Statistik-Daten Auswertung

Masterarbeit HSR-MAS-SE 2016

Anforderungsspezifikation
# Inhaltsverzeichnis

## 1 Zu diesem Dokument
- 1.1 Zweck ................................................................. 5
- 1.2 Gültigkeit .......................................................... 5
- 1.3 Verfügbarkeit des Dokumentes ................................ 5

## 2 Weiterführende Informationen
- 2.1 Definitionen, Akronyme und Abkürzungen .................. 6
- 2.2 Ergänzende Dokumente .......................................... 6
- 2.3 Prozessbezogene Dokumente ................................... 6
- 2.4 Versionshistorie ..................................................... 6
- 2.5 Verteilung ............................................................ 6

## 3 Einführung und Projektziel
- 3.1 Vision / Ziel ....................................................... 7
- 3.2 Stakeholder ......................................................... 7

## 4 Systemabgrenzung
- 4.1 Systemkontext ...................................................... 8
- 4.1.1 Zentrale Datenstrukturen (vgl. Kap: 7) ..................... 8

## 5 Akteure
- 5.1 A1: Natürliche Person ........................................... 9
- 5.1.1 A1a: System-Administrator .................................. 9
- 5.1.2 A1b: Daten-Manipulator ..................................... 9
- 5.1.3 A1c: Daten-Verbraucher ..................................... 9
- 5.1.4 A1d: Informations-Verbraucher ............................. 9
- 5.1.5 Rollenzuteilung .................................................. 9
- 5.2 A2: Datenquellen .................................................. 10
- 5.2.1 A2a: Aktuator ................................................... 10
- 5.2.2 A2b: Legacy-System .......................................... 10
- 5.2.3 A2c: Daten-Import ............................................. 10
- 5.3 A3: Verwaltungs-/Hostsysteme ................................ 10
- 5.3.1 A3a: Daten-Manipulations-System ......................... 10
- 5.3.2 A3b: Daten-Verbraucher-System ........................... 10
- 5.3.3 A3c: Informations-Verbraucher-System ................... 10

## 6 Anwendungsfälle
- 6.1 Übersicht ........................................................... 11
- 6.2 UC10: Statistik-Daten-Selektion bekannt geben ............. 12
- 6.2.1 Beschreibung .................................................... 12
- 6.2.2 Ablauf ............................................................ 13
- 6.3 UC11: Alle Statistik-Daten-Selektions-Parametersets bekannt geben ..................................................... 13

Status: RELEASE  
Version: 006  
anforderungsspezifikation_public.docx  
Classification: PUBLIC  
Seite 2 / 29
7 Datenstrukturen der Systemschnittstellen ................................................. 22
  7.1 Einführung................................................................................................. 22
  7.2 Logische Datenstrukturen ........................................................................ 22
  7.2.1 Statistik-Daten-Tupel ......................................................................... 22
  7.2.2 Statistik-Daten-Tupel-Ack .................................................................. 22
  7.2.3 Statistik-Daten-Selektions-Parameterset ............................................. 22
  7.2.4 Statistik-Daten-Selektion .................................................................... 22
  7.2.5 Statistik-Daten-Selektions-Parameterset-Req ...................................... 22
  7.2.6 Statistik-Daten-Selektions-Parameterset-Ack ...................................... 22
  7.2.7 Statistik-Daten-Report-Parameterset .................................................. 22
  7.2.8 Statistik-Daten-Report ................................................................. 22
  7.2.9 Statistik-Daten-Report-Parameterset-Req .......................................... 22
  7.2.10 Statistik-Daten-Report-Parameterset-Ack ......................................... 22
  7.3 Physische Systemschnittstellen ............................................................... 23
  7.3.1 UC10, UC11, UC12 Statistik-Daten-Selektion ....................................... 23
  7.3.2 UC20, UC21, UC23 Statistik-Daten-Report ......................................... 24
  7.3.3 UC30, UC31 Statistik-Daten-Tupel ...................................................... 24

8 Nicht-funktionale Systemanforderungen ......................................................... 25
  8.1 Benutzbarkeit ............................................................................................ 25
  8.2 Verfügbarkeit .............................................................................................. 25
  8.3 Robustheit ................................................................................................. 25
  8.4 Zuverlässigkeit ........................................................................................... 25
  8.5 Effizienz ...................................................................................................... 25
  8.6 Systemsicherheit ....................................................................................... 25
  8.7 Änderbarkeit .............................................................................................. 25
9  Klassenbeschreibungen ................................................................. 26
  9.1  Einführung ............................................................................. 26
  9.2  Essentielles Klassendiagramm ............................................... 26
  9.3  Klassenbeschreibung ............................................................ 27
  9.3.1  Statistik-Daten-Tupel ......................................................... 27
  9.3.2  Statistik-Daten-Tupel-Modifikation ................................. 27
  9.3.3  Statistik-Daten-Vergleichs-Tupel ....................................... 27
  9.3.4  Statistik-Daten-Selektion .................................................. 27
  9.3.5  Statistik-Daten-Selektions-Parameterset ............................ 28
  9.3.6  Statistik-Daten-Report ....................................................... 28
  9.3.7  Statistik-Daten-Report-Parameterset ................................... 28

10  Randbedingungen an die Realisierung ........................................ 29
  10.1  Benutzerbarkeit .................................................................. 29
  10.2  Verfügbarkeit ...................................................................... 29
  10.3  Robustheit ......................................................................... 29
  10.4  Zuverlässigkeit ................................................................... 29
  10.5  Effizienz ............................................................................ 29
  10.6  Systemsicherheit ................................................................. 29
  10.7  Änderbarkeit ...................................................................... 29
  10.8  Betriebliche Anforderungen ............................................... 29

Abbildungsverzeichnis

Abbildung 1: Kontextdiagramm ................................................................. 8
Abbildung 2: Übersicht Anwendungsfälle ................................................. 11
Abbildung 3: UC10 - Statistik-Daten-Selektion bekannt geben ............... 13
Abbildung 4: UC11 - Alle Statistik-Daten-Selektions-Parametersets bekannt geben ......................................................... 14
Abbildung 5: UC12 - Statistik-Daten-Selektions-Parameterset entgegennommen ......................................................... 15
Abbildung 6: UC20 - Statistik-Daten-Report bekannt geben ................. 16
Abbildung 7: UC21 - Alle Statistik-Daten-Report-Parametersets bekannt geben ......................................................... 17
Abbildung 8: UC22 - Statistik-Daten-Report-Parameterset entgegennommen ......................................................... 19
Abbildung 9: UC30 - Statistik-Daten-Tupel entgegennommen .............. 20
Abbildung 10: UC31 - Modifizierte Statistik-Daten-Tupel entgegennommen ......................................................... 21
Abbildung 11: Daten-Selektion ............................................................. 23
Abbildung 12: Report-Funktion ............................................................ 24
Abbildung 13: Daten-Eingabe .............................................................. 24
Abbildung 14: Essentielles Klassendiagramm ....................................... 26
1 Zu diesem Dokument

1.1 Zweck

Die Anforderungsspezifikation zeigt die Abgrenzung zwischen der „Statistik-Daten Auswertung“ und deren Umsysteme, deren Anwendungsfälle, die logischen Datenstrukturen der Systemschnittstellen, die nicht-funktionellen Anforderungen und den Kern der für die Funktionserfüllung notwendigen Information und Abläufe.

1.2 Gültigkeit

Während der Inception Phase des Projekts „Statistik-Daten Auswertung“ wird eine erste Stufe der Anforderungsspezifikation ausgearbeitet. Diese Stufe hält den Focus auf das geplante «Minimal Viable Product».

In der Elaborations-Phase wird diese Anforderungsspezifikation sukzessive erweitert, so dass schlussendlich alle Anforderungen für das Endprodukt abgedeckt sind.


1.3 Verfügbarkeit des Dokumentes

Dieses Dokument befindet sich im Projektverzeichnis: 10_Anforderungsspezifikation
2 Weiterführende Informationen

2.1 Definitionen, Akronyme und Abkürzungen

Sämtliche Definitionen, Akronyme und Abkürzungen werden zentral in einem Projektglossar dokumentiert. Das Projektglossar wird kontinuierlich angepasst.

Das Projektglossar befindet sich im Projektverzeichnis: 09_Glossar

2.2 Ergänzende Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>
ISBN: 978-3-8274-2903-2 |

2.3 Prozessbezogene Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

2.4 Versionshistorie

(nicht öffentlich)

2.5 Verteilung

<table>
<thead>
<tr>
<th>Ver.</th>
<th>Name</th>
<th>Rolle / Titel / Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>006</td>
<td>MAS-Arbeit</td>
<td>MAS-Arbeit</td>
</tr>
</tbody>
</table>
3 Einführung und Projektziel

3.1 Vision / Ziel


3.2 Stakeholder

(nicht öffentlich)
4 Systemabgrenzung

4.1 Systemkontext


Es soll die Möglichkeit bestehen, dass verschiedene Akteure die Daten selektiv abfragen können. Zudem soll das System über eine Report-Funktion verfügen, die es erlaubt, eine automatische Ist-Soll-Analyse der Daten durchzuführen.

Das Abfragen und Analysieren der Daten soll für jeden Akteur ohne fremdes Zutun möglich sein. Das neue System muss daher die Rollen der verschiedenen Akteure unterstützen.

4.1.1 Zentrale Datenstrukturen (vgl. Kap: 7)

<table>
<thead>
<tr>
<th>System empfängt</th>
<th>System versendet</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Statistik-Daten-Tupel vgl. 7.2.1</td>
</tr>
<tr>
<td>II</td>
<td>Statistik-Daten-Selektions-Parameterset vgl. 7.2.3</td>
</tr>
<tr>
<td>III</td>
<td>Statistik-Daten-Selektions-Parameterset-Req vgl. 7.2.5</td>
</tr>
<tr>
<td>IV</td>
<td>Statistik-Daten-Selektions-Parameterset vgl. 7.2.3</td>
</tr>
<tr>
<td>V</td>
<td>Statistik-Daten-Report-Parameterset vgl. 7.2.7</td>
</tr>
<tr>
<td>VI</td>
<td>Statistik-Daten-Report-Parameterset-Req vgl. 7.2.9</td>
</tr>
<tr>
<td>VII</td>
<td>Statistik-Daten-Report-Parameterset vgl. 7.2.7</td>
</tr>
</tbody>
</table>
5 Akteure

5.1 A1: Natürliche Person

5.1.1 A1a: System-Administrator

Der System-Administrator verwaltet und konfiguriert das System. Das heisst, er erstellt Statistik-Daten-Selektions-Parametersets und Statistik-Daten-Report-Parametersets.

5.1.2 A1b: Daten-Manipulator

Der Daten-Manipulator verändert und löscht einzelne Statistik-Daten-Tupel.

5.1.3 A1c: Daten-Verbraucher

Der Daten-Verbraucher konsumiert Statistik-Daten-Selektionen.

5.1.4 A1d: Informations-Verbraucher

Der Informations-Verbraucher konsumiert Statistik-Daten-Reports.

5.1.5 Rollenzuteilung

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>System-Administrator</th>
<th>Daten-Manipulator</th>
<th>Daten-Verbraucher</th>
<th>Informations-Verbraucher</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entwickler (S3, S4)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Supporter (S5)</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Test-Verifikation-Validation-Team (S6)</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Projektleiter (S7)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Feldtest-Manager (S6)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
5.2 **A2: Datenquellen**


5.2.1 **A2a: Aktuator**

Der Aktuator liefert zu einem bestimmten Zeitpunkt ein bestimmtes Statistik-Daten-Tupel.

5.2.2 **A2b: Legacy-System**

Das Legacy-System liefert sämtlich bisher erfassten Statistik-Daten-Tupels. (Migration)

5.2.3 **A2c: Daten-Import**

Für Testzwecke sollen Statistik-Daten-Tupel via Dateisystem importiert werden können.

5.3 **A3: Verwaltungs-/Hostsysteme**

(nicht öffentlich)

5.3.1 **A3a: Daten-Manipulations-System**


5.3.2 **A3b: Daten-Verbraucher-System**

Ein Daten-Verbraucher-System konsumiert Statistik-Daten-Selektionen. Das «Statistik-Daten Auswertungs»-System kennt ein allfälliges Daten-Verbraucher-System nicht. Es wird eine Schnittstelle angeboten, die durch ein allfälliges Daten-Verbraucher-System angesprochen werden kann.

5.3.3 **A3c: Informations-Verbraucher-System**

6 Anwendungsfälle

Die im folgenden aufgeführten Anwendungsfälle beruhen darauf, dass die Infrastruktur bereits alle fehlerhaften Eingaben, welche in den physischen Schnittstellen entstehen können, bereinigt hat. Somit beschreiben die Anwendungsfälle die Abläufe und Regeln der Administration und natürlich deren Kern.

Anwendungsfälle sind aus Sicht des Systems formuliert, d.h.:

- Das System nimmt Daten entgegen und gibt bekannt.
- Der Akteur startet eine Aktion z.B. mit dem Aufruf von «entgegennehmen()» oder «bekannt geben()».

6.1 Übersicht

Das folgende Diagramm ist eine Übersicht der essentiellen Anwendungsfälle.
6.2 UC10: Statistik-Daten-Selektion bekannt geben

6.2.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC10</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>Statistik-Daten-Selektion bekannt geben</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Hoch</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>Auf Anfrage gibt das System dem Akteur die gewünschte Statistik-Daten-Selektion bekannt.</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Akteur setzt mittels Statistik-Daten-Selektions-Parameterset eine Anfrage ab.</td>
</tr>
<tr>
<td>Akteure</td>
<td>A1c: Daten-Verbraucher</td>
</tr>
<tr>
<td></td>
<td>A3b: Daten-Verbraucher-System</td>
</tr>
<tr>
<td>Datenstruktur (Anfrage): Akteur → System</td>
<td>Statistik-Daten-Selektions-Parameterset vgl. 7.2.3</td>
</tr>
<tr>
<td>Datenstruktur (Antwort): System → Akteur</td>
<td>Statistik-Daten-Selektion vgl. 7.2.4</td>
</tr>
<tr>
<td>Vorbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Nachbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Ergebnis</td>
<td>Akteur nimmt gewünschte Statistik-Daten-Selektion entgegen.</td>
</tr>
<tr>
<td>Hauptszenario</td>
<td>1) Anfrage zur Bekanntgabe einer Statistik-Daten-Selektion durch den Akteur.</td>
</tr>
<tr>
<td></td>
<td>2) System verwendet das gesendete Statistik-Daten-Selektions-Parameterset zur Erstellung der gewünschten Statistik-Daten-Selektion.</td>
</tr>
<tr>
<td></td>
<td>3) System gibt die Statistik-Daten-Selektion dem Akteur bekannt.</td>
</tr>
<tr>
<td>Alternativszenarien</td>
<td>-</td>
</tr>
<tr>
<td>Ausnahmeszenarien</td>
<td>1) Anfrage zur Bekanntgabe einer Statistik-Daten-Selektion durch den Akteur.</td>
</tr>
<tr>
<td></td>
<td>2) System verwendet das gesendete Statistik-Daten-Selektions-Parameterset zur Erstellung der gewünschten Statistik-Daten-Selektion.</td>
</tr>
<tr>
<td></td>
<td>3) Fehlerfall</td>
</tr>
<tr>
<td></td>
<td>4) System gibt dem Akteur bekannt, dass das Statistik-Daten-Selektions-Parameterset nicht verarbeitet werden konnte.</td>
</tr>
<tr>
<td>Fehlerfall</td>
<td>• Verbindungsfehler können detektiert und dem Benutzer bekannt gegeben werden.</td>
</tr>
<tr>
<td></td>
<td>• Ein fehlendes Statistik-Daten-Selektions-Parameterset bei der Anfrage kann detektiert und dem Benutzer bekannt gegeben werden.</td>
</tr>
<tr>
<td></td>
<td>• Ein nicht verarbeitbares Statistik-Daten-Selektions-Parameterset kann detektiert und dem Benutzer bekannt gegeben werden.</td>
</tr>
</tbody>
</table>
6.2.2 Ablauf

Abbildung 3: UC10 - Statistik-Daten-Selektion bekannt geben

6.3 UC11: Alle Statistik-Daten-Selektions-Parametersets bekannt geben

6.3.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC11</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>Alle Statistik-Daten-Selektions-Parametersets bekannt geben</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Mittel</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>Auf Anfrage gibt das System dem Akteur sämtliche Statistik-Daten-Selektions-Parametersets bekannt.</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Akteur setzt mittels Statistik-Daten-Selektions-Parameterset-Req eine Anfrage ab.</td>
</tr>
<tr>
<td>Akteure</td>
<td>A1c: Daten-Verbraucher A3b: Daten-Verbraucher-System</td>
</tr>
<tr>
<td>Datenstruktur (Anfrage)</td>
<td>Statistik-Daten-Selektions-Parameterset vgl. 7.2.5</td>
</tr>
<tr>
<td>Datenstruktur (Antwort)</td>
<td>Statistik-Daten-Selektions-Parameterset vgl. 7.2.3</td>
</tr>
<tr>
<td>Vorbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Nachbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Ergebnis</td>
<td>Akteur nimmt sämtliche Statistik-Daten-Selektions-Parametersets entgegen.</td>
</tr>
</tbody>
</table>
| Hauptszenario             | 1) Anfrage zur Bekanntgabe sämtlicher Statistik-Daten-Selektions-Parametersets durch den Akteur.  
|                           | 2) System gibt sämtliche Statistik-Daten-Selektions-Parametersets dem Akteur bekannt. |
Alternativszenarien -
Ausnahmeszenarien -
Fehlerfall

- Verbindungsfehler können detektiert und dem Benutzer bekannt gegeben werden.

6.3.2 Ablauf

UC11: Alle Statistik-Daten-Selektions-Parametersets bekannt geben

![Diagramm UC11]

Abbildung 4: UC11 - Alle Statistik-Daten-Selektions-Parametersets bekannt geben

6.4 UC12: Statistik-Daten-Selektions-Parameterset entgegennehmen

6.4.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC12</td>
</tr>
<tr>
<td>Anwendungssfallname</td>
<td>Statistik-Daten-Selektions-Parameter entgegennehmen</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Mittel</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>Das System nimmt ein Statistik-Daten-Selektions-Parameterset entgegen, welches vom Akteur übermittelt wurde.</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Akteur übermittelt Statistik-Daten-Selektions-Parameterset.</td>
</tr>
<tr>
<td>Akteure</td>
<td>A1a: System-Administrator</td>
</tr>
<tr>
<td>Datenstruktur (Anfrage): Akteur → System</td>
<td>Statistik-Daten-Selektions-Parameterset vgl. 7.2.3</td>
</tr>
<tr>
<td>Datenstruktur (Antwort): System → Akteur</td>
<td>Statistik-Daten-Selektions-Parameterset-Ack vgl. 7.2.8</td>
</tr>
<tr>
<td>Vorbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Nachbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Ergebnis</td>
<td>Akteur nimmt triviale Empfangsbestätigung zur Kenntnis.</td>
</tr>
<tr>
<td>Hauptszenario</td>
<td>1) Eingabe eines Statistik-Daten-Selektions-Parametersets durch den Akteur.</td>
</tr>
</tbody>
</table>
2) System verarbeitet das Statistik-Daten-Selektions-Parameterset. 
3) System gibt triviale Empfangsbestätigung dem Akteur bekannt.

Alternativszenarien: 
- 

Ausnahmeszenarien: 
1) Eingabe eines Statistik-Daten-Selektions-Parametersets durch den Akteur. 
2) System überprüft das Statistik-Daten-Selektions-Parameterset. 
3) System gibt dem Akteur bekannt, dass die Eingabe nicht verarbeitet werden konnte.

Fehlerfall: 
- Nicht-verarbeitbare Statistik-Daten-Selektions-Parametersets können detektiert und dem Benutzer bekannt gegeben werden.

6.4.2 Ablauf

UC12: Statistik-Daten-Selektions-Parameterset entgegennehmen

Abbildung 5: UC12 - Statistik-Daten-Selektions-Parameterset entgegennehmen

6.5 UC20: Statistik-Daten-Report bekannt geben

6.5.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC20</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>Statistik-Daten-Report bekannt geben</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Mittel</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Akteur setzt mittels Statistik-Daten-Report-Parameterset eine Anfrage ab.</td>
</tr>
<tr>
<td>Akteure</td>
<td>A1d: Informations-Verbraucher</td>
</tr>
<tr>
<td></td>
<td>A3c: Informations-Verbraucher-System</td>
</tr>
<tr>
<td>Datenstruktur (Anfrage): Akteur → System</td>
<td>Statistik-Daten-Report-Parameterset vgl. 7.2.7</td>
</tr>
<tr>
<td>Datenstruktur (Antwort):</td>
<td>Statistik-Daten-Report vgl. 7.2.8</td>
</tr>
</tbody>
</table>
System → Akteur

Vorbedingungen

Nachbedingungen

Ergebnis

Akteur nimmt den gewünschten Statistik-Daten-Report entgegen.

Hauptszenario
1) Anfrage zur Bekanntgabe eines Statistik-Daten-Reports durch den Akteur.
2) System verwendet das gesandete Statistik-Daten-Report-Parameterset zur Erstellung des gewünschten Statistik-Daten-Reports.
3) System gibt den Statistik-Daten-Report dem Akteur bekannt.

Alternativszenarien

Ausnahmeszenarien
1) Anfrage zur Bekanntgabe eines Statistik-Daten-Reports durch den Akteur.
2) System verwendet das gesandete Statistik-Daten-Report-Parameterset zur Erstellung des gewünschten Statistik-Daten-Reports.
3) Fehlerfall!
4) System gibt dem Akteur bekannt, dass das Statistik-Daten-Report-Parameterset nicht verarbeitet werden konnte.

