

# Handin 4 - Journal

## *gruppe 13*

Bjørn Nørgaard Sørensen  
stud.nr: 201370248  
bjornnorgaard@post.au.dk

Joachim Dam Andersen  
stud.nr: 201370031  
mr.anderson@post.au.dk

Joachim Wind Someting Someting  
stud.nr: 201370318  
mr.derp@post.au.dk

May 17, 2016

## Indholdsfortegnelse

<b>1</b>	<b>Database design</b>	<b>1</b>
1.1	Rules/requirements describing the associations/relationships	1
1.1.1	Antagelser	1
1.1.2	Regler	1
1.2	UML og ER diagrams	1
1.3	Stored procedures	1
<b>2</b>	<b>JSON og netværk</b>	<b>1</b>
<b>3</b>	<b>Konklusion</b>	<b>1</b>

# 1 Database design

## 1.1 Rules/requirements describing the associations/relationships

### 1.1.1 Antagelser

For at designe system mest effektivt, kiggede vi først på hvad vi faktisk kunne trække ud af serveren. Dette viste sig at være følgende attributter:

- SensorId.
- ApartmentId.
- Value.
- Timestamp.

Af dette kom vi frem til to entities: *Sensor* og *Measurement*, som hver kan ses beskrevet vha. ER og UML diagram på figur 1 og 2.

### 1.1.2 Regler

Med dette kunne vi da opstille to regler for hvordan relationen skulle opføre sig:

- Sensor can have \* (Many) instances of Measurement.
- Measurement can have 1 (One) instance of Sensor.

## 1.2 UML og ER diagrams

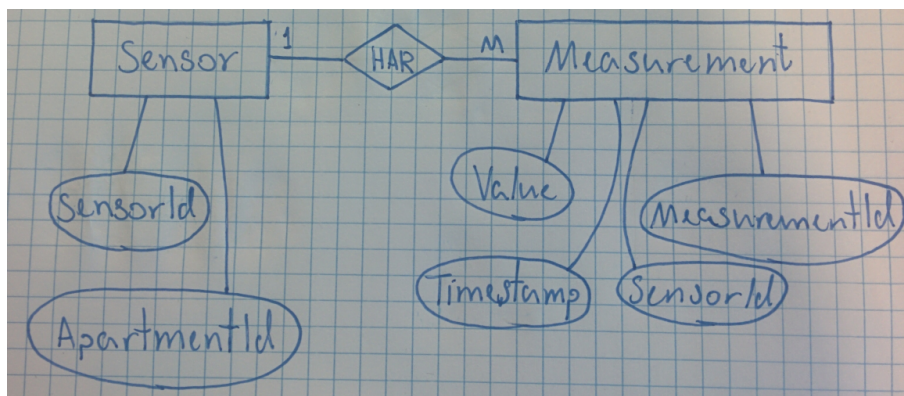


Figure 1: ER diagram for databasedesign.

## 1.3 Stored procedures

# 2 JSON og netværk

# 3 Konklusion

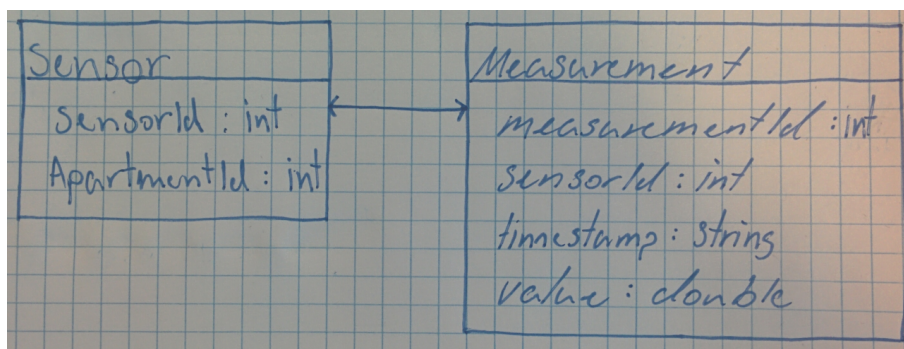


Figure 2: UML diagram for databasedesign.