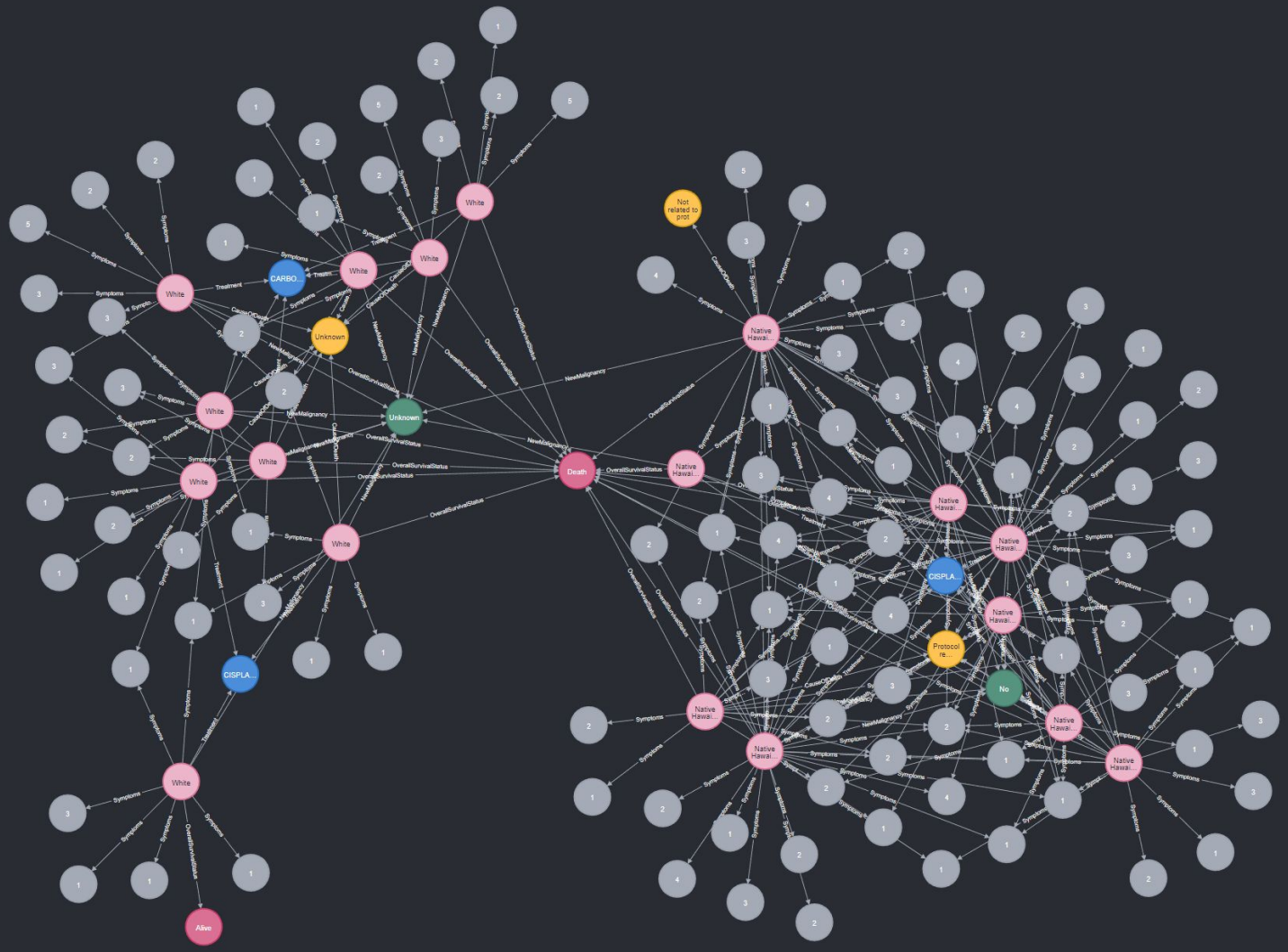
```java
@NotBlank
private int age;

@NotBlank
private Gender gender;

@NotBlank
private String ethnicity;

@NotBlank
private int subjectId;


@Relationship(type = "Treatment", direction = Relationship.Direction.OUTGOING)
private Treatment treatment;
```


Death
Alive
Unknown
White
Native Hawai...
CARBO...
CISPLA...
Not related to prot
Protocol re...
No
Symptoms
Treatment
NewMalignancy
OverallSurvivalStatus
CauseDeath

Patient

<id>	779
age	68
dataset	266
ethnicity	White
failureFreeSurvivalStatus	Unknown
failureFreeSurvivalTime	-1.0
gender	MALE
relapse	Unknown
relapseTime	-1.0
subjectId	293
survivalTime	-1.0

Patient

<id>	457
age	68
dataset	261
ethnicity	Native Hawaiian
failureFreeSurvivalStatus	Survived
failureFreeSurvivalTime	1.675564681724846
gender	MALE
relapse	Yes
relapseTime	1.675564681724846
subjectId	76723
survivalTime	2.0698151950718686

Node labels

*(1,958) CauseOfDeath NewMalignancy OverAllSurvivalStatus Patient Symptoms Treatment User

Relationship types

*(19,082) CauseOfDeath NewMalignancy OverallSurvivalStatus Symptoms Treatment

Limitations and
Creating new nodes or value/header definitions

Code only works for datasets that have the same header design and value definitions as the dataset in the switch case and entities classes

Creating a new type of node demands quite a bit of new code

- A new entity class needs to be created
- Old entities classes may need to be updated with new the relation to the node
- Service layer needs to be updated

implementing a new datasets is much simpler

- Add header to the switch case in the service class
- Add value definition to the switch case in the entity class

Further development

Implement support for more headers and values or make it possible for the person whom adds a new datasets to define theses values

Implement support for more file types

Implement security (TLS)

Implement different quarry methods

Currently the software is case sensitive, if this would change more connections between datasets might be made

Info mail

Contact information from us

Link to both githubs

Excel/CSV file of the database with both datasets combined

Copy of the presentation