Fehlerfall

- Verbindungsfehler können detektiert und dem Benutzer bekannt gegeben werden.
- Ein fehlendes Statistik-Daten-Report-Parameterset bei der Anfrage kann detektiert und dem Benutzer bekannt gegeben werden.
- Ein nicht verarbeitbares Statistik-Daten-Report-Parameterset kann detektiert und dem Benutzer bekannt gegeben werden.

6.5.2 Ablauf

Abbildung 6: UC20 - Statistik-Daten-Report bekannt geben
6.6 UC21: Alle Statistik-Daten-Report-Parametersets bekannt geben

6.6.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC21</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>Alle Statistik-Daten-Report-Parametersets bekannt geben</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Mittel</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
</tbody>
</table>
| Kurzbeschreibung| Auf Anfrage gibt das System dem Akteur sämtliche Statistik-Daten-Report-
|                | - Parametersets bekannt.                                              |
| Auslösendes Ereignis | Akteur setzt mittels Statistik-Daten-Report-Parameterset-Req eine Anfrage ab. |
| Akteure         | A1d: Informations-Verbraucher                                         |
|                  | A3c: Informations-Verbraucher-System                                  |
| Datenstruktur (Anfrage): Akteur → System | Statistik-Daten-Report-Parameterset-Req vgl. 7.2.9 |
| Datenstruktur (Antwort): System → Akteur | Statistik-Daten-Report-Parameterset vgl. 7.2.7 |
| Vorbedingungen  | -                                                                      |
| Nachbedingungen | -                                                                      |
| Ergebnis        | Akteur nimmt sämtliche Statistik-Daten-Report-Parametersets entgegen. |
| Hauptszenario   | 1) Anfrage zur Bekanntgabe sämtlicher Statistik-Daten-Report-
|                  | Parametersets durch den Akteur.                                      |
|                  | 2) System gibt sämtliche Statistik-Daten-Report-Parametersets dem Akteur
|                  | bekannt.                                                               |
| Alternativszenarien | -                                                                    |
| Ausnahmeszenarien | -                                                                    |
| Fehlerfall       | • Verbindungsfehler können detektiert und dem Benutzer bekannt
|                  | gegeben werden.                                                       |

6.6.2 Ablauf

Abbildung 7: UC21 - Alle Statistik-Daten-Report-Parametersets bekannt geben

Anforderungsspezifikation
6.7 UC22: Statistik-Daten-Report-Parameterset entgegennehmen

6.7.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC22</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>Statistik-Daten-Report-Parameterset entgegennehmen</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Mittel</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>Das System nimmt ein Statistik-Daten-Report-Parameterset entgegen, welches vom Akteur übermittelt wurde.</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Akteur übermittelt Statistik-Daten-Report-Parameterset.</td>
</tr>
<tr>
<td>Akteure</td>
<td>A1a: System-Adminstrator</td>
</tr>
<tr>
<td>Datenstruktur (Anfrage): Akteur → System</td>
<td>Statistik-Daten-Report-Parameterset vgl. 7.2.7</td>
</tr>
<tr>
<td>Datenstruktur (Antwort): System → Akteur</td>
<td>Statistik-Daten-Report-Parameterset-Ack vgl. 7.2.10</td>
</tr>
<tr>
<td>Vorbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Nachbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Ergebnis</td>
<td>Akteur nimmt triviale Empfangsbestätigung zur Kenntnis.</td>
</tr>
</tbody>
</table>

Hauptszenario
2) System verarbeitet das Statistik-Daten-Report-Parameterset.
3) System gibt triviale Empfangsbestätigung dem Akteur bekannt.

Alternativszenarien
- 

Ausnahmeszenarien
1) Eingabe eines Statistik-Daten-Report-Parametersets durch den Akteur.
2) System überprüft das Statistik-Daten-Report-Parameterset.
3) System gibt dem Akteur bekannt, dass die Eingabe nicht verarbeitet werden konnte.

Fehlerfall
- Nicht-verarbeitbare Statistik-Daten-Report-Parametersets können detektiert und dem Benutzer bekannt gegeben werden.
6.7.2 Ablauf

Abbildung 8: UC22 - Statistik-Daten-Report-Parameterset entgegennehmen

6.8 Anforderungsspezifikation

6.8.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC30</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>Statistik-Daten-Tupel entgegennehmen</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Hoch</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Hoch</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>Der Akteur übermittelt dem System ein Statistik-Daten-Tupel, welches vom System gespeichert wird.</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Akteur übermittelt Statistik-Daten-Tupel.</td>
</tr>
<tr>
<td>Akteure</td>
<td>A2a: Aktuator</td>
</tr>
<tr>
<td></td>
<td>A2b: Legacy-System</td>
</tr>
<tr>
<td></td>
<td>A2c: Daten-Import</td>
</tr>
<tr>
<td></td>
<td>A3a: Daten-Manipulations-System</td>
</tr>
<tr>
<td>Datenstruktur (Anfrage): Akteur → System</td>
<td>Statistik-Daten-Tupel vgl. 7.2.1</td>
</tr>
<tr>
<td>Datenstruktur (Antwort): System → Akteur</td>
<td>Statistik-Daten-Tupel-Ack vgl. 7.2.2</td>
</tr>
<tr>
<td>Vorbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Nachbedingungen</td>
<td>-</td>
</tr>
<tr>
<td>Ergebnis</td>
<td>Akteur nimmt Speicherung des Statistik-Daten-Tupels zur Kenntnis.</td>
</tr>
<tr>
<td>Hauptszenario</td>
<td>1) Übermittlung des Statistik-Daten-Tupels durch den Akteur.</td>
</tr>
<tr>
<td></td>
<td>2) System überprüft das Statistik-Daten-Tupel.</td>
</tr>
<tr>
<td></td>
<td>3) System speichert das Statistik-Daten-Tupel.</td>
</tr>
</tbody>
</table>
4) System gibt Speicherung des Statistik-Daten-Tupels dem Akteur bekannt.

Alternativszenarien

1) Übermittlung des Statistik-Daten-Tupels durch den Akteur.  
2) System überprüft das Statistik-Daten-Tupel.  
3) Extension Point: «Statistik-Daten-Tupel unvollständig»  
4) System ermöglicht dem Akteur den UC31b durchzuführen.  
5) Nach UC31b wird das Hauptszenario ab 2) ausgeführt.

Ausnahmeszenarien

1) Übermittlung des Statistik-Daten-Tupels durch den Akteur.  
2) System überprüft das Statistik-Daten-Tupel.  
3) System gibt dem Akteur bekannt, dass das Statistik-Daten-Tupel nicht verarbeitet werden konnte.

Fehlerfall

- Verbindungsfehler können detektiert und dem Benutzer bekannt gegeben werden.  
- Nicht-verarbeitbare Statistik-Daten-Tupels können detektiert und dem Benutzer bekannt gegeben werden.  
- Nicht erfolgreiche Speicherung eines Statistik-Daten-Tupels kann detektiert und dem Benutzer bekannt gegeben werden.

6.8.2 Ablauf

Abbildung 9: UC30 - Statistik-Daten-Tupel entgegennehmen

6.9 UC31: Modifiziertes Statistik-Daten-Tupel entgegennehmen

6.9.1 Beschreibung

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC31</td>
</tr>
<tr>
<td>Anwendungssfallname</td>
<td>Modifiziertes Statistik-Daten-Tupel entgegennehmen</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Mittel</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Hoch</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>UC31a) Ein bereits vorhandenes Tupel muss modifiziert werden (update).</td>
</tr>
<tr>
<td></td>
<td>UC31b) Ein in UC30 übermitteltes Statistik-Daten-Tupel ist unvollständig. Der Akteur kann ein modifiziertes/vollständiges Statistik-Daten-Tupel übermitteln.</td>
</tr>
</tbody>
</table>
### Auslösendes Ereignis

| UC31a) | Akteur übermittelt ein modifiziertes Statistik-Daten-Tupel |
| UC31b) | Extension Point «Statistik-Daten-Tupel unvollständig» UC30 |

### Akteure

| UC31a) | A1b: Daten-Manipulator |
| UC31b) | A3a: Daten-Manipulations-System |

### Datenstruktur (Anfrage):

**Akteur → System**

Statistik-Daten-Tupel vgl. 7.2.1

### Datenstruktur (Antwort):

**System → Akteur**

Statistik- Daten-Tupel-Ack vgl. 7.2.2

### Vorbedingungen

| UC31a) | Ein in UC30 übermitteltes Statistik-Daten-Tupel ist unvollständig. Der Extension Point «Statistik-Daten-Tupel unvollständig» wird ausgelöst. |

### Nachbedingungen

| UC31a) | Nach Abschluss von UC31, wird UC30 fortgesetzt. |
| UC31b) | |

### Ergebnis

- 

### Hauptszenario

1) Übermittlung des modifizierten Statistik-Daten-Tupels durch den Akteur.
2) System setzt UC30 fort: «System überprüft das Statistik-Daten-Tupel.»

### Alternativen- und Ausnahmen

- 

### Fehlerfall

vgl. UC30

---

### 6.9.2 Ablauf

**Abbildung 10: UC31 - Modifiziertes Statistik-Daten-Tupel entgegennehmen**
7 Datenstrukturen der Systemschnittstellen

7.1 Einführung

7.2 Logische Datenstrukturen

7.2.1 Statistik-Daten-Tupel
(nicht öffentlich)

7.2.2 Statistik-Daten-Tupel-Ack
(nicht öffentlich)

7.2.3 Statistik-Daten-Selektions-Parameterset
(nicht öffentlich)

7.2.4 Statistik-Daten-Selektion
(nicht öffentlich)

7.2.5 Statistik-Daten-Selektions-Parameterset-Req
(nicht öffentlich)

7.2.6 Statistik-Daten-Selektions-Parameterset-Ack
(nicht öffentlich)

7.2.7 Statistik-Daten-Report-Parameterset
(nicht öffentlich)

7.2.8 Statistik-Daten-Report
(nicht öffentlich)

7.2.9 Statistik-Daten-Report-Parameterset-Req
(nicht öffentlich)

7.2.10 Statistik-Daten-Report-Parameterset-Ack
(nicht öffentlich)
### 7.3 Physische Systemschnittstellen

Folgende Wireframes zeigen eine mögliche physische Repräsentation der logischen Datenstrukturen und die Schnittstellen des Systems.

#### 7.3.1 UC10, UC11, UC12 Statistik-Daten-Selektion

![Abbildung 11: Daten-Selektion](image-url)
### 7.3.2 UC20, UC21, UC23 Statistik-Daten-Report

![Report-Funktion](image)

**Abbildung 12: Report-Funktion**

### 7.3.3 UC30, UC31 Statistik-Daten-Tupel

![Daten-Eingabe](image)

**Abbildung 13: Daten-Eingabe**
8 Nicht-funktionale Systemanforderungen

8.1 Benutbarkeit

NF01: Die Durchführung der Anwendungsfälle sollte für einen Mitarbeiter mittels Tutorial innerhalb 30 Minuten selbstständig erlernbar sein.

NF02: Die Reports sollen komplette Artefakte darstellen, d.h. keine Nachbearbeitung oder Konvertierungen benötigen. Die genaue Form des Artefaks (z.B. pdf, Excel-Dokument) muss noch geklärt werden.

8.2 Verfügbarkeit

NF03: Der Zugriff auf das System muss während den Bürozeiten (7:00 – 18:00) gewährleistet sein.

NF04: Das System muss von 5 Benutzern gleichzeitig benutzt werden können.

8.3 Robustheit

NF05: Das System prüft die eingegebenen Daten auf Vollständigkeit. Sind die eingegebenen Daten nicht vollständig, wird der Benutzer darüber informiert.

8.4 Zuverlässigkeit

NF06: Die Fehlertoleranz des Systems gemessen in MTBF (Mean Time Between Failure) soll mindestens 30 Tage im ersten Betriebsjahr sein, danach ist eine Reevaluation notwendig.

NF07: Statistik-Daten, die vom System geprüft und verarbeitet wurden, müssen nach einem Betriebsausfall rekonstruierbar sein, sofern die Randbedingung geben ist, dass ein allfälliges Datenbank- und Betriebssystem gegen einen Betriebsausfall gesichert ist.

8.5 Effizienz

NF08: Eine Datenabfrage muss innerhalb 5 Sekunden eine gültige Antwort zurückgeben.

NF09: Eine Reportabfrage muss innerhalb 5 Sekunden eine gültige Antwort zurückgeben.

8.6 Systemsicherheit

NF10: Der Zugriff auf die Statistik-Daten ist rollenbasiert und mittels Passwort geschützt.

NF11: Das System darf nur von Mitarbeitern bedient und benutzt werden.

NF12: Das System darf nur in einem Netzwerk betrieben werden, welches vor Fremdzugriffen geschützt ist. (z.B. Firewall)

8.7 Änderbarkeit

NF13: Änderungen am System müssen innerhalb eines Arbeitstages verbreitet werden können. d.h. Deployment innerhalb eines Arbeitstages

8.8 Betriebliche Anforderungen

NF14: Das System muss auf einem virtuellen Microsoft Windows Server (ab 2008) lauffähig sein.

NF15: Für die Wartung des Systems dürfen nicht mehr als 12 Arbeitstage pro Jahr aufgewendet werden.
9 Klassenbeschreibungen

9.1 Einführung

Abfragen entsprechen im Wesentlichen einer Datenbank-Sicht (View), daher wurde das essenzielle Klassendiagramm ähnlich moduliert, wie das für Views von Datenbanken gilt.


Zudem wird das Statistik-Daten-Tupel um mindestens eine Statistik-Daten-Tupel-Modifikation erweitert, aus welcher ersichtlich ist, wann der Datenbank Eintrag entstanden ist oder zusätzlich modifiziert wurde.

9.2 Essenzielles Klassendiagramm

Abbildung 14: Essenzielles Klassendiagramm
9.3 Klassenbeschreibung

9.3.1 Statistik-Daten-Tupel

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>Objekte der Klasse Statistik-Daten-Tupel werden erstellt, wenn neue Statistik-Daten ins System eingegeben werden.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zustände</td>
<td>keine relevanten Zustände</td>
</tr>
<tr>
<td>Attribute</td>
<td>vgl. 7.2.1</td>
</tr>
</tbody>
</table>
| entgegennehmen() | Argument: DTO gemäss 7.2.1  
Antwort: Erfolgsmeldung  
Ablauf: 1) Übergebene Statistik-Daten werden überprüft.  
2a) Bei Erfolg wird ein Statistik-Daten-Tupel-Modifikations Objekt erstellt und sämtliche Informationen persistiert. Zudem wird eine positive Erfolgsmeldung abgesetzt.  
2b) Im Fehlerfall, wird eine negative Erfolgsmeldung abgesetzt. |

9.3.2 Statistik-Daten-Tupel-Modifikation

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>Die Klasse Statistik-Daten-Tupel-Modifikation erweitert die Klasse Statistik-Daten-Tupel dahingehend, dass eine Modifikation der Daten nachverfolgbar ist.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zustände</td>
<td>keine relevanten Zustände</td>
</tr>
</tbody>
</table>
| Attribute    | ModifikationsDatum: Date  
ModifikationsUser: String  
ModifikationsType: enum |
| Methode:     | -                                                                                                                                        |

9.3.3 Statistik-Daten-Vergleichs-Tupel

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Zustände</td>
<td>keine relevanten Zustände</td>
</tr>
<tr>
<td>Attribute</td>
<td>vgl. 7.2.1</td>
</tr>
<tr>
<td>Methode:</td>
<td>-</td>
</tr>
</tbody>
</table>

9.3.4 Statistik-Daten-Selektion

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>Objekte der Klasse Statistik-Daten-Selektion werden erstellt, wenn eine neue Statistik-Daten-Selektion angefordert wird. Sie basieren auf einem entsprechenden Statistik-Daten-Selektions-Parameterset und benützten eine Untermenge von Statistik-Daten-Tupel Objekten.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zustände</td>
<td>keine relevanten Zustände</td>
</tr>
<tr>
<td>Attribute</td>
<td>vgl. 7.2.4</td>
</tr>
</tbody>
</table>
| bekannt geben() | Argument: Referenz auf ein Statistik-Daten-Selektions-Parameterset  
Antwort: DTO gemäss 7.2.4  
Ablauf: 1) DTO wird gemäss dem referenzierten Statistik-Daten-Selektions-Parameterset erstellt.  
2) Das DTO wird dargestellt. |
| zum Download anbieten() | Argument: Referenz auf ein Statistik-Daten-Selektions Objekt  
Antwort: Download-Link  
Ablauf: 1) DTO wird serialisiert und in eine Datei ausgegeben  
2) Download Link wird bekanntgegeben |
### 9.3.5 Statistik-Daten-Selektions-Parameterset

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>Objekte der Klasse Statistik-Daten-Selektions-Parameterset werden erstellt, wenn eine neue Statistik-Daten-Auswahl entgegengenommen wird.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zustände</td>
<td>keine relevanten Zustände</td>
</tr>
<tr>
<td>Attribute</td>
<td>vgl. 7.2.3</td>
</tr>
</tbody>
</table>
| entgegennehmen() | Argument: DTO gemäss 7.2.3  
Antwort: Erfolgsmeldung  
Ablauf: 1) Übergebenes Statistik-Daten-Selektions-Parameterset wird überprüft.  
2a) Bei Erfolg wird ein Statistik-Daten-Selektions-Parameterset Objekt erstellt und sämtliche Informationen persistiert. Zudem wird eine positive Erfolgsmeldung abgesetzt.  
2b) Im Fehlerfall, wird eine negative Erfolgsmeldung abgesetzt. |
| verwenden()  | Argument: Referenz auf ein Statistik-Daten-Selektions-Parameterset  
Antwort: DTO gemäss 7.2.3  
Ablauf: einfache getter-Funktion |

### 9.3.6 Statistik-Daten-Report

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Zustände</td>
<td>keine relevanten Zustände</td>
</tr>
<tr>
<td>Attribute</td>
<td>vgl. 7.2.8</td>
</tr>
</tbody>
</table>
| bekannt geben() | Argument: Referenz auf ein Statistik-Daten-Report-Parameterset  
Antwort: DTO gemäss 7.2.8  
2) Das DTO wird dargestellt. |
| zum Download anbieten() | Argument: Referenz auf ein Statistik-Daten-Report Objekt  
Antwort: Download-Link  
Ablauf: 1) DTO wird serialisiert und in eine Datei ausgegeben  
2) Download Link wird bekanntgegeben |

### 9.3.7 Statistik-Daten-Report-Parameterset

<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>Objekte der Klasse Statistik-Daten-Report-Parameterset werden erstellt, wenn eine neue Report-Anfrage entgegengenommen wird.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zustände</td>
<td>keine relevanten Zustände</td>
</tr>
<tr>
<td>Attribute</td>
<td>vgl. 7.2.7</td>
</tr>
</tbody>
</table>
| entgegennehmen() | Argument: DTO gemäss 7.2.7  
Antwort: Erfolgsmeldung  
Ablauf: 1) Übergebenes Statistik-Daten-Report-Parameterset wird überprüft.  
2a) Bei Erfolg wird ein Statistik-Daten-Report-Parameterset Objekt erstellt und sämtliche Informationen persistiert. Zudem wird eine positive Erfolgsmeldung abgesetzt.  
2b) Im Fehlerfall, wird eine negative Erfolgsmeldung abgesetzt. |
| verwenden()  | Argument: Referenz auf ein Statistik-Daten-Report-Parameterset  
Antwort: DTO gemäss 7.2.7  
Ablauf: einfache getter-Funktion |
10 Randbedingungen an die Realisierung

10.1 Benutbarkeit
   RB01: (nicht öffentlich)

10.2 Verfügbarkeit
   RB02: (nicht öffentlich)

10.3 Robustheit
   (keine)

10.4 Zuverlässigkeit
   RB03: Sämtlich Software- und Hardwaresysteme die zum Betrieb des «Statistik-Daten Auswertungs»
   Systems benötigte werden müssen gegen Betriebsausfall gesichert sein.

10.5 Effizienz
   RB04: Die maximale Latenzzeit und die minimale Bandbreite, die gegeben sein müssen, sind noch zu
   bestimmen.

10.6 Systemsicherheit
   RB05: Das System darf nur von Mitarbeitern bedient und benutzt werden.
   RB06: Das System darf nur in einem Netzwerk betrieben werden, welches vor Fremdzugriffen geschützt ist.
   (z.B. Firewall)

10.7 Änderbarkeit
   RB07: (nicht öffentlich)

10.8 Betriebliche Anforderungen
   RB08: Virtueller Microsoft Windows Server
   RB09: Datenbanksystem (DBS) muss noch definiert werden.
Statistik-Daten Auswertung

Masterarbeit HSR-MAS-SE 2016

Architekturdokumentation
Inhaltsverzeichnis

1 Zu diesem Dokument ..................................................................................................................................... 7
  1.1 Zweck .................................................................................................................................................. 7
  1.2 Gültigkeit ............................................................................................................................................. 7
  1.3 Verfügbarkeit des Dokumentes .............................................................................................................. 7
  1.4 Graphische Darstellungen .................................................................................................................. 7
  1.5 Dokumentation der Programmerschnittstellen (Swagger) ................................................................. 7

2 Weiterführende Informationen ....................................................................................................................... 8
  2.1 Entscheidungen .................................................................................................................................... 8
  2.2 Definitionen, Akronymen und Abkürzungen ......................................................................................... 8
  2.3 Ergänzende Dokumente ....................................................................................................................... 8
  2.4 Prozessbezogene Dokumente ............................................................................................................... 8
  2.5 Versionshistorie .................................................................................................................................... 9
  2.6 Verteilung ............................................................................................................................................ 9

3 Einführung und Ziele ...................................................................................................................................... 10
  3.1 Aufgabenstellung ................................................................................................................................ 10
  3.1.1 Abgrenzung gegenüber dem ursprünglichen Projektantrag ............................................................ 10
  3.2 Stakeholder .......................................................................................................................................... 10
  3.3 Akteure ................................................................................................................................................ 11
  3.3.1 A1: Natürliche Person ....................................................................................................................... 11
  3.3.2 A2: Datenquellen ............................................................................................................................ 12
  3.3.3 A3: Verwaltungs-/Hostsysteme ...................................................................................................... 12
  3.4 Erweiterung der Anwendungsfälle gegenüber der Anforderungsspezifikation ........................................ 13
  3.4.1 UC40: Statistik-Daten-Alarmierung bekannt geben ....................................................................... 13
  3.4.2 UCS0: Statistik-Daten-Aktuator bekannt geben .............................................................................. 14
  3.4.3 UC60: API-Informationen bekannt geben ....................................................................................... 15
  3.4.4 UC100: Benutzerregistrierung entgegennehmen ............................................................................. 15
  3.4.5 UC101: Benutzer-Rollen-Wechsel entgegennehmen ...................................................................... 15
  3.5 Übersicht Anwendungsfälle .................................................................................................................. 16
  3.6 Qualitätsziele / Qualitätsanforderungen .................................................................................................. 18

4 Randbedingungen ......................................................................................................................................... 20

5 Kontextabgrenzung ...................................................................................................................................... 21
  5.1 Fachlicher Kontext ............................................................................................................................... 21
  5.2 Technischer Kontext ............................................................................................................................ 23

6 Lösungsstrategien ......................................................................................................................................... 24
  6.1 Funktionale Lösungen .......................................................................................................................... 24
  6.1.1 Aufteilung in «AccessCoordination» und «DataPersistence»-Modul .................................................. 24
<table>
<thead>
<tr>
<th>Kapitel</th>
<th>Thema</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.2</td>
<td>Statistik-Daten-Aktor als Basis der Datenbank</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Statistik-Daten-Abfrage-Parameterset können gespeichert und verwaltet werden</td>
</tr>
<tr>
<td>6.1.4</td>
<td>Statistik-Daten-Selektion für mehrere Aktuatoren</td>
</tr>
<tr>
<td>6.2</td>
<td>Schnittstellen</td>
</tr>
<tr>
<td>6.2.1</td>
<td>http als Kommunikationsprotokoll</td>
</tr>
<tr>
<td>6.2.2</td>
<td>REST</td>
</tr>
<tr>
<td>6.3</td>
<td>Technologie</td>
</tr>
<tr>
<td>6.3.1</td>
<td>.Net, C#, ASP.NET Core 2.0 und Entity Framework Core</td>
</tr>
<tr>
<td>6.3.2</td>
<td>SQlite als Datenbanksystem</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Verteilung als «Self-Contained Windows Service»</td>
</tr>
<tr>
<td>7</td>
<td>Programmierschnittstelle des Gesamtsystems</td>
</tr>
<tr>
<td>7.1</td>
<td>REST-Ressourcen</td>
</tr>
<tr>
<td>7.1.1</td>
<td>StatisticDataActuators</td>
</tr>
<tr>
<td>7.1.2</td>
<td>StatisticDataInformation</td>
</tr>
<tr>
<td>7.1.3</td>
<td>StatisticDataQueryParameterset</td>
</tr>
<tr>
<td>7.1.4</td>
<td>StatisticDataTuple</td>
</tr>
<tr>
<td>7.2</td>
<td>Modelle</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Benutzte Modelle (für Anfrage) bei Anwendungsfällen</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Benutzte Modelle (für Antworten) bei Anwendungsfällen</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Aufbau: Statistik-Daten-Tupel / «StatisticDataTuple»</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Aufbau: Statistik-Daten-Abfrage-Parameterset / «StatisticDataQueryParameterset»</td>
</tr>
<tr>
<td>7.2.5</td>
<td>Statistik-Daten-Aktor: «StatisticDataActuator»</td>
</tr>
<tr>
<td>7.2.6</td>
<td>Statistik-Daten-Selektion: «StatisticDataSelection»</td>
</tr>
<tr>
<td>7.2.7</td>
<td>Statistik-Daten-Report: «StatisticDataReport»</td>
</tr>
<tr>
<td>7.2.8</td>
<td>Statistik-Daten-Alarmierung: «StatisticDataAlarm»</td>
</tr>
<tr>
<td>8</td>
<td>Typische Strukturen und übergreifende Konzepte</td>
</tr>
<tr>
<td>8.1</td>
<td>REST</td>
</tr>
<tr>
<td>8.2</td>
<td>HATEOAS (Hypertext As The Engine Of Application State)</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Vergleich der System-Schnittstellen</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Implementierung</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Interne Schnittstelle</td>
</tr>
<tr>
<td>8.3</td>
<td>Datenpersistenz in der Datenbank</td>
</tr>
<tr>
<td>8.4</td>
<td>Vorgaben von ASP.NET Core 2.0</td>
</tr>
<tr>
<td>8.4.1</td>
<td>«Konventionen vor Konfiguration»</td>
</tr>
<tr>
<td>8.4.2</td>
<td>«Model View Controller»-Muster</td>
</tr>
<tr>
<td>8.4.3</td>
<td>«Model View Controller»-Muster für Programmierschnittstellen</td>
</tr>
<tr>
<td>8.4.4</td>
<td>Dependency Injection</td>
</tr>
<tr>
<td>8.4.5</td>
<td>LINQ / Entity Framework Core</td>
</tr>
<tr>
<td>8.4.6</td>
<td>Konfiguration in Dateien</td>
</tr>
<tr>
<td>8.4.7</td>
<td>Log-Daten</td>
</tr>
<tr>
<td>9</td>
<td>Bausteinsicht</td>
</tr>
<tr>
<td>9.1</td>
<td>Übersicht (Level 1)</td>
</tr>
<tr>
<td>9.2</td>
<td>«AccessCoordination» (Level 2)</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Abhängigkeiten gegenüber dem ASP.NET Framework</td>
</tr>
</tbody>
</table>
9.3 DataPersistence (Level 2) ............................................................................................. 48
9.3.1 Abhängigkeiten gegenüber dem ASP.NET Framework ........................................... 49
9.4 AccessCoordination (Level 3) ....................................................................................... 50
9.5 DataPersistence (Level 3) ............................................................................................. 51
9.6 Datenmodelle «AccessCoordination» ........................................................................... 52
9.6.2 «InformationApiModel» ........................................................................................... 52
9.6.3 Statistik-Daten-Aktuator / «StatisticDataActuatorApiModel» .................................. 52
9.6.4 Statistik-Daten-Abfrage-Parameterset / «StatisticDataQueryParameterset» ......... 52
9.6.5 Statistik-Daten-Selektion / -Report / -Alarmierung / «StatisticDataQuery» ............. 54
9.6.6 Statistik-Daten-Tupel / «StatisticDataTuple» ............................................................. 54
9.7 Datenmodelle «DataPersistence» ................................................................................... 55
9.7.1 «InformationApiModel» .......................................................................................... 55
9.7.2 Statistik-Daten-Aktuator / «StatisticDataActuator» ................................................. 55
9.7.3 Statistik-Daten-Abfrage-Parameterset / «StatisticDataQueryParameterset» .......... 56
9.7.4 Statistik-Daten-Selektion / -Report / -Alarmierung / «StatisticDataQuery» .......... 57
9.7.5 Statistik-Daten-Tupel / «StatisticDataTuple» ............................................................. 58

10 Laufzeitsicht .................................................................................................................... 59
10.1 Services ....................................................................................................................... 59
10.1.1 Services «AccessCoordination» ............................................................................ 59
10.1.2 Services «DataPersistence» ................................................................................... 60
10.2 Hypermedia As The Engine Of Application State (HATEOAS) ..................................... 61
10.2.1 HATEOAS «AccessCoordination» ......................................................................... 61
10.2.2 HATEOAS «DataPersistence» ............................................................................... 63

11 Verteilungssicht .............................................................................................................. 65
11.1 Überblick ..................................................................................................................... 65
11.2 Abhängigkeiten und Kommunikation ........................................................................ 66
11.3 Konfiguration verteilter Anwendungen ....................................................................... 67
11.4 Datenbanken .............................................................................................................. 68
11.4.1 AccessCoordinationDb ........................................................................................ 68
11.4.2 DataPersistenceDb ............................................................................................... 69

12 Testinfrastruktur .......................................................................................................... 70
12.1 Unitest ....................................................................................................................... 70
12.2 Komponententest ...................................................................................................... 71
12.3 Integrationstest .......................................................................................................... 72
12.4 Systemtest .................................................................................................................. 73

13 Toolchain ....................................................................................................................... 74
13.1 Enterprise Architect, Sparx Systems, v7.1.832 ........................................................... 74
13.2 Swagger ....................................................................................................................... 74
13.3 SQLiteBrowser ........................................................................................................... 74
13.4 .NET Runtime und SDK ............................................................................................ 74
13.5 Visual Studio 2017, Microsoft
13.5.1 Zusätzliche Plug-Ins zu Visual Studio

14 Entwurfsentscheidungen
14.1 Abgrenzende Entscheidungen
14.1.1 EA01
14.1.2 EA02
14.1.3 EA03
14.2 Funktionale Entscheidungen
14.2.1 EF01
14.2.2 EF02
14.2.3 EF03
14.2.4 EF04
14.2.5 EF05
14.3 Technologische Entscheidungen
14.3.1 ET01
14.3.2 ET02
14.3.3 ET03
14.3.4 ET04

15 Risiken und technische Schulden
15.1 Funktionale Risiken und technische Schulden
15.2 Nicht-funktionale Risiken und technische Schulden
15.3 Infrastruktur Risiken und technische Schulden

16 Anhang
16.1 Statistik-Daten-Tupel
Abbildungsverzeichnis
Abbildung 1: Übersicht der Anwendungsfälle ................................................................. 17
Abbildung 2: Fachlicher Kontext .................................................................................. 21
Abbildung 3: Technischer Kontext ................................................................................ 23
Abbildung 4: Zustandsmaschine des Webinterfaces ..................................................... 34
Abbildung 5: Zustandsmaschine der Programmierschnittstelle ................................. 35
Abbildung 6: Zustandsmaschine der Ressource «/api/StatisticDataTuple» ................... 35
Abbildung 7: Klasseendiagramm des «HateosLinkHelpers» ........................................ 36
Abbildung 8: Zustandsmaschine der internen Programmierschnittstelle ..................... 36
Abbildung 9: MVC Muster in Web-Anwendungen ....................................................... 38
Abbildung 10: Muster für eine Programmierschnittstelle ........................................... 39
Abbildung 11: Aufruf der Methode «GetStatisticDataActuators()» ......................... 40
Abbildung 12: Klassen-Diagramm des «StatisticDataActuatorsControllers» .......... 41
Abbildung 13: Komponenten-Diagramm des Gesamtsystems / Level 1 ........................ 43
Abbildung 14: Paket-Diagramm «AccessCoordination» ............................................ 44
Abbildung 15: Abhängigkeiten gegenüber «NuGet-Packages» des ASP.NET Frameworks .... 46
Abbildung 16: Paket-Diagramm “DataPersistence” ...................................................... 48
Abbildung 17: Abhängigkeit gegenüber Bibliotheken des ASP.NET Frameworks ........ 49
Abbildung 18: Zusammenhang zwischen «Controller» und «Service» ....................... 50
Abbildung 19: Zusammenhang zwischen «Controller» und «Service» ....................... 51
Abbildung 20: Klasseendiagramm «StatisticDataQueryParameterset» .................... 53
Abbildung 21: Klasseendiagramm «StatisticDataQuery» ........................................... 54
Abbildung 22: Klasseendiagramm «StatisticDataActuator» ....................................... 55
Abbildung 23: Klasseendiagramm «StatisticDataQueryParameterset» ..................... 56
Abbildung 24: Klasseendiagramm «StatisticDataQuery» ........................................... 57
Abbildung 25: «/api» .................................................................................................. 61
Abbildung 26: «/api/StatisticDataTuple» .................................................................... 62
Abbildung 27: «/api/StatisticDataActuator» ............................................................. 62
Abbildung 28: «/api/StatisticDataQueryParametersets» .......................................... 62
Abbildung 29: «internalApi» ..................................................................................... 63
Abbildung 30: «internalApi/StatisticDataTuple» ....................................................... 63
Abbildung 31: «internalApi/StatisticDataActuator» .................................................. 63
Abbildung 32: «internalApi/StatisticDataSelection» .................................................. 64
Abbildung 33: «internalApi/StatisticDataReport» ...................................................... 64
Abbildung 34: «internalApi/StatisticDataSelection» .................................................. 64
Abbildung 35: Gesamtübersicht über die Verteilung des Systems ............................. 65
Abbildung 36: «self-contained service» .................................................................... 66
Abbildung 37: Datenbankschema der «AccessCoordinationDb» ............................... 68
Abbildung 38: Reduziertes Schema des «StatisticDataActuator» .............................. 69
Abbildung 39: Aufbau eines Unitests ............................................................................ 70
Abbildung 40: Aufbau eines Komponententests ....................................................... 71
Abbildung 41: Aufbau eines Integrationstests ............................................................. 72
Abbildung 42: Aufbau eines Systemtests .................................................................... 73
1 Zu diesem Dokument

1.1 Zweck

Die Architekturdokumentation hilft Stakeholdern einerseits die Architektur des Systems zu verstehen und andererseits dokumentiert sie die zugrunde liegenden Entscheidungen und Konzepte. Zum Schluss listet sie offene Punkte und Risiken, die in Zukunft weiterverfolgt werden müssen.

Die Form und Gliederung der Architekturdokumentation folgt dem arc42-Template: https://arc42.org/

1.2 Gültigkeit

Dieses Dokument ist über die ganze Projektdauer gültig.

1.3 Verfügbarkeit des Dokumentes

Dieses Dokument befindet sich im Projektverzeichnis: 11_Architekturdokumentation

1.4 Graphische Darstellungen


Die Datei (StatisticDataAnalysis.eap) befindet sich im Projektverzeichnis: 11_Architekturdokumentation

Eine html-Version des EA-Projekts befindet sich im Projektverzeichnis: 11_Architekturdokumentation\EA_Doc

1.5 Dokumentation der Programmierschnittstellen (Swagger)

Eine interaktive Dokumentation der Programmierschnittstellen befindet sich im Projektverzeichnis: 11_Architekturdokumentation\Swagger_Doc\AccessCoordination

11_Architekturdokumentation\Swagger_Doc\DataPersistence
2 Weiterführende Informationen

2.1 Entscheidungen

Entscheidungen (z.B. Änderungen betreffend Projektantrag oder Anforderungsspezifikation) werden wie folgt im Text markiert und sind im Kapitel 14 erläutert:

<table>
<thead>
<tr>
<th>EAxx</th>
<th>Abgrenzende Entscheidungen</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFxx</td>
<td>Funktionale Entscheidungen</td>
</tr>
<tr>
<td>ETxx</td>
<td>Technologische Entscheidungen</td>
</tr>
</tbody>
</table>

2.2 Definitionen, Akronyme und Abkürzungen

Sämtliche Definitionen, Akronyme und Abkürzungen werden zentral in einem Projektglossar dokumentiert. Das Projektglossar wird kontinuierlich angepasst.

Das Projektglossar befindet sich im Projektverzeichnis: 09_Glossar

2.3 Ergänzende Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

2.4 Prozessbezogene Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>
2.5 Versionshistorie

(nicht öffentlich)

2.6 Verteilung

<table>
<thead>
<tr>
<th>Ver.</th>
<th>Name</th>
<th>Rolle / Titel / Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>006</td>
<td>MAS-Arbeit</td>
<td>MAS-Arbeit</td>
</tr>
</tbody>
</table>
3 Einführung und Ziele

Dieses Kapitel behandelt nochmals die Themen der Anforderungsspezifikation und erweitert oder korrigiert Punkte, die bei der Freigabe der Anforderungsspezifikation noch nicht geklärt waren, oder aber aufgrund von Architekturentscheidungen geändert werden mussten.

3.1 Aufgabenstellung


3.1.1 Abgrenzung gegenüber dem ursprünglichen Projektantrag

<table>
<thead>
<tr>
<th>EA01</th>
<th>Keine direkte Anbindung von Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA02</td>
<td>Event-Log Daten werden nicht in das System übernommen</td>
</tr>
</tbody>
</table>

3.2 Stakeholder

(nicht öffentlich)
3.3 Akteure

Bei der Implementierung hat sich gezeigt, dass die Aufteilung der Akteure nicht ideal gewählte wurde. Die nachfolgende Aufteilung unterscheidet sich daher von der Anforderungsspezifikation.

<table>
<thead>
<tr>
<th>Rolle</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guest</td>
<td>Ein nicht registrierter Anwender. Er hat keine Rollen und kann (außer der Registrierung) keine Anwendungsfälle ausführen.</td>
</tr>
<tr>
<td>Administrator</td>
<td>Der Administrator verwaltet und konfiguriert das System. Er teilt anderen Akteuren ihre Rollen zu.</td>
</tr>
<tr>
<td>QueryManipulator</td>
<td>Der QueryManipulator erstellt, editiert oder löscht Statistik-Daten-Abfrage-Parametersets.</td>
</tr>
<tr>
<td>DataManipulator</td>
<td>Der DataManipulator erstellt, importiert und verändert einzelne Statistik-Daten-Tupel.</td>
</tr>
</tbody>
</table>

**Rollenzuteilung**

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Guest</th>
<th>Administrator</th>
<th>Consumer</th>
<th>Query-Manipulator</th>
<th>Data-Manipulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entwickler (S3, S4)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Supporter (S5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Test-Verifikation-Validation-Team (S6)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Feldtest-Manager (S6)</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Projektleiter (S7)</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EF03: Für die Implementierung der Rollen musste eine Benutzerverwaltung mit Authentifizierung und Autorisierung eingeführt werden. Dadurch entstand die Rolle «Guest».
3.3.2 A2: Datenquellen


A2a: DataProducer: Legacy
Das Legacy-System liefert sämtliche erfassten Statistik-Daten-Tupels. Es funktioniert als Pipeline zwischen Hardware und «Statistik-Daten Auswertung»-System.

A2b: DataProducer: FileSystem
Für Testzwecke sollen Statistik-Daten-Tupel via Dateisystem importiert werden können.

3.3.3 A3: Verwaltungs-/Hostsysteme

(nicht öffentlich)

A3a: DataFeeder

A3b: RawDataConsumer

A3c: InformationConsumer
### 3.4 Erweiterung der Anwendungsfälle gegenüber der Anforderungsspezifikation

#### 3.4.1 UC40: Statistik-Daten-Alarmierung bekannt geben


<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC40</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>Statistik-Daten-Alarmierung bekannt geben</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Tief</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Projektantrag [1]</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>Auf Anfrage gibt das System dem Akteur die gewünschte Statistik-Daten-Alarmierung bekannt.</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Akteur setzt mittels Statistik-Daten-Abfrage-Parameterset eine Anfrage ab.</td>
</tr>
</tbody>
</table>
| Akteure                  | A1c: Consumer  
A3c: InformationConsumer                                                   |
| Datenstruktur (Anfrage):| Statistik-Daten-Abfrage-Parameterset                                  |
| Akteur  System          | Statistik-Daten-Alarmierung                                             |
| Datenstruktur (Antwort): | System  Akteur                                                        |
| Vorbedingungen           | -                                                                      |
| Nachbedingungen          | -                                                                      |
| Ergebnis                 | Akteur nimmt gewünschte Statistik-Daten-Alarmierung entgegen.          |

**Hauptszenario**

1) Anfrage zur Bekanntgabe einer Statistik-Daten-Alarmierung durch den Akteur.  
2) System verwendet das gesendete Statistik-Daten-Abfrage-Parameterset zur Erstellung der gewünschten Statistik-Daten-Alarmierung.  
3) System gibt die Statistik-Daten-Alarmierung dem Akteur bekannt.

**Alternativszenarien**

-  

**Ausnahmeszenarien**

vgl. UC10 und UC20 der Anforderungsspezifikation [3]  

**Fehlerfall**

vgl. UC10 und UC20 der Anforderungsspezifikation [3]
## 3.4.2 UC50: Statistik-Daten-Aktuator bekannt geben

### Abschnitt | Inhalt
---|---
Bezeichner | UC50
Anwendungsfallname | Statistik-Daten-Aktuator bekannt geben
Autoren | Projektleitung
Prioritäten | Mittel
Kritikalität | Mittel
Quelle | Meeting_Minutes_20180514.docx
Verantwortlicher | Projektleitung
Kurzbeschreibung | Auf Anfrage gibt das System dem Akteur sämtliche Daten eines Aktuators bekannt.
Auslösendes Ereignis | Akteur setzt mittels Seriennummer eine Anfrage ab.
Akteure | A1c: Consumer
| A3b: RawDataConsumer
Datenstruktur (Anfrage): | Akteur $\rightarrow$ System
| Seriennummer
Datenstruktur (Antwort): | System $\rightarrow$ Akteur
| sämtliche Statistik-Daten eines Aktuators (Statistik-Daten-Aktuator)
Vorbedingungen | -
Nachbedingungen | -
Ergebnis | Akteur nimmt gewünschte Statistik-Daten des Aktuators entgegen.
Hauptszenario | 1) Anfrage zur Bekanntgabe der Statistik-Daten eines Aktuators durch den Akteur.  
2) System verwendet die gesendete Seriennummer und gibt die Statistik-Daten des Aktuators bekannt.
Alternativszenarien | -
Ausnahmeszenarien | 1) Anfrage zur Bekanntgabe der Statistik-Daten eines Aktuators durch den Akteur.  
2) Fehlerfall!  
3) System gibt dem Akteur bekannt, dass die Statistik-Daten des Aktuators nicht verarbeitet werden konnte.
Fehlerfall | • Verbindungsfehler können detektiert und dem Benutzer bekannt gegeben werden.  
• Eine unbekannte Seriennummer kann detektiert und dem Benutzer bekannt gegeben werden.  
• Eine nicht-verarbeitbare Seriennummer kann detektiert und dem Benutzer bekannt gegeben werden.
3.4.3 UC60: API-Informationen bekannt geben

<table>
<thead>
<tr>
<th>Abschnitt</th>
<th>Inhalt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bezeichner</td>
<td>UC60</td>
</tr>
<tr>
<td>Anwendungsfallname</td>
<td>API-Information bekannt geben</td>
</tr>
<tr>
<td>Autoren</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Prioritäten</td>
<td>Mittel</td>
</tr>
<tr>
<td>Kritikalität</td>
<td>Mittel</td>
</tr>
<tr>
<td>Quelle</td>
<td>Keine</td>
</tr>
<tr>
<td>Verantwortlicher</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Kurzbeschreibung</td>
<td>Auf Anfrage gibt das System dem Akteur die API-Informationen bekannt.</td>
</tr>
<tr>
<td>Auslösendes Ereignis</td>
<td>Triviale Anfrage auf Ressource</td>
</tr>
</tbody>
</table>

**Akteure**

- A3a: DataFeeder
- A3b: RawDataConsumer
- A3c: InformationConsumer

**Datenstruktur (Anfrage):**

<table>
<thead>
<tr>
<th>Akteur</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keine (einfache GET Anfrage)</td>
<td></td>
</tr>
</tbody>
</table>

**Datenstruktur (Antwort):**

<table>
<thead>
<tr>
<th>System</th>
<th>Akteur</th>
</tr>
</thead>
<tbody>
<tr>
<td>API-Informationen</td>
<td></td>
</tr>
</tbody>
</table>

**Vorbedingungen:**

- 

**Nachbedingungen:**

- 

**Ergebnis**

| Akteur nimmt API-Informationen entgegen. |

**Hauptszenario**

1) Anfrage zur Bekanntgabe der API-Informationen durch den Akteur.  
2) System gibt API-Informationen bekannt.

**Alternativszenarien**

- 

**Ausnahmeszenarien**

1) Anfrage zur Bekanntgabe der API-Informationen durch den Akteur.  
2) Fehlerfall!  
3) System gibt dem Akteur bekannt, dass die Anfrage nicht verarbeitet werden konnte.

**Fehlerfall**

- Verbindungsfehler können detektiert und dem Benutzer bekannt gegeben werden.

3.4.4 UC100: Benutzerregistrierung entgegennehmen

(siehe unten)

3.4.5 UC101: Benutzer-Rollen-Wechsel entgegennehmen

Diese beiden Anwendungsfälle wurden durch die Implementierung der Benutzerverwaltung mit Authentifizierung und Autorisierung eingeführt. Sie sind der Vollständigkeit halber aufgeführt. Die Architektur beruht auf der Dokumentation von Microsoft. Aus diesem Grund wird hier auf die Dokumentation von Microsoft verwiesen:

**Authentifizierung:** [https://docs.microsoft.com/en-us/aspnet/core/security/authentication/](https://docs.microsoft.com/en-us/aspnet/core/security/authentication/)

**Autorisierung:** [https://docs.microsoft.com/en-us/aspnet/core/security/authorization/](https://docs.microsoft.com/en-us/aspnet/core/security/authorization/)
3.5 Übersicht Anwendungsfälle


UC21 (Report) wurde in UC11 (Selektion) integriert, welcher zudem auch für die Handhabung von Alarmierungen verwendet werden kann. Das Gleiche gilt für UC12 (Report) und UC22 (Selektion).

<table>
<thead>
<tr>
<th>Das System muss folgende Anwendungsfälle erfüllen können</th>
<th>Akteur</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC10: Statistik-Daten-Selektion bekannt geben</td>
<td>A1c, A3c</td>
</tr>
<tr>
<td>UC11: Alle Statistik-Daten-Abfrage-Parametersets bekannt geben</td>
<td>A1c, A3c</td>
</tr>
<tr>
<td>UC12: Statistik-Daten-Abfrage-Parameterset entgegennehmen</td>
<td>A1d</td>
</tr>
<tr>
<td>UC20: Statistik-Daten-Report bekannt geben</td>
<td>A1c, A3c</td>
</tr>
<tr>
<td>UC40: Statistik-Daten-Alarmierung bekannt geben</td>
<td>A1c, A3c</td>
</tr>
<tr>
<td>UC30: Statistik-Daten-Tupel entgegennehmen</td>
<td>A1e, A2a, A2b, A3a</td>
</tr>
<tr>
<td>UC31: Modifiziertes Statistik-Daten-Tupel entgegennehmen</td>
<td>A1e</td>
</tr>
<tr>
<td>UC50: Statistik-Daten-Aktuator bekannt geben</td>
<td>A1c, A3b</td>
</tr>
<tr>
<td>UC60: API-Informationen bekannt geben</td>
<td>A3a, A3b, A3c</td>
</tr>
<tr>
<td>UC100: Benutzerregistrierung entgegennehmen</td>
<td>A1a</td>
</tr>
<tr>
<td>UC101: Benutzer-Rollen-Wechsel entgegennehmen</td>
<td>A1b</td>
</tr>
</tbody>
</table>

Das folgende Diagramm (nächste Seite) ist eine Übersicht der Anwendungsfälle:
Abbildung 1: Übersicht der Anwendungsfälle
### 3.6 Qualitätsziele / Qualitätsanforderungen

Die nicht-funktionalen Systemanforderungen der Anforderungspezifikation (Kapitel 8) sind ein wesentlicher Teil der Qualitätsziele. Nicht-funktionale Systemanforderungen, die einen direkten Einfluss auf die Architektur haben, wurden daher in diese Liste übernommen und mit weiteren Punkten ergänzt.

Nicht alle Qualitätsanforderungen können durch die jetzige Architektur erfüllt werden. Ziele wäre aber, dass die hier gelisteten Qualitäten den Weg für die Weiterentwicklung aufzeigen.

**Bemerkung:** Einträge, die mit «new» gekennzeichnet sind, konnten bei der Implementierung aus Zeitgründen nicht berücksichtigt werden.

<table>
<thead>
<tr>
<th>Qualitätsziel</th>
<th>Erläuterung</th>
<th>Stand</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZ01 NF10</td>
<td>Funktionalität</td>
<td>Der Zugriff auf die Statistik-Daten ist rollenbasiert und mittels Passwort geschützt.</td>
</tr>
<tr>
<td>QZ01.1</td>
<td>Anbindung an das Active Directory wäre wünschenswert.</td>
<td>new</td>
</tr>
<tr>
<td>QZ01.2</td>
<td>Rollen müssen eindeutig zugeteilt werden können. Es wird daher eine Benutzerverwaltung mit Authentifizierung und Autorisierung benötigt.</td>
<td></td>
</tr>
<tr>
<td>QZ01.3</td>
<td>Rollen müssen zugeteilt und entfernt werden können.</td>
<td></td>
</tr>
<tr>
<td>QZ01.4</td>
<td>Neue Rolle sollten erzeugt werden können.</td>
<td></td>
</tr>
<tr>
<td>QZ02</td>
<td>Funktionalität</td>
<td>Gespeicherte Statistik-Daten-Abfrage-Parametersets müssen permanent gespeichert und für alle autorisierten Benutzer sichtbar sein.</td>
</tr>
<tr>
<td>QZ02.1</td>
<td>Statistik-Daten-Abfrage-Parametersets können gruppiert und spezifisch an Benutzergruppen zugewiesen werden.</td>
<td>new</td>
</tr>
<tr>
<td>QZ03 NF04</td>
<td>Multisession tauglich</td>
<td>Das System muss von 5 Benutzern gleichzeitig benutzt werden können.</td>
</tr>
<tr>
<td>QZ03.1</td>
<td>Benutzer können Daten-Importe persönlich verwalten.</td>
<td></td>
</tr>
<tr>
<td>QZ03.2</td>
<td>System muss verhindern, dass Benutzer zur gleichen Zeit die gleichen Daten manipulieren (z.B. Statistik-Daten-Abfrage-Parameterset).</td>
<td>new</td>
</tr>
<tr>
<td>QZ04 NF05</td>
<td>Robustheit</td>
<td>Das System prüft die eingegebenen Daten auf Vollständigkeit. Sind die eingegebenen Daten nicht vollständig, wird der Benutzer darüber informiert.</td>
</tr>
<tr>
<td>QZ04.1</td>
<td>Benutzereingaben werden bereits bei der Eingabe geprüft. Client-Validation</td>
<td></td>
</tr>
<tr>
<td>QZ04.2</td>
<td>Benutzereingaben werden bei der Verarbeitung geprüft. Server-Validation</td>
<td>new</td>
</tr>
</tbody>
</table>
QZ05  NF07  Zuverlässigkeit  Statistik-Daten, die vom System geprüft und verarbeitet wurden, müssen nach einem Betriebsausfall rekonstruierbar sein, sofern die Randbedingung geben ist, dass ein allfälliges Datenbank- und Betriebssystem gegen einen Betriebsausfall gesichert ist.

QZ05.1  Die Anwendung verwalten nur minimale Nutzerdaten des Session-States im Speicher des Clients oder Servers. Ziel ist es, dass der Session-State in der Datenbank verwaltet wird.

QZ06  NF08  Effizienz  Eine Datenabfrage muss innerhalb von 5 Sekunden eine gültige Antwort zurückgeben.

QZ07  NF09  Effizienz  Eine Reportabfrage muss innerhalb von 5 Sekunden eine gültige Antwort zurückgeben.

QZ08  Sicherheit  Der Inhalt der Legacy Datenbank darf durch das «Statistik-Daten Auswertung»-System nicht verändert werden.

QZ08.1  Die Legacy Datenbank verwaltet weiterhin die Rohdaten, dadurch wird das Gesamtsystem nicht eingeschränkt, sondern erweitert.

QZ09  Datenintegrität  Neue Statistik-Daten-Tupel werden bestehenden Aktuatoren korrekt zugewiesen. Falls die Seriennummer des Statistik-Daten-Tupel noch nicht in der Datenbank existiert, wird ein neuer Aktuator angelegt.

QZ09.1  Für spezielle Anwendungsfälle soll eine Verknüpfung zwischen zwei Seriennummer möglich sein. New

QZ10  Änderbarkeit  Änderungswünsche zu einem späteren Zeitpunkt (nach Pilotbetrieb) müssen möglich sein. Open-Closed-Prinzip

QZ11  NF06  Stabilität  Die Fehlertoleranz des Systems gemessen in MTBF (Mean Time Between Failure) soll mindestens 30 Tage im ersten Betriebsjahr sein

QZ12  Benutzbarkeit  Das System soll im Fehlerfall in einen definierten Zustand übergehen. (z.B. Error-Seite)
Randbedingungen

Die in der Anforderungsspezifikation gelisteten Randbedingungen (Kapitel 10) wurden neu beurteilt. Die dort genannten Randbedingungen betreffen grössten Teils den Betrieb und die Umsysteme des Systems und haben daher wenig Einfluss auf die Architektur. Es gibt aber sehr wohl Randbedingungen, die beachtet werden müssen:

<table>
<thead>
<tr>
<th>Id</th>
<th>Randbedingung</th>
<th>Erläuterung</th>
</tr>
</thead>
<tbody>
<tr>
<td>RB11</td>
<td>Analysierbarkeit</td>
<td>Es muss möglich sein, Fehler die im Betrieb auftreten, zu analysieren. Eine Möglichkeit ist das Loggen in Dateien.</td>
</tr>
<tr>
<td>RB12</td>
<td>Technologie</td>
<td>Die Anzahl eingesetzter Technologien soll niedrig gehalten werden.</td>
</tr>
<tr>
<td>RB14</td>
<td>Interoperabilität</td>
<td>Das «Statistik-Daten Auswertung»-System darf andere Webservices nicht beeinflussen, d.h. entweder zu Apache HTTP Server kompatibel sein oder aber Apache nicht an der Ausführung hindern</td>
</tr>
</tbody>
</table>
5 Kontextabgrenzung

5.1 Fachlicher Kontext

Abbildung 2: Fachlicher Kontext
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>
| P1  | BrowserInterface            | Typ: Provided Interface, Akteure greifen darauf zu  
Implementierung: Browser Schnittstelle (Webinterface)  
Protokoll: http  
| P2  | /api/StatisticDataInformation | Typ: Provided Interface, externe Systeme greifen darauf zu  
Implementierung: REST-API Ressource  
Protokoll: http  
Hauptzweck: Über dieses Interface können externe Systeme Information zu weiteren REST-API Ressourcen abfragen.                                                                                                                                                                                                                     |
| P3  | /api/StatisticDataTuple     | Typ: Provided Interface, externe Systeme greifen darauf zu  
Implementierung: REST-API Ressource  
Protokoll: http  
Hauptzweck: Über dieses Interface können externe Systeme neue Daten in das System einspeisen.                                                                                                                                                                                                                                             |
| P4  | /api/StatisticDataActuators | Typ: Provided Interface, externe Systeme greifen darauf zu  
Implementierung: REST-API Ressource  
Protokoll: http  
Hauptzweck: Über dieses Interface können externe Systeme die Aktuatorkenngrössen auslesen.                                                                                                                                                                                                                                 |
| P5  | /api/StatisticDataQueryParameterset | Typ: Provided Interface, externe Systeme greifen darauf zu  
Implementierung: REST-API Ressource  
Protokoll: http  
Hauptzweck: Über diese Schnittstelle können externe Systeme einerseits Query-Parametersets verwalten und andererseits mittels dieser Query-Parametersets Statistik-Daten in Form von Selektionen, Reports oder Alarmierung abfragen.                                                                                   |
| R1  | Legacy                      | Typ: Required Interface, für Daten-Import benötigt  
Implementierung: Datenbank-Zugriff  
Protokoll: TCP/IP  
Hauptzweck: Über diese Schnittstelle greift das System auf das bestehende Legacy-Data-Repository zu.                                                                                                                                                                                                 |
| R2  | FileSystem                  | Typ: Required Interface, für Entwicklung und Test benötigt  
Implementierung: .NET-Framework  
Protokoll: filesystem  
Hauptzweck: Über diese Schnittstelle werden Daten aus Dateien gelesen.                                                                                                                                                                                                                                                                       |

Eine ausführliche Beschreibung der Programmierschnittstelle folgt in Kapitel 7.
5.2 Technischer Kontext

Abbildung 3: Technischer Kontext

<table>
<thead>
<tr>
<th>Name</th>
<th>Protokoll</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>BrowserInterface</td>
<td>http</td>
<td>Verbindung zum Gesamtsystem, die ein Anwender mittels Webbrowser (Client) öffnet.</td>
</tr>
<tr>
<td>internalApi</td>
<td>http</td>
<td>Verbindung zwischen den beiden unabhängigen Modulen «AccessCoordination» und «DataPersistence». Im Diagramm sind beide «Self-Contained Services» dem gleichen Device zugeteilt. Die Services könnten auch auf eigene Devices verteilt sein.</td>
</tr>
<tr>
<td>DB1a</td>
<td>TCP/IP</td>
<td>Verbindung zur externen Legacy-Datenbank, welche auf einem dezidierten Datenbanks-Server betrieben wird.</td>
</tr>
<tr>
<td>DB1b</td>
<td>Filesystem</td>
<td>Die Daten werden vom «DataPersistence» Modul in einer SQLite Datenbank verwaltet, welche direkt in die Anwendung integriert ist und über Dateisystem-Mechanismen des jeweiligen Betriebssystems verwaltet wird.</td>
</tr>
</tbody>
</table>

Eine ausführliche Beschreibung zur Verteilung der Module folgt in Kapitel 11: Verteilungssicht.
6 Lösungsstrategien

Folgend sind zentrale Strategien und Grundlagen für die Architektur gelistet. Details sind in den nachfolgenden Kapiteln beschrieben und Entscheidungen sind in Kapitel 14 festgehalten.

6.1 Funktionale Lösungen

6.1.1 Aufteilung in «AccessCoordination» und «DataPersistence»-Modul

- Jedes Modul hat einen definierten Aufgabenbereich:
  AccessCoordination: öffentliche Schnittstellen, Zugangsverwaltung, Verwaltung von Benutzerdaten
  DataPersistence: Verwaltung von Statistik-Daten, Generierung von Vergleichsdaten
- Module können unabhängig entwickelt, erweitert, ausgetauscht, betrieben und verteilt werden.

EF01 Lose Kopplung zwischen «AccessCoordination» und «DataPersistence»-Modul

6.1.2 Statistik-Daten-Aktuator als Basis der Datenbank

- Statistik-Daten werden gemäss Businesslogik geordnet, d.h. Betriebskenngrössen sind bereits in der Datenbank dem Aktuator zugeordnet und müssen nicht jeweils wieder bei einer Abfrage nach Seriennummer geordnet werden.

EF02 Änderung des zentralen Datenmodells vom Statistik-Daten-Tupel zum Statistik-Daten-Aktuator

6.1.3 Statistik-Daten-Abfrage-Parameterset können gespeichert und verwaltet werden

In den Anforderungen ist der Ablauf für die Anwendungsfälle UC10, UC11 und UC12 wie folgt beschrieben:

<table>
<thead>
<tr>
<th>Anwendungsfall</th>
<th>Hauptszenario</th>
</tr>
</thead>
</table>
| UC10           | 1) Anfrage zur Bekanntgabe einer Statistik-Daten-Selektion durch den Akteur.  
                 2) System verwendet das gesendete Statistik-Daten-Selektions-Parameterset zur Erstellung der gewünschten Statistik-Daten-Selektion.  
                 3) System gibt die Statistik-Daten-Selektion dem Akteur bekannt. |
| UC11           | 1) Anfrage zur Bekanntgabe sämtlicher Statistik-Daten-Selektions-Parametersets durch den Akteur.  
                 2) System gibt sämtliche Statistik-Daten-Selektions-Parametersets dem Akteur bekannt. |
| UC12           | 1) Eingabe eines Statistik-Daten-Selektions-Parametersets durch den Akteur.  
                 2) System verarbeitet das Statistik-Daten-Selektions-Parameterset.  
                 3) System gibt triviale Empfangsbestätigung dem Akteur bekannt. |

Wie im Abschnitt 3.5 erläutert, wurden Statistik-Daten-Selektions-Parameterset, -Report-Parameterset und -Alarm-Parameterset zu Statistik-Daten-Abfrage-Parameterset vereinheitlicht.

Aufgrund der Aufteilung der Anwendung in «DataPersistence» und «AccessCoordination»-Modul werden die Anwendungsfälle UC10, UC11, UC12, UC20 und UC40 wie folgt gelöst:
«DataPersistence»-Modul
Das «DataPersistence»-Modul erfüllt die Anwendungsfälle UC10, UC20 und UC40 wie folgt:
- Es erzeugt aufgrund eines gesendeten Statistik-Daten-Abfrage-Parametersets eine/n passende/n Statistik-Daten-Selektion/Report/Alarm und gibt diese/n zurück.
Das «DataPersistence»-Modul erfüllt die Anwendungsfälle UC10, UC20 und UC40 unter folgenden Bedingungen:
- Es verwaltet keine Statistik-Daten-Abfrage-Parametersets.
- Es hat kein Gedächtnis bezüglich Statistik-Daten-Abfrage-Parametersets.
- Das Statistik-Daten-Abfrage-Parameterset muss bei jeder Anfrage gesendet werden.

«AccessCoordination»-Modul
Das «AccessCoordination»-Modul erfüllt die Anwendungsfälle UC11 und UC12, indem sämtliche Statistik-Daten-Abfrage-Parametersets durch das Modul verwaltet werden.
- Statistik-Daten-Abfrage-Parameterset entgegennehmen (Create)
- Statistik-Daten-Abfrage-Parameterset bekannt geben (Read)
- Änderungen an einem bestehenden Statistik-Daten-Abfrage-Parameterset entgegennehmen (Update)
- Statistik-Daten-Abfrage-Parameterset löschen (Delete)
Das «AccessCoordination»-Modul erfüllt die Anwendungsfälle UC10, UC20 und UC40 wie folgt:
- Es sendet auf Anfrage eines Akteurs ein selektiertes bereits gespeichertes Statistik-Daten-Abfrage-Parameterset an das «DataPersistence»-Modul, nimmt die/den Statistik-Daten-Selektion/Report/Alarm entgegen, und gibt diese/n als Antwort auf die Anfrage an den Akteur zurück.

Um die Statistik-Daten-Abfrage-Parametersets verwalten zu können, wurden die Struktur gegenüber der Anforderungsspezifikation um eine «Identifikation» und eine «Beschreibung» erweitert.

<table>
<thead>
<tr>
<th>StatisticDataQueryParameterset (Erweiterung)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Name</td>
</tr>
<tr>
<td>StatisticDataQueryParametersetId</td>
</tr>
</tbody>
</table>

Bemerkung: - wird als Identifikation (PK) in der Datenbank verwendet
- bei PUT Methoden zwingend
- muss bei POST Methoden auf «null» gesetzt werden.

<table>
<thead>
<tr>
<th>Description</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Text (string)</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: - textuelle Identifikation für Benutzer

6.1.4 Statistik-Daten-Selektionen für mehrere Aktuatoren

data (unbenanntes Objekt, dass die Liste kapselt)

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>List&lt;StatisticDataSelection&gt;</td>
<td>zusammengesetzt</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: Liste vom Typ «StatisticDataSelection»
6.2 Schnittstellen

6.2.1 http als Kommunikationsprotokoll
- Webinterface und Programmierschnittstelle können das selbe Protokoll verwenden.
- Gute Unterstützung für Entwickler (es existieren für alle gängigen Programmiersprachen Bibliotheken).

6.2.2 REST
- Implementation von Ressourcen strukturiert die Schnittstelle.
- Ressourcen können eindeutig angesprochen werden und sind zustandslos:
  z.B. /api/StatisticDataActuators/12345678 ist unveränderlich und liefert immer den Aktuator mit der S/N: 12345678
- Verwendung der http-Verben: GET, POST, PUT, DELETE, Auswertung des http-Status-Codes

6.3 Technologie

6.3.1 .Net, C#, ASP.NET Core 2.0 und Entity Framework Core
Diese Toolchain von Microsoft bietet für sämtliche Problemstellungen eine Lösung und ist sehr gut dokumentiert, zudem ist sie verfügbar.
Informationen: https://docs.microsoft.com/en-us/aspnet/core/

Eine Offline-Version der Dokumentation befindet sich im Projektverzeichnis: 99_Anhang

<table>
<thead>
<tr>
<th>ET01</th>
<th>Verwendung: ASP.NET Core 2.0 Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>ET02</td>
<td>Verwendung: Entity Framework Core als OR-Mapper</td>
</tr>
</tbody>
</table>

6.3.2 SQLite als Datenbanksystem
SQLite ist ein relationales Datenbanksystem. Die gesamte Datenbank befindet sich in einer einzigen Datei und wird zusammen mit der Anwendung verteilt.
Informationen: https://sqlite.org/index.html

| ET03 | SQLite-Datenbanksystem |

6.3.3 Verteilung als «Self-Contained Windows Service»
Für die Verteilung wird eine einfache Lösung benötigt, die möglichst wenig Abhängigkeiten zu Drittsystemen hat.

| ET04 | Verteilung als «Self-Contained Windows Service» |
7 Programmierschnittstelle des Gesamtsystems

Es wird nur die Programmierschnittstelle ausführlich dokumentiert. Die Schnittstelle des «BrowserInterface» ist Teil des Webinterface, und es ist nicht Sinn und Zweck diese Schnittstelle explizit anzusprechen. Die Funktionalitäten der Programmierschnittstelle und des Webinterface sind bis auf die Anwendungsfälle UC100 und UC101 deckungsgleich. Für die Anwendungsfälle UC100 und UC101 (Benutzeroberfläche mit Authentifizierung und Autorisierung) gibt es keine äquivalente Programmierschnittstelle.

Die aktuellste und akkuratest Beschreibung liefert das System selbst. Die Programmierschnittstelle wird mittels Swagger (https://swagger.io) jeweils bei der Kompilierung des Moduls aus dem Quellcode erzeugt. Man kann die Online-Dokumentation unter folgender URI abrufen: https://[authority]/swagger

7.1 REST-Ressourcen

Es gibt Ressourcen, die implementiert wurden, die aber in keinem Anwendungsfall explizit gefordert wurden. So ist es z.B. möglich ein Statistik-Daten-Abfrage-Parameterset zu löschen, oder sämtliche Statistik-Daten-Tupel abzurufen, die nicht synchronisiert wurden.

Nachfolgende sind sämtliche REST-Ressourcen und der dazugehörige Anwendungsfall gelistet.

7.1.1 StatisticDataActuators

<table>
<thead>
<tr>
<th>Methode</th>
<th>Ressource</th>
<th>Anwendungsfall</th>
<th>Akteur</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/StatisticDataActuators</td>
<td>UC50</td>
<td>A1c, A3b</td>
</tr>
<tr>
<td>GET</td>
<td>/api/StatisticDataActuators/{id}</td>
<td>UC50</td>
<td>A1c, A3b</td>
</tr>
</tbody>
</table>

7.1.2 StatisticDataInformation

<table>
<thead>
<tr>
<th>Methode</th>
<th>Ressource</th>
<th>Anwendungsfall</th>
<th>Akteur</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api</td>
<td>UC60</td>
<td>A3a, A3b, A3c</td>
</tr>
</tbody>
</table>

7.1.3 StatisticDataQueryParameterset

<table>
<thead>
<tr>
<th>Methode</th>
<th>Ressource</th>
<th>Anwendungsfall</th>
<th>Akteur</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/StatisticDataQueryParametersets</td>
<td>UC11</td>
<td>A1c, A3c</td>
</tr>
<tr>
<td>POST</td>
<td>/api/StatisticDataQueryParametersets</td>
<td>UC12</td>
<td>A1d</td>
</tr>
<tr>
<td>GET</td>
<td>/api/StatisticDataQueryParametersets/{id}</td>
<td>UC11</td>
<td>A1c, A3c</td>
</tr>
<tr>
<td>PUT</td>
<td>/api/StatisticDataQueryParametersets/{id}</td>
<td>UC12</td>
<td>A1d</td>
</tr>
<tr>
<td>DELETE</td>
<td>/api/StatisticDataQueryParametersets/{id}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GET</td>
<td>/api/StatisticDataQueryParametersets/Selection/{id}</td>
<td>UC10</td>
<td>A1c, A3c</td>
</tr>
</tbody>
</table>
7.1.4 **StatisticDataTuple**

<table>
<thead>
<tr>
<th>Methode</th>
<th>Ressource</th>
<th>Anwendungfall</th>
<th>Akteur</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/StatisticDataQueryParametersets/Report/{id}</td>
<td>UC20</td>
<td>A1c, A3c</td>
</tr>
<tr>
<td>GET</td>
<td>/api/StatisticDataQueryParametersets/Alarm/{id}</td>
<td>UC40</td>
<td>A1c, A3c</td>
</tr>
</tbody>
</table>

### 7.2 Modelle

Es gibt folgende zwei Modelle, die mit einer POST, oder PUT Methode übermittelt werden können:

- Statistik-Daten-Tupel: «StatisticDataTuple»
- Statistik-Daten-Abfrage-Parameterset: «StatisticDataQueryParameterset»

Auf eine Daten-Anfrage antwortet das System entweder mit einem Statistik-Daten-Aktuator oder aber einer Statistik-Daten-Selektion / -Report oder -Alarmierung:

- Statistik-Daten-Aktuator: «StatisticDataActuator»
- Statistik-Daten-Selektion: «StatisticDataSelection»
- Statistik-Daten-Report: «StatisticDataReport»
- Statistik-Daten-Alarmierung: «StatisticDataAlarm»

#### 7.2.1 Benutzte Modelle (für Anfrage) bei Anwendungsfällen

<table>
<thead>
<tr>
<th>Anwendungsfall</th>
<th>Modell</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC12: Statistik-Daten-Abfrage-Parameterset entgegennehmen</td>
<td>Statistik-Daten-Abfrage-Parameterset</td>
</tr>
<tr>
<td>UC30: Statistik-Daten-Tupel entgegennehmen</td>
<td>Statistik-Daten-Tupel</td>
</tr>
<tr>
<td>UC31: Modifiziertes Statistik-Daten-Tupel entgegennehmen</td>
<td>Statistik-Daten-Tupel</td>
</tr>
</tbody>
</table>
### 7.2.2 Benutzte Modelle (für Antworten) bei Anwendungsfällen

<table>
<thead>
<tr>
<th>Anwendungsfall</th>
<th>Modell</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Statistik-Daten-Aktuator</td>
</tr>
<tr>
<td>UC50: Statistik-Daten-Aktuator bekannt geben</td>
<td></td>
</tr>
<tr>
<td>UC10: Statistik-Daten-Selektion bekannt geben</td>
<td></td>
</tr>
<tr>
<td>UC20: Statistik-Daten-Report bekannt geben</td>
<td></td>
</tr>
<tr>
<td>UC40: Statistik-Daten-Alarmierung bekannt geben</td>
<td></td>
</tr>
</tbody>
</table>

### 7.2.3 Aufbau: Statistik-Daten-Tupel / «StatisticDataTuple»

<table>
<thead>
<tr>
<th>StatisticDataTuple</th>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pKey</td>
<td>UInt32</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Klasse mit primitiven Attributen vom Typ: Integer, String, Datum
Die Anzahl und der jeweilige Typ der Attribute kann dem Anhang 16.1 entnommen werden.

### 7.2.4 Aufbau: Statistik-Daten-Abrufe-Parameterset / «StatisticDataQueryParameterset»

<table>
<thead>
<tr>
<th>StatisticDataQueryParameterset</th>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StatisticDataQueryParametersetId</td>
<td>numerisch</td>
<td>Nein</td>
</tr>
</tbody>
</table>

Bemerkung: - wird als Identifikation (PK) in der Datenbank verwendet  
- bei PUT Methoden zwingend  
- muss bei POST Methoden auf «null» gesetzt werden.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Text (string)</td>
<td>Text (string)</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: - textuelle Identifikation für Benutzer

<table>
<thead>
<tr>
<th></th>
<th>SelectedStatisticDataTupleNumericAttributes</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>zusammengesetzt</td>
<td>zusammengesetzt</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: Liste vom Typ «SelectedStatisticDataTupleNumericAttribute» (siehe unten)

<table>
<thead>
<tr>
<th></th>
<th>ConditionSetOperator</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enumeration</td>
<td>Enumeration</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: - gibt an, ob die «Conditions» (Bedingungen) als Schnitt- (AND) oder Vereinigungsmenge (OR) behandelt werden
### Conditions

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditions</td>
<td>zusammengesetzt</td>
<td>Ja</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>Liste vom Typ «Condition» (siehe unten)</td>
<td></td>
</tr>
<tr>
<td>StartDate</td>
<td>Datum</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>Startdatum eines Auswahlbereichs</td>
<td></td>
</tr>
<tr>
<td>EndDate</td>
<td>Datum</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>Enddatum eines Auswahlbereichs</td>
<td></td>
</tr>
</tbody>
</table>

### SelectedStatisticDataTupleNumericAttribute

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatisticDataTupleNumericAttribute</td>
<td>Enumeration</td>
<td>Ja</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>eines von 23 möglichen numerischen Attributen eines Statistik-Daten-Tupels</td>
<td></td>
</tr>
<tr>
<td>IncreaseByPeriodType</td>
<td>Enumeration</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>gibt an, ob der folgende Wert das Wachstum pro Tag, Woche, Monat oder Jahr angibt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mögliche Werte sind: DAY, WEEK, MONTH, YEAR (vgl. «IncreaseByPeriod»)</td>
<td></td>
</tr>
<tr>
<td>IncreaseByPeriod</td>
<td>numerisch</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>Wachstum pro Tag, Woche, Monat oder Jahr (vgl. «IncreaseByPeriodType»)</td>
<td></td>
</tr>
<tr>
<td>InitialValue</td>
<td>numerisch</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>Startwert der Wachstumkurve</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Falls keine Angaben, dann 0</td>
<td></td>
</tr>
<tr>
<td>AlarmLimit</td>
<td>numerisch</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>konstante Limite für Alarmierung</td>
<td></td>
</tr>
<tr>
<td>AlarmCompareOperator</td>
<td>Enumeration</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>gibt an, ob der Alarm bei Über- oder Unterschreitung der Limite ausgelöst wird (vgl. AlarmLimit)</td>
<td></td>
</tr>
<tr>
<td>AlarmNotificationEmail</td>
<td>Text (string)</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>E-Mail-Adresse, die alarmiert wird bei Über- oder Unterschreitung der Limite</td>
<td></td>
</tr>
<tr>
<td>AlarmNotificationEnabled</td>
<td>Boolean</td>
<td>Nein</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>Alarmierung kann ein- oder ausgeschaltet sein</td>
<td></td>
</tr>
</tbody>
</table>

### Condition

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatisticDataTupleQueryAttribute</td>
<td>Enumeration</td>
<td>Ja</td>
</tr>
<tr>
<td>Bemerkung:</td>
<td>eines von 9 möglichen Abfrage-Attributen eines Statistik-Daten-Tupels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mittels dieser Attribute werden die Seriennummern bestimmt, die für die Daten-Auswahl in Frage kommen.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mehrere «Condition» können mittels «ConditionSetOperator» eingeschränkt oder vereinigt werden. (vgl. «ConditionSetOperator»)</td>
<td></td>
</tr>
</tbody>
</table>
7.2.5 Statistik-Daten-Aktuator: «StatisticDataActuator»


(nicht öffentlich)

7.2.6 Statistik-Daten-Selektion: «StatisticDataSelection»

data (unbenanntes Objekt, dass die Liste kapselt)

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>List&lt;StatisticDataSelection&gt;</td>
<td>zusammengesetzt</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: Liste vom Typ «StatisticDataSelection» (siehe unten)

StatisticDataSelection

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial</td>
<td>numerisch</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: Seriennummer des Aktuators

SingleAttributes

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>zusammengesetzt</td>
<td></td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: - Liste vom Typ «SingleAttribute» (siehe unten)

SingleAttribute

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>NumericAttribute</td>
<td>Enumeration</td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: - eines von 23 möglichen numerischen Attributen eines Statistik-Daten-Tupels

ResultTuples

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>zusammengesetzt</td>
<td></td>
<td>Ja</td>
</tr>
</tbody>
</table>

Bemerkung: - Liste vom Typ «ResultTuple» (siehe unten)
### 7.2.7 Statistik-Daten-Report: «StatisticDataReport»
Ein Statistik-Daten-Report ist analog zu einer Statistik-Daten-Selektion (vgl. 0), der Unterschied besteht im Typ «ResultTuple».

<table>
<thead>
<tr>
<th>ResultTuple</th>
<th>Typ</th>
<th>Notwendig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DatapointDate</td>
<td>Datum</td>
<td>Ja</td>
</tr>
<tr>
<td>Bemerkung: Zeitstempel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ValueDevice</td>
<td>numerisch</td>
<td>Ja</td>
</tr>
<tr>
<td>Bemerkung: Zahlenwert vom Typ «NumericAttribute» mit Zeitstempel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 7.2.8 Statistik-Daten-Alarmierung: «StatisticDataAlarm»
Eine Statistik-Daten-Alarmierung ist analog zu einer Statistik-Daten-Selektion (vgl. 0). Der Unterschied ist nicht das Format, sondern dass nur Werte zurückgeliefert werden, die über-/unter dem Limit sind.
8 Typische Strukturen und übergreifende Konzepte

8.1 REST

Ziel: Die Schnittstellen sollen REST-konform sein

Es wird an dieser Stelle nicht beschrieben was REST-konform bedeutet. Information dazu findet man unter anderem in folgenden Quellen:

REST und HTTP, 3. Auflage, Tilkov, Eigenbrodt, Schreier, Wolf [16]
https://de.wikipedia.org/wiki/Representational_State_Transfer
https://www.restapitutorial.com

Um das Ziel zu erreichen, wurde versucht die Schnittstellen so zu implementieren, dass sie folgende Stufen erfüllen.

Quelle: https://www.martinfowler.com/articles/richardsonMaturityModel.html

<table>
<thead>
<tr>
<th>Bezeichnung</th>
<th>Bemerkung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stufe 0</td>
<td>http als Protokoll • Ein wesentliches Merkmal von REST ist es, dass Technologien des World Wide Webs verwendet werden.</td>
</tr>
<tr>
<td>Stufe 1</td>
<td>Ressourcen • Informationen und Informationsgruppen haben eindeutige Adressen. • Anfragen eines Clients an den Server sollen sämtliche Information über den Zustand beinhalten. • Zugriff auf eine Ressource sollte immer das gleiche Ergebnis liefern: → Zustandslosigkeit</td>
</tr>
<tr>
<td>Stufe 2</td>
<td>http Methoden • http Methoden und http Status Codes sind standardisiert. • Die Wahl der Methode hat Einfluss auf die Behandlung der Anfrage, z.B. gilt für GET, dass keine Veränderung vorgenommen wird: → Zustandslosigkeit</td>
</tr>
<tr>
<td>Stufe 3</td>
<td>Hypermedia Controls HATEOAS (Hypertext As The Engine Of Application State) • Antworten werden mit Links erweitert, die aufzeigen, welche möglichen weiteren Anfragen die Anwendung im momentanen Zustand entgegennehmen kann.</td>
</tr>
</tbody>
</table>
8.2 **HATEOAS (Hypertext As The Engine Of Application State)**


Eine Maschine kann so, ähnlich wie ein Mensch im World Wide Web, durch eine Anwendung surfen, ohne dabei auf Vorwissen angewiesen zu sein.

**EF05** Programmierschnittstellen um HATEOAS Links erweitern

### 8.2.1 Vergleich der System-Schnittstellen

Nachfolgend sind die Zustände des Webinterfaces der «Statistik-Daten Auswertung» dargestellt. Öffnet man die Anwendung, startet man, wie bei den meisten anderen Webanwendungen, auf der Startseite. Der Benutzer bekommt dort die Möglichkeit zur gewünschten Funktion zu navigieren und weitere Zustände zu durchlaufen.

Abbildung 4: Zustandsmaschine des Webinterfaces

Dasselbe gilt auch für die Programmierschnittstelle der «Statistik-Daten Auswertung». Eine Übersicht der Möglichkeiten in Form von Links erhält man, wenn man eine GET Anfrage auf die Ressource «/api» macht.

Da es bei der «Statistik-Daten Auswertung» Anwendung ein Hauptanliegen ist, dass Menschen und Maschinen als Akteure auftreten können, ist die Zustandsmaschine für die Programmierschnittstelle der Zustandsmaschine des Webinterfaces sehr ähnlich.

In der momentanen Implementation wird aber bei der Programmierschnittstelle auf Authentifizierung und Autorisierung verzichtet, daher fehlen Pendants für «Account» und «Manage».
Abbildung 5: Zustandsmaschine der Programmierschnittstelle


Abbildung 6: Zustandsmaschine der Ressource «/api/StatisticDataTuple»
### 8.2.2 Implementierung

Mittels «HateoasLinkHelper»-Klasse können jegliche Arten von Objekten gekapselt und mit Links versehen werden, bevor sie an den Konsumenten versendet werden. Der Konsument entpackt die Nutzdaten und verwendet die Link-Information für seine nächste Aktion.

#### Abbildung 7: Klassendiagramm des «HateoasLinkHelpers»

### 8.2.3 Interne Schnittstelle


#### Abbildung 8: Zustandsmaschine der internen Programmierschnittstelle

8.3 Datenpersistenz in der Datenbank

Für die Anwendung gilt, dass Zustände in Datenbanken gespeichert werden (Database Session State [10]). Zum Beispiel gibt es in der Datenbank «AccessCoordinationDb» die Tabelle «TemporaryStatisticDataTuples», in welcher sämtliche Statistik-Daten-Tupel verwaltet werden, die importiert aber noch nicht akzeptiert wurden. Temporäre Daten könnten auch im Memory des Webservers (Server Session State [10]) oder aber als Cookies im Browser (Client Session State [10]) gespeichert werden. Der Vorteil von Datenbanken liegt aber darin, dass der Session State nicht an den Client gebunden ist. So können temporäre Daten in einer ersten Session importiert und bei einem nächsten Login bearbeitet werden.

8.4 Vorgaben von ASP.NET Core 2.0

Aufgrund der Wahl des Frameworks werden viele typische Strukturen und Konzepte durch das Framework vorgegeben. Diese Vorgaben sollten auch so angewendet werden. Nachfolgende Aufzählung weist auf Vorgaben hin, die bei der Architektur berücksichtigt wurden, hat aber keinen Anspruch auf Vollständigkeit. Es sei hier auf die Dokumentation von Microsoft verwiesen:
https://docs.microsoft.com/en-us/aspnet/core/

Eine Offline-Version der Dokumentation befindet sich im Projektverzeichnis: 99_Anhang

<table>
<thead>
<tr>
<th>Methode</th>
<th>http</th>
<th>Routing-Endpunkt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index()</td>
<td>GET</td>
<td>/DataTupleImport/</td>
</tr>
<tr>
<td>Details(uint? id)</td>
<td>GET</td>
<td>/DataTupleImport/Details/5</td>
</tr>
<tr>
<td>Create()</td>
<td>GET</td>
<td>/DataTupleImport/Create</td>
</tr>
<tr>
<td>Create(StatisticDataTuple statisticDataTuple)</td>
<td>POST</td>
<td>/DataTupleImport/Create</td>
</tr>
<tr>
<td>Edit(uint? id)</td>
<td>GET</td>
<td>/DataTupleImport/Edit/5</td>
</tr>
<tr>
<td>Edit(uint? id, StatisticDataTuple statisticDataTuple)</td>
<td>POST</td>
<td>/DataTupleImport/Edit/5</td>
</tr>
<tr>
<td>AcceptAll(string currentFilter)</td>
<td>GET</td>
<td>/DataTupleImport/AcceptAll</td>
</tr>
</tbody>
</table>
8.4.2 «Model View Controller»-Muster


https://docs.microsoft.com/en-us/aspnet/core/mvc/overview


Das Resultat der Anfrage («Request») wird vom «Controller» in Form eines «ViewModel» an die «View» übergeben, welche ihrerseits den Inhalt des «ViewModel» als Antwort («Response») dem Benutzer anzeigt.

Abbildung 9: MVC Muster in Web-Anwendungen


Ein «Entry Point» (oder auch Front Controller) wird bei ASP.NET als "Middleware" durch das Framework implementiert und bedient mehrere Controller. Verschiedene «Middleware» Stufen funktionieren wie eine Pipeline.
Eine Anfrage durchläuft mehrere Stufen, bevor sie an einen «Controller» übergeben wird. Es können auch benutzerspezifische «Middleware»-Elemente selber hinzugefügt werden.

Das Konzept ist ausführlich in der Dokumentation von Microsoft beschrieben:
https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/


Als Beispiel folgt die Klasse «DataTupleImportController» und ihre Methoden mit passender View:

<table>
<thead>
<tr>
<th>Methode</th>
<th>http</th>
<th>Routing-Endpunkt</th>
<th>View-Pfad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index()</td>
<td>GET</td>
<td>DataTupleImport/</td>
<td>\Views\DataTuple\Import\Index.cshtml</td>
</tr>
<tr>
<td>Details(uint? id)</td>
<td>GET</td>
<td>DataTupleImport/Details/5</td>
<td>\Views\DataTuple\Import\Details.cshtml</td>
</tr>
<tr>
<td>Create()</td>
<td>GET</td>
<td>DataTupleImport/Create</td>
<td>\Views\DataTuple\Import\Create.cshtml</td>
</tr>
<tr>
<td>Edit(uint? id)</td>
<td>GET</td>
<td>DataTupleImport/Edit/5</td>
<td>\Views\DataTuple\Import\Edit.cshtml</td>
</tr>
<tr>
<td>Accept(uint? id)</td>
<td>GET</td>
<td>DataTupleImport/Accept/5</td>
<td>\Views\DataTuple\Import\Accept.cshtml</td>
</tr>
<tr>
<td>Reject(uint? id)</td>
<td>GET</td>
<td>DataTupleImport/Reject/5</td>
<td>\Views\DataTuple\Import\Reject.cshtml</td>
</tr>
<tr>
<td>Import()</td>
<td>GET</td>
<td>DataTupleImport/Import</td>
<td>\Views\DataTuple\Import\Import.cshtml</td>
</tr>
</tbody>
</table>

8.4.3 «Model View Controller» Muster für Programmierschnittstellen


Im Fall einer Anfrage an eine API Ressource liefert der Controller ein «ApiModel» an die Middleware. Die Middleware wiederum erzeugt keine «View», sondern sendet die Daten in Form eines serialisierten Objects an den Benutzer.

Abbildung 10: Muster für eine Programmierschnittstelle
8.4.4 Dependency Injection


https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection


Auszug Registrierung eines «Service» in der Datei «Startup.cs»:

services.AddTransient<StatisticDataTupleService, StatisticDataTupleService>();


REST-Ressource: GET /internalApi/StatisticDataActuators

Abbildung 11: Aufruf der Methode «GetStatisticDataActuators()»
8.4.5 LINQ / Entity Framework Core


Die Spracherweiterung LINQ für den einfachen Datenzugriff ist ein Bestandteil von .NET und versucht eine einheitliche Methode für jeglichen Datenzugriff zur Verfügung zu stellen.

Für die Architektur hat es den Vorteil, dass sämtliche Datenbanken Typen verwendet werden können, für welche ein Provider existiert. Es müssen nicht zwingend SQL-Datenbanken sein. So ist z.B. die .NET In-Memory-Datenbank eigentlich nicht vollständig SQL kompatibel, wird dank ihrer Geschwindigkeit aber trotzdem für Unitests verwendet, da keine direkten SQL-Abfragen im Projekt verwendet werden.

Folgende Tabelle zeigt eine Übersicht der verwendeten Datenbanken und in welcher Umgebung sie verwendet werden. Die Wahl der verwendeten Datenbank muss dabei nicht zur Kompilationszeit erfolgen, sondern kann per Konfigurationsfile bestimmt werden.

### Eingesetzte Datenbanken

<table>
<thead>
<tr>
<th>Datenbank</th>
<th>Produktiv</th>
<th>Systemtests</th>
<th>Unitests</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessCoordinationDb</td>
<td>sqlite</td>
<td>inmemory-sqlite</td>
<td>inmemory</td>
</tr>
<tr>
<td>EloVarDb</td>
<td>mysql</td>
<td>inmemory</td>
<td>inmemory</td>
</tr>
<tr>
<td>DataPersistenceDb</td>
<td>sqlite</td>
<td>inmemory-sqlite</td>
<td>inmemory</td>
</tr>
</tbody>
</table>
### Typ der Datenbank

<table>
<thead>
<tr>
<th>config-switch</th>
<th>Typ</th>
<th>Bemerkung</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlite</td>
<td>SQLite Filesystem Database</td>
<td>DBMS mit minimalen externen Abhängigkeiten wird für Neuentwicklungen produktiv eingesetzt</td>
</tr>
<tr>
<td>mysql</td>
<td>Oracle MySQL Database</td>
<td>DBMS von Oracle Legacy-System beruht auf diesem DBMS</td>
</tr>
<tr>
<td>inmemory-sqlite</td>
<td>In-Memory SQLite Database</td>
<td>DBMS Simulation mit dem gleichen SQL-Umfang wie SQLite schneller zum Auf-/Absetzen als reales SQLite-DBMS</td>
</tr>
<tr>
<td>inmemory</td>
<td>.NET Core In-Memory Database</td>
<td>DBMS Simulation mit eingeschränktem SQL-Umfang sehr schnell zum Auf-/Absetzen</td>
</tr>
</tbody>
</table>

#### 8.4.6 Konfiguration in Dateien

Die Konfiguration der Module erfolgt mittels jeweiliger Konfigurations-Datei. Dabei handelt es sich um eine Textdatei im JSON-Format, welche im selben Verzeichnis wie die ausführbare Datei abgelegt sein muss. Dabei gelten folgende Namens-Konventionen:

<table>
<thead>
<tr>
<th>Umgebung</th>
<th>Konfigurationsdatei</th>
</tr>
</thead>
<tbody>
<tr>
<td>Produktiv</td>
<td>appsettings.json</td>
</tr>
<tr>
<td>Entwicklung</td>
<td>appsettings.Development.json</td>
</tr>
<tr>
<td>Integrationtests</td>
<td>appsettings.IntegrationTest.json</td>
</tr>
<tr>
<td>Systemtests</td>
<td>appsettings.SystemTest.json</td>
</tr>
</tbody>
</table>


#### 8.4.7 Log-Daten


https://github.com/serilog/serilog-aspnetcore

9 Bausteinsicht

9.1 Übersicht (Level 1)


Abbildung 13: Komponenten-Diagramm des Gesamtsystems / Level 1
9.2 «AccessCoordination» (Level 2)


Abbildung 14: Paket-Diagramm «AccessCoordination»
<table>
<thead>
<tr>
<th>Paket</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Eine einfache Objekte Struktur, die es erlaubt, die Konfigurationsdatei «appsettings.json» in die Anwendung zu binden.</td>
</tr>
<tr>
<td>Controllers/Api</td>
<td>Diese Gruppe von Kontroller implementieren die REST-Ressourcen der Systemschnittstelle.</td>
</tr>
<tr>
<td>Helpers</td>
<td>Helper-Klassen (z.B. Kapselung für HATEOAS-Links)</td>
</tr>
<tr>
<td>Services</td>
<td>Business-Logik des «AccessCoordination»-Moduls</td>
</tr>
<tr>
<td>Views</td>
<td>Die «Views» implementieren zusammen mit den Kontrollern das Webinterface.</td>
</tr>
</tbody>
</table>
9.2.1 Abhängigkeiten gegenüber dem ASP.NET Framework


Abbildung 15: Abhängigkeiten gegenüber «NuGet-Packages» des ASP.NET Frameworks
<table>
<thead>
<tr>
<th>NuGet-Package</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft.AspNetCore.All Version=&quot;2.0.9&quot;</td>
<td>Bibliotheken des ASP.NET Frameworks (Grundfunktionen)</td>
</tr>
<tr>
<td>Microsoft.EntityFrameworkCore.Tools Version=&quot;2.1.1&quot;</td>
<td>OR-Mapper (vgl. 8.4.5)</td>
</tr>
<tr>
<td>Pomelo.EntityFrameworkCore.MySql Version=&quot;2.1.1&quot;</td>
<td>MySQL-Connector für Entity Framework Core</td>
</tr>
<tr>
<td>Serilog.AspNetCore Version=&quot;2.1.1&quot;</td>
<td>Logging-Mechanismus</td>
</tr>
<tr>
<td>Serilog.Settings.Configuration Version=&quot;2.6.1&quot;</td>
<td>Logging-Mechanismus</td>
</tr>
<tr>
<td>Serilog.Sinks.File Version=&quot;4.0.0&quot;</td>
<td>Logging-Mechanismus-Erweiterung für Log-Dateien</td>
</tr>
<tr>
<td>Swashbuckle.AspNetCore Version=&quot;3.0.0&quot;</td>
<td>Swagger-Dokumentation</td>
</tr>
</tbody>
</table>
9.3 DataPersistence (Level 2)

 Folgendes Diagramm zeigt die Aufteilung des «DataPersistence»-Moduls in dessen Pakete und Schnittstellen. Die Darstellung soll einen Einblick geben, von welchen Paketen Anfragen bearbeitet werden, und wie die Abhängigkeiten zwischen den Paketen aufgebaut sind. Im Diagramm ist das Paket für die Implementierung der reinen Datenmodelle wiederum bewusst nicht dargestellt.

Abbildung 16: Paket-Diagramm “DataPersistence”

<table>
<thead>
<tr>
<th>Paket</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Eine einfache Objekte Struktur, die es erlaubt, die Konfigurationsdatei «appsettings.json» in die Anwendung zu binden.</td>
</tr>
<tr>
<td>Controllers</td>
<td>Diese Gruppe von Kontrollern implementieren die REST-Ressourcen der Systemschnittstelle.</td>
</tr>
<tr>
<td>Helpers</td>
<td>Helper-Klassen (z.B. Kapselung für HATEOAS-Links)</td>
</tr>
<tr>
<td>Services</td>
<td>Business-Logik des «DataPersistence»-Moduls</td>
</tr>
</tbody>
</table>
9.3.1 Abhängigkeiten gegenüber dem ASP.NET Framework


Abbildung 17: Abhängigkeit gegenüber Bibliotheken des ASP.NET Frameworks

<table>
<thead>
<tr>
<th>NuGet-Package</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft.AspNetCore.All Version=&quot;2.0.9&quot;</td>
<td>Bibliotheken des ASP-.NET Frameworks (Grundfunktionen)</td>
</tr>
<tr>
<td>Microsoft.EntityFrameworkCore.Tools Version=&quot;2.1.1&quot;</td>
<td>OR-Mapper (vgl. 8.4.5)</td>
</tr>
<tr>
<td>Serilog.AspNetCore Version=&quot;2.1.1&quot;</td>
<td>Logging-Mechanismus</td>
</tr>
<tr>
<td>Serilog.Settings.Configuration Version=&quot;2.6.1&quot;</td>
<td>Logging-Mechanismus</td>
</tr>
<tr>
<td>Serilog.Sinks.File Version=&quot;4.0.0&quot;</td>
<td>Logging-Mechanismus-Erweiterung für Log-Dateien</td>
</tr>
<tr>
<td>Swashbuckle.AspNetCore Version=&quot;3.0.0&quot;</td>
<td>Swagger-Dokumentation</td>
</tr>
</tbody>
</table>
9.4 AccessCoordination (Level 3)

Am Beispiel des «StatisticDataActuatorsController» soll nochmals gezeigt werden, wie mittels «Dependency Injection» (vgl. 8.4.4) die Abhängigkeiten zwischen einem Kontroller und der Business-Logik aufgebaut ist.

Der «StatisticDataActuatorsController» verwaltet die Ressource «/api/StatisticDataActuators». Diese Ressource hat folgende Beschreibung:

<table>
<thead>
<tr>
<th>Methode</th>
<th>Ressource</th>
<th>Anwendungsfall</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/api/StatisticDataActuators</td>
<td>UC50: Statistik-Daten-Aktuator bekannt geben</td>
</tr>
<tr>
<td>GET</td>
<td>/api/StatisticDataActuators/{id}</td>
<td>UC50: Statistik-Daten-Aktuator bekannt geben</td>
</tr>
</tbody>
</table>


Abbildung 18: Zusammenhang zwischen «Controller» und «Service»
9.5 DataPersistence (Level 3)

Die zentrale Komponente des «DataPersistence» Moduls ist der «StatisticDataReportController». Er verwaltet die Ressource «/internalApi/StatisticDataReport». Diese Ressource hat folgende Beschreibung:

<table>
<thead>
<tr>
<th>Methode</th>
<th>Ressource</th>
<th>Anwendungsfall</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/internalApi/StatisticDataReport</td>
<td>Gibt leeres Statistik-Daten-Report-Parameterset zurück.</td>
</tr>
<tr>
<td>POST</td>
<td>/internalApi/StatisticDataReport</td>
<td>UC20: Statistik-Daten-Report bekannt geben</td>
</tr>
</tbody>
</table>

Um einen Statistik-Daten-Report bekannt zu geben, müssen folgende Schritte durchgeführt werden:

   → Diese Aufgabe übernimmt die Klasse «ReportParametersetHelpers».
   (Die Klasse existiert, die Funktionalität ist aber noch nicht implementiert.)

2. Sämtliche Statistik-Daten-Aktuatoren finden, die den Bedingungen entsprechen.
   → Diese Aufgabe übernimmt die Klasse «StatisticDataFindActuatorService».

3. Für jeden gefundenen Statistik-Daten-Aktuator muss die passende Statistik-Daten-Selektion erstellt werden.
   → Diese Aufgabe übernimmt die Klasse «StatisticDataSelectionService».

   → Diese Aufgabe übernimmt die Klasse «StatisticDataReportService» mit Hilfe der Klasse «ValueCompareLinear», welcher das Interface «IValueCompare» realisiert.

Das Klassen-Modell zeigt, wie die verschiedenen Klassen miteinander verknüpft sind.

Abbildung 19: Zusammenhang zwischen «Controller» und «Service»
9.6 Datenmodelle «AccessCoordination»


9.6.2 «InformationApiModel»
Ein einfaches Objekt mit Schnittstellen-Informationen, dass bei einer Anfrage auf die Basis-Ressource «/api» zurückgegeben wird. (vgl. 3.4.3 «UC60: API-Information bekannt geben»)

9.6.3 Statistik-Daten-Aktuator / «StatisticDataActuatorApiModel»
Dieses Modell entspricht der Schnittstellenbeschreibung (vgl. 7.2.5).

9.6.4 Statistik-Daten-Abfrage-Parameterset / «StatisticDataQueryParameterset»
Die zentrale Datenstruktur des «AccessCoordination»-Moduls ist das «StatisticDataQueryParameterset». Es wird daher in verschiedenen Versionen benötigt.

<table>
<thead>
<tr>
<th>StatisticDataQueryParameterset</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>*ApiModel</td>
<td>Basis-Version</td>
</tr>
<tr>
<td>*DbModel</td>
<td>Für die Speicherung in einer Datenbank muss die Struktur auf Tabellen verteilt werden. (vgl. 11.4.1) Aus diesem Grund wurden «StatisticDataQueryParameterset», «Condition» und «SelectedStatisticDataQueryTupleNumericAttribute» um Referenzen erweitert.</td>
</tr>
<tr>
<td>*ViewModel</td>
<td>Bei der Implementierung gab es Probleme mit «nullable Types». Eine Lösung war das «SelectedStatisticDataQueryTupleNumericAttributeViewModel», welches ohne «nullable Type» auskommt.</td>
</tr>
</tbody>
</table>
Abbildung 20: Klassendiagramm «StatisticDataQueryParameterset»
9.6.5 Statistik-Daten-Selektion / -Report / -Alarmierung / «StatisticDataQuery»


Abbildung 21: Klassendiagramm «StatisticDataQuery»

9.6.6 Statistik-Daten-Tupel / «StatisticDataTuple»

Das Statistik-Daten-Tupel ist ein einfaches Objekt, welches sämtliche primitiven Werte eines Datensatzes enthält. Für die temporäre Speicherung in der Datenbank, gibt es auch von diesem Modell ein Pedant, das um einen Primärschlüssel erweitert wurde. Auf die Darstellung wird aus Platzgründen verzichtet. (vgl. 7.2.3 & 16.1)
9.7 Datenmodelle «DataPersistence»

9.7.1 «InformationApiModel»

Einfaches Objekt mit Schnittstellen-Informationen, dass bei einer Anfrage auf die Basis-Ressource «/internalApi» zurückgegeben wird. (vgl. 9.6.2)

9.7.2 Statistik-Daten-Aktuator / «StatisticDataActuator»

Die zentrale Datenstruktur des «DatenPersistence»-Moduls ist der «StatisticDataActuator». Die Grundlage für das Modell ist in der Schnittstellenbeschreibung (vgl. 7.2.5) dokumentiert.


Das folgende Klassen-Diagramm zeigt den Aktuator, wobei aufgrund der Übersicht die «Aggregation» zwischen den Klassen für die rechte Seite nur exemplarisch gezeichnet wurde. Die Erweiterung um Referenzen (mittels Vererbung) wurde ganz weggelassen.

(nicht öffentlich)

Abbildung 22: Klassendiagramm «StatisticDataActuator»
9.7.3 Statistik-Daten-Abfrage-Parameterset / «StatisticDataQueryParameterset»


Abbildung 23: Klassendiagramm «StatisticDataQueryParameterset»
9.7.4 Statistik-Daten-Selection / -Report / -Alarmierung / «StatisticDataQuery»

Das Klassen-Diagramm für die Statistik-Daten-Selection / -Report und -Alarmierung entspricht weitgehend dem Modell des «AccessCoordination»-Moduls (vgl. 0), wobei auf das «ViewModel» verzichtet werden kann.

Abbildung 24: Klassendiagramm «StatisticDataQuery»
9.7.5 Statistik-Daten-Tuple / «StatisticDataTuple»

Das Statistik-Daten-Tuple entspricht weitgehend dem Modell des «AccessCoordination»-Moduls, welches auch in der Schnittstellenbeschreibung und im Anhang erläutert ist (vgl. 7.2.3 und 16.1).

10 Laufzeitsicht

10.1 Services


10.1.1 Services «AccessCoordination»


<table>
<thead>
<tr>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services.StatisticDataQueryService(transient)</td>
<td>Der «QueryService» beinhaltet folgende Funktionalitäten: - Parametersets aus der «AccessCoordinationDb» lesen - Parametersets aus der «AccessCoordinationDb» hinzufügen - Parametersets aus der «AccessCoordinationDb» editieren - Parametersets aus der «AccessCoordinationDb» löschen</td>
</tr>
</tbody>
</table>

Zudem verwaltet er die Kommunikationsverbindung gegenüber folgenden Ressourcen des «DataPersistence» Moduls:

1. Der «AlarmingService» prüft periodisch alle Statistik-Daten-Query-Parametersets, ob bei diesen die Alarmierung aktiviert ist.
2. Falls dies zutrifft, startet der Service eine Anfrage beim «DataPersistence» Modul.
3. Ist die Antwort positiv, wird eine E-Mail an die Adresse, die im Parameterset hinterlegt ist, versendet.
10.1.2 Services «DataPersistence»


<table>
<thead>
<tr>
<th>Name</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>
1. Die Datenbank wird dabei aufgrund einer Query-Bedingung nach passenden Aktuatoren durchsucht.  
2. Sämtliche gefundenen Aktuatoren werden in einer temporären Liste vermerkt. (select)  
3. Besteht das Parameterset aus mehreren Bedingungen, wird dieser Vorgang wiederholt.  
4. Wird die Schnittmenge gesucht, wird die temporäre Liste jeweils mit den gefundenen Aktuatoren vergangener Suchvorgänge verglichen und geschnitten. (intersect)  
5. Wird die Vereinigung gesucht, werden sämtliche temporären Listen zusammengefügt. (addrange)  
6. Für die Bestimmung der endgültigen Menge werden mehrfach gefundene Aktuatoren zusammengefasst. (distinct) |
| Services.StatisticDataSelectionService | Der «SelectionService» erzeugt die gewünschte Statistik-Daten-Selektion für die gefundenen Aktuatoren. |


10.2 Hypermedia As The Engine Of Application State (HATEOAS)

Die folgenden Darstellungen ergänzen das Kapitel 8.2 und zeigen die vollständige Zustandsmaschine der REST-Programmierschnittstelle und deren Ressourcen.

10.2.1 HATEOAS «AccessCoordination»

Abbildung 25: «/api»
10.2.2 HATEOAS «DataPersistence»

Abbildung 29: «/internalApi»

Abbildung 30: «/internalApi/StatisticDataTuple»

Abbildung 31: «/internalApi/StatisticDataActuator»
Abbildung 32: «/internalApi/StatisticDataSelection»

Abbildung 33: «/internalApi/StatisticDataReport»

Abbildung 34: «/internalApi/StatisticDataSelection»
11 Verteilungssicht

11.1 Überblick


Abbildung 35: Gesamtüberblick über die Verteilung des Systems
### 11.2 Abhängigkeiten und Kommunikation


![Abbildung 36: «self-contained service»](image-url)
### Komponente | Beschreibung
--- | ---
DataPersistenceDb | Die SQLite-Datenbank-Datei im Dateisystem des Windows Server.
ASP.NET Core 2.0 | Bibliotheken, die es erlauben eine Webanwendung auf Basis von .NET Core 2.0 zu entwickeln. z.B. Middleware, Dependency Injection, u.s.w.
SQLite | Bibliothek, die die SQLite-Datenbank-Datei verwaltet.
Kestrel | Ein leichtgewichtiger Web Server, der auf .NET Core 2.0 läuft. [https://docs.microsoft.com/en-us/aspnet/core/fundamentals/servers/](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/servers/)
.NET Core 2.0 Runtime | Laufzeitumgebung, die den .NET Common Intermediate Language Code ausführt.

### 11.3 Konfiguration verteilter Anwendungen

Im Verzeichnis wo sich die ausführbare Datei (DataPersistence.exe oder AccessCoordination.exe) befindet, ist jeweils auch die Datei «appsettings.json» abgelegt. Diese Datei ist die zentrale Komponente, die es erlaubt Konfigurationseinstellungen nach der Kompilierung durchzuführen. So ist es zum Beispiel möglich, die TCP/IP-Port-Einstellungen zu verändern, den Speicherort und -name der SQLite-Datenbank-Datei anzupassen, Log-Level zu verändern oder den Namen des Systemadministrators anzupassen.

Mehr Details dazu sind in der Installationsdokumentation festgehalten oder aber direkt als Kommentar in der Datei.

Als Beispiel die etwas gekürzte «appsettings.json»-Datei des «DataPersistence»-Moduls ohne Logging:

```json
{
    // basic setup to run the DataPersistence module
    "DataPersistenceConfiguration": {
        "SystemName": "DataPersistence",
        "Version": "1.0.0.0",
        "ApiVersion": "v1",
        "ConnectionStrings": {
            // options are sqlite|inmemory
            "DataPersistenceDbConnectionType": "sqlite",
            "DataPersistenceDbConnection": "DataSource=DataPersistenceDb.db"
        }
    },
    // registered user with this address has all right and can promote other users
    "SystemAdmin": "admin@company.com",
    "ApplicationName": "Statistic Data Analysis, DP",
    "TermsOfService": "Release Candidate",
    "License": "For Internal Use Only",
    "Description": "Data Persistence Module",
    // port to access the system, must also be configured for AccessCoordination
    "Port": "5010",
    // log level options are Information|Warning|Error
    "Logging": {
        "..."
    }
}
```
11.4 Datenbanken

11.4.1 AccessCoordinationDb

Die «AccessCoordinationDb»-Datenbank speichert folgende Datenmodelle:

- StatisticDataQueryParametersetDbModel (vgl. 9.6.4)
- StatisticDataTupleDbModels (vgl. 9.6.6)

Abbildung 37: Datenbankschema der «AccessCoordinationDb»
11.4.2 DataPersistenceDb

Die «DataPersistenceDb»-Datenbank speichert folgende Datenmodelle

- StatisticDataActuator (vgl. 9.7.2)
- StatisticDataTupleLogDbModel (vgl. 0)


(nicht öffentlich)

Abbildung 38: Reduziertes Schema des «StatisticDataActuator»
12 Testinfrastruktur

Details wie man in ASP.NET Core 2.0 testet, findet man wiederum in der Dokumentation des Frameworks:

Unittest:  https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-dotnet-test
Integrationstest:  https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests


12.1 Unitest

Der Aufbau und Ablauf eines Unitests ist trivial. Die Test-Klasse erzeugt eine Instanz der zu testenden Klasse. Ruft die zu testende Methode auf und wertet deren Resultat aus.

![Abbildung 39: Aufbau eines Unitests](image.png)
12.2 Komponententest

Im Unterschied zu einem einfachen Unitest hat beim Komponententest das «Device Under Testing» (DUT) Abhängigkeiten, die entweder mittels Mock abstrahiert werden müssen, oder aber in die Testumgebung eingebunden werden.

Im Beispiel verfügt die Testklasse über eine «ServiceCollection» mittels derer die verschiedenen Services registriert werden und dann dem DUT «StaticDataActuatorController» bei der Initialisierung mittels Konstruktor Injektion zur Verfügung gestellt werden. Somit wird die «Dependency Injection» -Unit des Frameworks (vgl. 8.4.4) nicht benötigt, was den Test wesentlich vereinfacht.

Abbildung 40: Aufbau eines Komponententests
12.3 Integrationstest


Abbildung 41: Aufbau eines Integrationstests
12.4 Systemtest


Abbildung 42: Aufbau eines Systemtests


13 Toolchain

13.1 Enterprise Architect, Sparx Systems, v7.1.832

Graphische Dokumentation der Architektur

Bemerkung: Diese etwas ältere Version unterstützt nicht den vollen Sprachumfang von C#. Bevor der Quellcode importiert werden kann, sollten die kritischen Befehle «async» und «await» durch ein Batch-Skript auskommentiert werden. replace_async_await_for_EA.bat

Die Projektdatei befindet sich im Verzeichnis: 11_Architekturdokumentation

13.2 Swagger

Dokumentation der Schnittstelle

Bemerkung: Wie bereits in Kapitel 7 erwähnt, wird Swagger auch benützt, um eine eingebettete Dokumentation der Schnittstelle zu erzeugen.

Die statische Dokumentation befindet sich in den Verzeichnissen:

11_Architekturdokumentation\Swagger_Doc\AccessCoordination
11_Architekturdokumentation\Swagger_Doc\DataPersistance

13.3 SQLiteBrowser

Version: v3.10.1
Verwaltung von SQLite Datenbanken

13.4 .NET Runtime und SDK

Runtime: v2.0.9
SDK: v2.1.202

13.5 Visual Studio 2017, Microsoft

Version: v15.7.3
IDE für Entwicklung

13.5.1 Zusätzliche Plug-Ins zu Visual Studio

Xunit Testrunner: v2.4.0
Visual Assist: v10.9.2283.2
SQLite/SQL Server Compact Toolbox: v4.7.534
14 Entwurfsentscheidungen

14.1 Abgrenzende Entscheidungen

<table>
<thead>
<tr>
<th>14.1.1 EA01</th>
<th>Keine direkte Anbindung von Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datum</td>
<td>11.06.2018</td>
</tr>
<tr>
<td>Entscheidungsträger</td>
<td>Teamleiter Projektleitung</td>
</tr>
<tr>
<td>Gründe</td>
<td>• Anbindung von Hardware ist schwierig und daher sehr zeitaufwendig.</td>
</tr>
<tr>
<td>Konsequenzen</td>
<td>• Legacy-System wird vorerst als Schnittstelle zwischen Hardware (z.B. Aktuatoren) und der «Statistik-Daten Auswertung» benutzt.</td>
</tr>
<tr>
<td>Alternativen</td>
<td>• Erste Versuche, Hardware mittels «serial-port-json-server» [*] zu integrieren, waren zwar erfolgreich, die Lösung von Problemen mit dem Protokoll und der Schnittstelle wären aber zu zeitaufwendig gewesen.</td>
</tr>
<tr>
<td></td>
<td>[*] <a href="https://github.com/chilipeppr/serial-port-json-server">https://github.com/chilipeppr/serial-port-json-server</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14.1.2 EA02</th>
<th>Event-Log Daten werden nicht in das System übernommen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datum</td>
<td>11.06.2018</td>
</tr>
<tr>
<td>Entscheidungsträger</td>
<td>Teamleiter Projektleitung</td>
</tr>
<tr>
<td>Gründe</td>
<td>• Event-Log Daten sind temporäre Daten, die bei der Entwicklung und Fehlersuche helfen.</td>
</tr>
<tr>
<td>Konsequenzen</td>
<td>• Es gibt keinen Mehrwert, die Event-Log Daten in einer Datenbank zu sammeln.</td>
</tr>
<tr>
<td>Alternativen</td>
<td>• n/a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14.1.3 EA03</th>
<th>Alarmierungsfunktion mittels E-Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datum</td>
<td>03.07.2018</td>
</tr>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Gründe</td>
<td>• Funktionsweise und Ablauf der Alarmierung sind noch nicht abschliessend geklärt.</td>
</tr>
<tr>
<td>Konsequenzen</td>
<td>• Für einen Pilot-Betrieb werden Alarmierungen als E-Mails versendet.</td>
</tr>
<tr>
<td>Alternativen</td>
<td>• Verknüpfung zu Jira</td>
</tr>
</tbody>
</table>

[*] https://github.com/chilipeppr/serial-port-json-server
### 14.2 Funktionale Entscheidungen

<table>
<thead>
<tr>
<th>14.2.1 EF01</th>
<th>Lose Kopplung zwischen «AccessCoordination» und «DataPersistence»-Modul</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Datum</strong></td>
<td>10.05.2018</td>
</tr>
<tr>
<td><strong>Entscheidungsträger</strong></td>
<td>Projektleitung</td>
</tr>
<tr>
<td><strong>Gründe</strong></td>
<td>• Ziel ist es, ein lose gekoppeltes System zu entwickeln. Die Vorteile sind, dass die Module unabhängig entwickelt, erweitert, ausgetauscht, betrieben und verteilt werden können.</td>
</tr>
</tbody>
</table>
| **Konsequenzen** | • Es werden zwei Software-Projekte geführt.  
|                | • Die Datenmodelle müssen in beiden Modulen bekannt sein, das führt zur Quellcode-Verdoppelungen.  
|                | • Systemtests müssen verteilte Systeme handhaben können. |
| **Alternativen** | • Entwicklung eines monolithischen Moduls.  
|                | → Das Ziel der MAS-Arbeit ist aber die Entwicklung einer verteilten Architektur. |

<table>
<thead>
<tr>
<th>14.2.2 EF02</th>
<th>Änderung des zentralen Datenmodells vom Statistik-Daten-Tupel zum Statistik-Daten-Aktuator</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Datum</strong></td>
<td>15.05.2018</td>
</tr>
<tr>
<td><strong>Entscheidungsträger</strong></td>
<td>Projektleitung</td>
</tr>
</tbody>
</table>
| **Gründe**  | • Der Betreuer hat vorgeschlagen, das bereits bestehende Datenmodell des Legacy-Systems zu überdenken.  
|                | • Ein hierarchisches Datenmodell, dass den Aktuator als zentrales Element hat, modelliert die Realität besser.  
|                | • Es kann einfacher erweitert werden.  
|                | • Daten sind einfacher abfragbar, wenn sie bereits thematisch und nach Aktuator, in die Datenbank eingeordnet werden.  
|                | • (vgl. 6.1.2)                                                |
| **Konsequenzen** | • Umbau des «DataPersistence» Moduls notwendig.  
|                | • Suchalgorithmus muss angepasst werden.  
|                | • Einführung des Anwendungsfalls: UC50: Statistik-Daten-Aktuator bekannt geben  
|                | (vgl. 3.4.2)                                                |
| **Alternativen** | • Kein Umbau  
|                | → «Design smell»
### 14.2.3 EF03 Benutzerverwaltung mit Authentifizierung und Autorisierung

<table>
<thead>
<tr>
<th>Datum</th>
<th>11.07.2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
</tbody>
</table>

**Gründe**
- Für die Implementierung der Rollen musste eine Benutzerverwaltung mit Authentifizierung und Autorisierung eingeführt werden.

**Konsequenzen**
- Es gibt eine weitere Rolle: Der «Guest» ist ein Benutzer, der sich registriert hat, aber noch nicht auf die E-Mail zu Authentifizierung reagiert hat.
- Das Software-Projekt muss mit einem neuen Template nochmals aufgesetzt werden.
- Der E-Mail Versand für die Authentifizierung musste implementiert sein.

**Alternativen**
- Benutzerverwaltung ohne Authentifizierung und Autorisierung wäre hinfällig, da keine Berechtigungen verteilt werden könnten.

### 14.2.4 EF04 Programmierschnittstellen implementieren keine Authentifizierung und Autorisierung

<table>
<thead>
<tr>
<th>Datum</th>
<th>21.07.2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
</tbody>
</table>

**Gründe**
- Aus Zeitgründen wird die Authentifizierung und Autorisierung nicht auf die Programmierschnittstellen erweitert.

**Konsequenzen**
- System API: Den Umserver kann keine Rolle zugewiesen werden. Sie müssen sich an die ihnen zugedachte Rolle halten.
- interne API: Die Kommunikation zwischen «AccessCoordination» und «DataPersistence» Modul ist nicht verschlüsselt.

**Alternativen**
- Zeitplan lässt keine Alternativen zu.

### 14.2.5 EF05 Programmierschnittstellen um HATEOAS Links erweitern

<table>
<thead>
<tr>
<th>Datum</th>
<th>29.07.2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
</tbody>
</table>

**Gründe**
- Ziel ist eine möglichst REST-konforme Schnittstelle.
  - (vgl. 8.2)

**Konsequenzen**
- System API:
  - Erweiterung der Schnittstelle um Links
  - Einführung des Anwendungsfalls: UC60: API-Informationen bekannt geben (vgl. 3.4.3)
- interne API:
  - Erweiterung der Schnittstelle um Links (analog System API)
  - Das «AccessCoordination»-Modul bindet sich beim Startvorgang dynamisch an das «DataPersistence» Modul.

**Alternativen**
- ein tieferer REST-Level (vgl. 8.1)
### 14.3 Technologische Entscheidungen

<table>
<thead>
<tr>
<th>14.3.1 ET01</th>
<th>Verwendung: ASP.NET Core 2.0 Framework</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datum</td>
<td>05.03.2018</td>
</tr>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Gründe</td>
<td>• Es bestehen bereits minimale Erfahrung mit dem Umgang des Web Frameworks.</td>
</tr>
<tr>
<td></td>
<td>• C#-Knowhow innerhalb des Teams vorhanden.</td>
</tr>
<tr>
<td></td>
<td>• Verfügbarkeit der Toolchain (13.4) und IDE (13.5) ist gegeben.</td>
</tr>
<tr>
<td></td>
<td>• sehr gute Dokumentation</td>
</tr>
<tr>
<td></td>
<td>• ASP.NET Core 2.1 ist erst seit Ende Mai 2018 verfügbar.</td>
</tr>
<tr>
<td>Konsequenzen</td>
<td>• Anwendung läuft nur auf Systemen, für die eine .NET Runtime besteht.</td>
</tr>
<tr>
<td>Alternativen</td>
<td>• Python als Programmiersprache</td>
</tr>
<tr>
<td></td>
<td>→ keine Erfahrung mit grossen Projekten nur Skripts</td>
</tr>
<tr>
<td></td>
<td>• Django oder Flask als Web Framework</td>
</tr>
<tr>
<td></td>
<td>→ keine Erfahrung</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>14.3.2 ET02</th>
<th>Verwendung: Entity Framework Core als OR-Mapper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datum</td>
<td>05.03.2018</td>
</tr>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
<tr>
<td>Gründe</td>
<td>• Wird von Microsoft für neue ASP.NET Core Projekte empfohlen.</td>
</tr>
<tr>
<td></td>
<td>• Unterstützt LINQ</td>
</tr>
<tr>
<td></td>
<td>• Benötigt keine SQL-Experten-Kenntnisse.</td>
</tr>
<tr>
<td></td>
<td>• Unterstützt die Code-First Herangehensweise.</td>
</tr>
<tr>
<td></td>
<td>• Unterstützt das SQLite-Datenbanksystem.</td>
</tr>
<tr>
<td></td>
<td>• Unterstützt das MySQL-Datenbanksystem.</td>
</tr>
<tr>
<td></td>
<td>• Abstrahiert Datenbank → Datenbank kann einfach ausgetauscht werden. Z.B. Unitests</td>
</tr>
<tr>
<td>Konsequenzen</td>
<td>• Anwendung läuft nur auf Systemen, für die eine .NET Runtime besteht.</td>
</tr>
<tr>
<td>Alternativen</td>
<td>• Entity Framework 6</td>
</tr>
<tr>
<td></td>
<td>→ Mix von Standard .NET und .NET Core ist nicht gewünscht.</td>
</tr>
<tr>
<td></td>
<td>• Third-party OR-Mapper</td>
</tr>
<tr>
<td></td>
<td>→ keine Erfahrung</td>
</tr>
</tbody>
</table>
## 14.3.3 ET03  SQLite-Datenbanksystem

<table>
<thead>
<tr>
<th>Datum</th>
<th>10.05.2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
</tbody>
</table>
| Gründe      | • Hat minimale externe Abhängigkeiten.  
              • Wird von Entity Framework Core unterstützt.  
              • Kann in die Anwendung integriert werden → keine Abhängigkeiten.  
              • Wird in der Dokumentation von Microsoft neben den Microsoft-Produkten auch unterstützt. |
| Konsequenzen | • SQLite unterstützt nicht den vollen SQL-Sprachumfang.  
               • Ist nicht so performant wie andere Datenbanksysteme (z.B. MySQL)  
               • Anwendung und Datenbank können nicht verteilt werden. |
| Alternativen | • MySQL Datenbanksystem des Legacy-Systems  
                → Keine Testumgebung verfügbar, Schreibzugriff auf Produktivsystem wäre sehr riskant.  
                • Microsoft SQL Server oder anderes SQL-Datenbanksystem  
                → keine Erfahrung  
                • NoSQL-Datenbanksystem  
                → keine Erfahrung |

## 14.3.4 ET04  Verteilung als «Self-Contained Windows Service»

<table>
<thead>
<tr>
<th>Datum</th>
<th>30.07.2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entscheidungsträger</td>
<td>Projektleitung</td>
</tr>
</tbody>
</table>
| Gründe      | • Sämtliche Abhängigkeiten sind in einem Paket gebündelt.  
              • «Self-Contained» Anwendungen sind portabel. |
| Konsequenzen | • Installationspakete werden sehr umfangreich.  
               • Keine Integration in die Managementmöglichkeiten von Microsoft IIS. |
| Alternativen | • verwaltet durch Microsoft IIS  
               • verwaltet durch Apache HTTP Server |
15 Risiken und technische Schulden

<table>
<thead>
<tr>
<th>Id</th>
<th>Risiko</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>15.1 Funktionale Risiken und technische Schulden</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS01</td>
<td>Erweiterung der Anforderungsspezifikation</td>
<td>Die Anwendung hat im Moment einen guten Stand und kann von den Stakeholdern benutzt werden. Die Pilotphase muss aber unbedingt ausgewertet werden, bevor die momentan bekannten technischen Schulden abgearbeitet werden. Es ist wahrscheinlich, dass der Pilotbetrieb neue Anforderungen ergeben wird, wodurch sich die Prioritäten ändern werden.</td>
</tr>
<tr>
<td>TS03</td>
<td>Import von Daten</td>
<td>Die Funktionalität zum Import von Daten via Dateisystem wurde nur zu Test- und Demozwecken implementiert. Es hat sich aber gezeigt, dass eine solche Funktion durchaus wünschenswert wäre, falls die Daten aus einem Zip-Container mit Binärdaten extrahiert werden könnten. Der Aufwand dafür darf aber nicht unterschätzt werden, zudem ist zu klären, ob das Extrahieren der Daten auf Client- oder Server-Seite implementiert werden kann.</td>
</tr>
<tr>
<td>TS04</td>
<td>Hardware-Anbindung</td>
<td>Ursprünglich wurde eine direkte Anbindung von Hardware angedacht. Es wurde aber keine Möglichkeit gefunden, wie dies im Rahmen der Arbeit zu bewerkstelligen gewesen wäre. Diese Anforderung muss noch einmal überdacht und konkretisiert werden. (vgl. EA01)</td>
</tr>
<tr>
<td><strong>15.2 Nicht-funktionale Risiken und technische Schulden</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS05</td>
<td>Code-Verdoppelung</td>
<td>Die beiden Module wurden strikt getrennt entwickelt, dadurch wurden Quellcode-Teile kopiert. Vor allem bei den Schnittstellen und deren Datenmodellen gibt es Überschneidungen. Es sollte ein Weg gefunden werden, wie dieser Quellcode vereinheitlicht und zusammengeführt werden kann, z.B. mit einem GIT submodule auf Stufe Repo, oder die gemeinsame Quellcode-Basis wird in einem dritten GIT-Projekt verwaltet.</td>
</tr>
<tr>
<td>TS06</td>
<td>Benennung von Variablennamen</td>
<td>Das gewählte Schema führt zu sehr langen Variablennamen. Das Schema sollte daher nochmals kritisch beurteilt werden.</td>
</tr>
<tr>
<td>TS08</td>
<td>Testframework</td>
<td>Unittest haben zu viel repetitiven Quellcode. Das Framework sollte dahingehend erweitert werden, das mehrfach verwendete Funktionen zentral verwaltet werden.</td>
</tr>
<tr>
<td>TS09</td>
<td>Testframework</td>
<td>Das Testframework muss um eine Möglichkeit erweitert werden, dass Testdaten zentral erzeugt werden können. Momentan müssen sie oft «on-the-fly» erzeugt werden. Tests sollten parametrisierbar sein, so dass systematisch erzeugte Testdaten verwendet werden können.</td>
</tr>
<tr>
<td>Id</td>
<td>Risiko</td>
<td>Beschreibung</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TS10</td>
<td>UI-Darstellung</td>
<td>Die Darstellung entspricht dem Standard-Beispiel Template von Microsoft. Das UI sollte übersichtlicher gestaltet werden, besonders im Hinblick darauf, dass sehr viele Aktuatoren verwaltet werden müssen.</td>
</tr>
<tr>
<td>TS11</td>
<td>Eingabe-Validierung</td>
<td>Bis jetzt werden Eingaben nur auf Typ, Länge und Bedingung geprüft. Es könnten aber auch noch die Abhängigkeiten zwischen den Eingaben geprüft werden. Zum Beispiel, wird in einer Bedingung eine Zahl geprüft, dann soll die Eingabe auch auf numerische Eingabe prüfen auch wenn das Feld generisch ist und daher Text zu lässt.</td>
</tr>
<tr>
<td>TS12</td>
<td>Migration von Datenbanken</td>
<td>Sollte das Datenbankschema oft Änderungen unterliegen muss ein Weg gefunden werden, wie man das Problem umgehen kann, dass SQLite nur einen limitierten ALTER TABLE Befehl unterstützt. <a href="https://www.sqlite.org/lang_altertable.html">https://www.sqlite.org/lang_altertable.html</a></td>
</tr>
<tr>
<td>TS14</td>
<td>ASP.NET Core 2.0 End of Life</td>
<td>ASP.NET Core 2.1 wurde im Mai 2018 freigegeben, daher wurde es nicht für die Arbeit in Betracht gezogen. Leider ist der Vorgänger ASP.NET Core 2.0 bereits abgekündigt. Eine Migration sollte daher angestrebt werden.</td>
</tr>
<tr>
<td>TS15</td>
<td>Authentifizierung und Autorisierung wurden nicht auf Sicherheit geprüft</td>
<td>Authentifizierung und Autorisierung wurden gemäss den Vorgaben von Microsoft übernommen, dabei wurde aber kein Sicherheitsreview gemacht, um sicher zu gehen, dass bei der Implementation keine Fehler gemacht wurden.</td>
</tr>
<tr>
<td>TS16</td>
<td>Authentifizierung und Autorisierung Programmierschnittstellen aktivieren</td>
<td>Aus Zeitgründen wurde die Authentifizierung und Autorisierung nicht für die Programmierschnittstellen aktiviert. Auch die Kommunikation zwischen den Modulen ist nicht verschlüsselt. Diese sollte aber auch geschützt sein.</td>
</tr>
<tr>
<td>TS17</td>
<td>Logging/Serilog</td>
<td>Eine Funktionalität zum Loggen wurde integriert. Die Logging-Punkt sind aber zu wenig strukturiert gesetzt, so ist eine saubere Analyse des Laufzeitverhaltens nicht immer möglich.</td>
</tr>
<tr>
<td>TS18</td>
<td>Language localisation</td>
<td>Die Anwendung ist im Moment nur in englischer Sprache verfügbar. Die Anwendung soll dahingehend erweitert werden, dass die Sprache durch den Benutzer gewählt werden kann.</td>
</tr>
</tbody>
</table>

### 15.3 Infrastruktur Risiken und technische Schulden

<table>
<thead>
<tr>
<th>Id</th>
<th>Risiko</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS20</td>
<td>Versionsverwaltung</td>
<td>Im Rahmen der MAS-Arbeit wurde GIT als Versionsverwaltung benutzt, da eine verteilte Versionsverwaltung wesentliche Vorteile gegenüber einer zentralen hat, wenn der Zugriff zur Versionsverwaltung nicht immer gewährleistet ist. Schussendlich muss die Anwendung aber in das Subversion-System der Abteilung integriert werden.</td>
</tr>
<tr>
<td>TS21</td>
<td>Release Version</td>
<td>Es ist noch nicht geklärt, welche Informationen auf der Frontseite dargestellt werden sollen.</td>
</tr>
</tbody>
</table>
16 Anhang

16.1 Statistik-Daten-Tupel

(nicht öffentlich)
Statistik-Daten Auswertung

Masterarbeit HSR-MAS-SE 2016

Testkonzept
Inhaltsverzeichnis

Zu diesem Dokument ........................................................................................................... 4
  1.1 Zweck ........................................................................................................................ 4
  1.2 Gültigkeit ................................................................................................................... 4
  1.3 Verfügbarkeit des Dokumentes .................................................................................. 4

2 Weiterführende Informationen ....................................................................................... 5
  2.1 Definitionen, Akronymen und Abkürzungen .............................................................. 5
  2.2 Ergänzende Dokumente ............................................................................................ 5
  2.3 Prozessbezogene Dokumente ................................................................................... 5
  2.4 Versionshistorie ......................................................................................................... 5
  2.5 Verteilung ................................................................................................................ 5

3 Test-/Prüfobjekte .............................................................................................................. 6
  3.1 Statische Objekte ....................................................................................................... 6
  3.2 Dynamische Objekte .................................................................................................. 7
  3.2.1 Kontext des «System Under Test» ..................................................................... 7
  3.2.2 Schnittstellen ....................................................................................................... 8
  3.2.3 Gesamtsystem ...................................................................................................... 9
  3.2.4 Komponenten ..................................................................................................... 10

4 Zu testende Leistungsmerkmale ..................................................................................... 11
  4.1 Funktionale Anforderungen ..................................................................................... 11
  4.2 Nicht-funktionale Systemanforderungen ................................................................. 11

5 Leistungsmerkmale, die nicht getestet werden .............................................................. 11
  5.1 Funktionale Anforderungen ..................................................................................... 11
  5.2 Nicht-funktionale Systemanforderungen ................................................................. 11

6 Testarten .......................................................................................................................... 12
  6.1 Durchgeführte Tests ............................................................................................... 12
  6.2 Nicht durchgeführte Tests ...................................................................................... 12

7 Risiken .................................................................................................................................. 12

8 Testziele ............................................................................................................................. 12

9 Teststrategie ..................................................................................................................... 13
  9.1 Statische Testobjekte ............................................................................................. 13
  9.2 Dynamische Testobjekte ....................................................................................... 14
  9.2.1 «DataPersistence»-Komponente ...................................................................... 14
9.2.2 «AccessCoordination»-Komponente ................................................................. 15
9.4 Teststufen ............................................................................................................. 18
9.4.1 Unit- / Komponententests .................................................................................. 19
9.4.2 Integrationstests .................................................................................................. 20
9.4.3 Automatisierte Systemtests ............................................................................... 21
9.4.4 Manuelle Systemtests ....................................................................................... 22
9.4.5 Abnahmetest ...................................................................................................... 23
9.5 Übersicht Testobjekte und Testtyp ...................................................................... 24
9.5.1 Statische Objekte (3.1) ...................................................................................... 24
9.5.2 DataPersistence (3.2.4) .................................................................................... 24
9.5.3 AccessCoordination (3.2.4) ............................................................................. 25

10 Zeit-/Arbeitsplan ...................................................................................................... 25

Abbildungsverzeichnis

Abbildung 1: Kontextdiagramm .................................................................................... 7
Abbildung 2: Gesamtsystem .......................................................................................... 9
Abbildung 3: Package Diagramm «DataPersistence» (ohne «Models» Package) .............. 15
Abbildung 4: Package Diagramm «AccessController» (ohne «Models» Package) ............ 16
Abbildung 5: Allgemeines V-Modell mit Teststufen. Source: [8] ..................................... 18
Zu diesem Dokument

1.1 Zweck

Im Testkonzept werden Entscheidungen, Vorgehensweise, Planung und Abgrenzung der geplanten Testaktivitäten bezüglich der «Statistik-Daten Auswertung» Software definiert und festgehalten. Das Testkonzept gibt die Richtung und das Ziel der Testaktivitäten vor.


1.2 Gültigkeit

Dieses Dokument ist über die ganze Projektdauer gültig.

1.3 Verfügbarkeit des Dokumentes

Dieses Dokument befindet sich im Projektverzeichnis: 12_Testaktivitaeten
2 Weiterführende Informationen

2.1 Definitionen, Akronyme und Abkürzungen
Sämtliche Definitionen, Akronyme und Abkürzungen werden zentral in einem Projektglossar dokumentiert. Das Projektglossar wird kontinuierlich angepasst. Das Projektglossar befindet sich im Projektverzeichnis: 09_Glossar

2.2 Ergänzende Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

2.3 Prozessbezogene Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

2.4 Versionshistorie
(nicht öffentlich)

2.5 Verteilung

<table>
<thead>
<tr>
<th>Ver.</th>
<th>Name</th>
<th>Rolle / Titel / Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>003</td>
<td>Abgabe MAS-Arbeit</td>
<td>Abgabe MAS-Arbeit</td>
</tr>
</tbody>
</table>
# 3 Test-/Prüfobjekte

## 3.1 Statische Objekte

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Objekt</th>
<th>Beschreibung</th>
<th>Prüfung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Projektantrag</td>
<td>Der Projektantrag beschreibt das generelle Konzept, das Einsatzgebiet und die Motivation für die Entwicklung des Produkts.</td>
<td>Nein</td>
</tr>
<tr>
<td>2</td>
<td>Projektleitdokumentation</td>
<td>Der Zweck der Projektleitdokumentation ist das Projekt zu definieren, um eine Basis für das Projektmanagement zu bilden und eine Bewertung des Projekterfolges zu ermöglichen.</td>
<td>Ja</td>
</tr>
<tr>
<td>3</td>
<td>Anforderungsspezifikation</td>
<td>Die Anforderungsspezifikation beschreibt die Abgrenzung zwischen „Statistik-Daten Auswertung“ Produkt und Umsystemen, Anwendungsfälle, die logischen Datenstrukturen der Systemschnittstellen, die nichtfunktionale Anforderungen und den Kern der für die Funktionserfüllung notwendigen Informationen und Abläufe.</td>
<td>Ja</td>
</tr>
<tr>
<td>4</td>
<td>Architekturdokumentation</td>
<td>Die Architekturdokumentation beschreibt Sachverhalte, Zusammenhänge und Voraussetzungen, die für das Verständnis des Systems oder einzelner Entwurfsentscheidungen relevant sind.</td>
<td>Ja</td>
</tr>
<tr>
<td>5</td>
<td>Testkonzept</td>
<td>Im Testkonzept werden Entscheidungen, Vorgehensweise, Planung und Abgrenzung der geplanten Testaktivitäten bezüglich des Produkts definiert und festgehalten.</td>
<td>Ja</td>
</tr>
<tr>
<td>6</td>
<td>Systemtest-Testspezifikation</td>
<td>Die Systemtest-Testspezifikation dokumentiert die Testfälle, mit denen das System getestet werden kann. Zudem dokumentiert es die Herleitung der benützten Äquivalenzklassen.</td>
<td>Ja</td>
</tr>
<tr>
<td>7</td>
<td>Testbericht</td>
<td>Zusammenfassung und Bestandesaufnahme der Testresultate am Ende der MAS-Arbeit.</td>
<td>Ja</td>
</tr>
<tr>
<td>8</td>
<td>Installations- und Betriebsdokumentation</td>
<td>Die Installations- und Betriebsdokumentation beschreibt die Voraussetzungen für eine erfolgreiche Installation und Betrieb des Produkts und deren Durchführung.</td>
<td>Ja</td>
</tr>
<tr>
<td>9</td>
<td>Benutzerdokumentation</td>
<td>Die Benutzerdokumentation ist eine Sammlung von Informationen für Benutzer zum sicheren und bestimmungsgemässen Umgang mit dem Produkt.</td>
<td>Ja</td>
</tr>
</tbody>
</table>
3.2 Dynamische Objekte

Es gibt nur ein dynamisches Objekt, das getestet wird: «Statistik-Daten Auswertung» Software

3.2.1 Kontext des «System Under Test»

Abbildung 1: Kontextdiagramm
### 3.2.2 Schnittstellen

Ausführliche Beschreibungen zu den Systemschnittstellen sind in der Architekturdokumentation zu finden. Folgende Tabelle gibt einen kurzen Überblick zum Verständnis:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Beschreibung</th>
<th>Prüfung</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>BrowserInterface</td>
<td>Typ: Provided Interface, Akteure greifen darauf zu Browser Schnittstelle (Webinterface)</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementierung: Browser Schnittstelle (Webinterface)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protokoll: http</td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>/api/StatisticDataInformation</td>
<td>Typ: Provided Interface, externe Systeme greifen darauf zu REST-API Ressource</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementierung: REST-API Ressource</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protokoll: http</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hauptzweck: Über dieses Interface können externe Systeme Information zu weiteren REST-API Ressourcen abfragen.</td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td>/api/StatisticDataTuple</td>
<td>Typ: Provided Interface, externe Systeme greifen darauf zu REST-API Ressource</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementierung: REST-API Ressource</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protokoll: http</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hauptzweck: Über dieses Interface können externe Systeme neue Daten in das System einspeisen.</td>
<td></td>
</tr>
<tr>
<td>P4</td>
<td>/api/StatisticDataActuators</td>
<td>Typ: Provided Interface, externe Systeme greifen darauf zu REST-API Ressource</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementierung: REST-API Ressource</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protokoll: http</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hauptzweck: Über dieses Interface können externe Systeme die Aktuatorenkenngrössen auslesen.</td>
<td></td>
</tr>
<tr>
<td>P5</td>
<td>/api/StatisticDataQueryParameterset</td>
<td>Typ: Provided Interface, externe Systeme greifen darauf zu REST-API Ressource</td>
<td>Ja</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementierung: REST-API Ressource</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protokoll: http</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hauptzweck: Über diese Schnittstelle können externe Systeme einerseits Query-Parametersets verwalten und andererseits mittels dieser Query-Parametersets Statistik-Daten in Form von Selektionen, Reports oder Alarmierung abfragen.</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>Legacy</td>
<td>Typ: Required Interface, für Daten-Import benötigt Datenbank-Zugriff</td>
<td>Nein</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementierung: Datenbank-Zugriff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protokoll: TCP/IP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hauptzweck: Über diese Schnittstelle greift das System auf das bestehende Legacy-Data-Repository zu.</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>FileSystem</td>
<td>Typ: Required Interface, für Entwicklung und Test benötigt .NET-Framework</td>
<td>Nein</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Implementierung: .NET-Framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protokoll: filesystem</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hauptzweck: Über diese Schnittstelle werden Daten aus Dateien gelesen.</td>
<td></td>
</tr>
</tbody>
</table>
3.2.3 Gesamtsystem

Abbildung 2: Gesamtsystem
### 3.2.4 Komponenten

Das System besteht im Wesentlichen aus zwei Blöcken, welche unabhängig voneinander getestet werden können.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Beschreibung</th>
<th>Prüfung</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>DataPersistence</td>
<td>Zentrale Komponente, welche die Statistik-Daten verwaltet, Reports generiert und Alarme auslöst.</td>
<td>Ja</td>
</tr>
</tbody>
</table>
4 Zu testende Leistungsmerkmale

4.1 Funktionale Anforderungen

UC10: Statistik-Daten-Selektion bekannt geben
UC11: Alle Statistik-Daten-Abfrage-Parametersets bekannt geben
UC12: Statistik-Daten-Abfrage-Parameterset entgegennehmen
UC20: Statistik-Daten-Report bekannt geben
UC40: Statistik-Daten-Alarmierung bekannt geben
UC30: Statistik-Daten-Tupel entgegennehmen
UC31: Modifiziertes Statistik-Daten-Tupel entgegennehmen
UC50: Statistik-Daten-Aktuator bekannt geben
UC60: API-Informationen bekannt geben

4.2 Nicht-funktionale Systemanforderungen

NF05: Das System prüft die eingegebenen Daten auf Vollständigkeit. Sind die eingegebenen Daten nicht vollständig, wird der Benutzer darüber informiert.
NF08/09: Eine Datenabfrage, Reportabfrage muss innerhalb 5 Sekunden eine gültige Antwort zurückgeben.
NF10: Der Zugriff auf die Statistik-Daten ist rollenbasiert und mittels Passwort geschützt.
NF14: Das System muss auf einem virtuellen Microsoft Windows Server (ab 2008) lauffähig sein.

5 Leistungsmerkmale, die nicht getestet werden

Aus Zeitgründen können folgende Anforderungen nicht geprüft werden.

5.1 Funktionale Anforderungen

UC100: Benutzerregistrierung entgegennehmen
UC101: Benutzer-Rollen-Wechsel entgegennehmen

5.2 Nicht-funktionale Systemanforderungen

NF01: Die Durchführung der Anwendungsfälle sollte für einen Mitarbeiter mittels Tutorial innerhalb 30 Minuten selbstständig erlernter sein.
NF02: Die Reports sollen komplette Artefakte darstellen, d.h. keine Nachbearbeitung oder Konvertierungen benötigen. Die genaue Form des Artefakts (z.B. pdf, Excel-Dokument) muss noch geklärt werden.
NF03: Der Zugriff auf das System muss während den Bürozeiten (7:00 – 18:00) gewährleistet sein.
NF04: Das System muss von 5 Benutzern gleichzeitig benutzt werden können.
NF06: Die Fehlertoleranz des Systems gemessen in MTBF (Mean Time Between Failure) soll mindestens 30 Tage im ersten Betriebsjahr sein, danach ist eine Reevaluation notwendig.
NF07: Statistik-Daten, die vom System geprüft und verarbeitet wurden, müssen nach einem Betriebsausfall rekonstruierbar sein, sofern die Randbedingung geben ist, dass ein allfälliges Datenbank- und Betriebssystem gegen einen Betriebsausfall gesichert ist.
NF11: Das System darf nur von Mitarbeitern bedient und benutzt werden.
NF12: Das System darf nur in einem Netzwerk betrieben werden, welches vor Fremdzugriffen geschützt ist.
NF13: Änderungen am System müssen innerhalb eines Arbeitstages verbreitet werden können.
NF15: Für die Wartung des Systems dürfen nicht mehr als 12 Arbeitstage pro Jahr aufgewendet werden.
6 Testarten

6.1 Durchgeführte Tests
- Funktionale Tests
- Schnittstellen Tests

6.2 Nicht durchgeführte Tests
Aus Zeitgründen können folgende Testarten nicht berücksichtigt werden.
- Usability Tests
- Reliability Tests
- Third Party Komponenten Tests
- Installation Tests
- Performance Tests
- Long-Term Tests
- Security und Vulnerability Tests
- Failure and Recovery Tests

7 Risiken


In erster Linie muss bei der Prüfung sichergestellt werden, dass bei der Aufbereitung in Form von Selektionen, Reports und Alarmierungen der Daten die Abfrage- und Auswahlbedingungen korrekt ausgewertet werden. Datenabfragen müssen korrekte Ergebnisse liefern.

8 Testziele

Im Rahmen der MAS-Arbeit sind die Ziele bezüglich der Testtätigkeiten wie folgt definiert:
1. Funktionale Tests bezüglich der Datenaufbereitung in Form von Selektionen, Reports und Alarmierungen
2. Funktionale Tests bezüglich der Auswertung von Abfrage- und Auswahlbedingungen
3. Schnittstellen Tests bezüglich der Kommunikation zwischen System und Umsystemen
4. Entwicklung und Bereitstellung der Testumgebung für die jeweiligen Teststufen
5. Bestandesaufnahme der erfüllten Anforderungen bei Abschluss der MAS-Arbeit
6. Bestandesaufnahme der gefundenen Defekte
7. Testfälle sollten für Regressionstests verwendbar sein.
8. Testfälle sollten für die kontinuierliche Integration verwendbar sein.
9 Teststrategie

9.1 Statische Testobjekte

Bei den zu prüfenden statischen Testobjekten handelt es sich grösstenteils um Textdokumente. Diese werden mittels informellen Reviews geprüft und freigegeben.


<table>
<thead>
<tr>
<th>Beschreibung</th>
<th>- Mittels informeller Reviews werden Dokumente auf Korrektheit und Konsistenz durch eine weitere Person geprüft.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testziele</td>
<td>- Dokumente sollen korrekt und konsistent sein.</td>
</tr>
<tr>
<td>Teststartkriterien</td>
<td>- Sämtliche Abschnitte sind eingearbeitet.</td>
</tr>
<tr>
<td></td>
<td>- Dokument wurde vom Autor für ein Review freigegeben.</td>
</tr>
<tr>
<td>Testendkriterien</td>
<td>- Review sämtlicher Abschnitte abgeschlossen.</td>
</tr>
<tr>
<td></td>
<td>- Reviewbefunde im Dokument: xx_ListeDerBefunde.doc eingetragen oder als Word-Datei mit Änderungsverfolgung festgehalten.</td>
</tr>
<tr>
<td>Testabbruchkriterien</td>
<td>- keine</td>
</tr>
<tr>
<td>Abnahmekriterien</td>
<td>- Sämtliche Befunde wurden beurteilt, ob eine Nachbearbeitung notwendig ist.</td>
</tr>
<tr>
<td></td>
<td>- Sämtliche Befunde, die einer Nachbearbeitung bedürfen, wurden im Dokument korrigiert.</td>
</tr>
<tr>
<td>Ausführung</td>
<td>- Microsoft Office 365</td>
</tr>
<tr>
<td>Dokumentation</td>
<td>- Liste der Befunde oder Kopie des Originals mit Review-Kommentaren im Unterverzeichnis «review».</td>
</tr>
<tr>
<td>Ressourcen/Personal</td>
<td>- Reviewer (vgl. Projektleitdokumentation)</td>
</tr>
</tbody>
</table>
9.2 Dynamische Testobjekte

Bei dem zu prüfenden dynamischen Testobjekt handelt es sich um die «Statistik-Daten Auswertung» Software. Das System besteht im Wesentlichen aus zwei Blöcken, welche unabhängig voneinander getestet werden können.

Anzumerken ist, dass im Rahmen der MAS-Arbeit aus Zeitgründen auf die kontinuierliche Integration mittels Continua CI verzichtet wird. Ziel ist es aber, die automatisierten Tests zu einem späteren Zeitpunkt in das System für die kontinuierliche Integration zu übernehmen.

9.2.1 «DataPersistence»-Komponente


Der Fokus liegt auf den selbst entwickelten Modulen, das heisst, die Anbindung der Datenbank und die reinen Daten-Objekte werden nicht explizit getestet.

Die folgende Tabelle und das Package Diagramm geben einen Überblick, welche Teile der «DataPersistence»-Komponente getestet werden:

<table>
<thead>
<tr>
<th>Wird getestet (Typ)</th>
<th>Wird nicht explizit getestet (Typ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller (Klassen)</td>
<td>Configuration (Klassen)</td>
</tr>
<tr>
<td>Helpers (Klassen)</td>
<td>DataPersistenceDb (SQLite-Datenbank)</td>
</tr>
<tr>
<td>Services (Klassen)</td>
<td>DataPersistenceDbContext (Klassen)</td>
</tr>
<tr>
<td>StatisticDataActuatorsController (Schnittstelle)</td>
<td>Models (Klassen)</td>
</tr>
<tr>
<td>StatisticDataAlarmController (Schnittstelle)</td>
<td></td>
</tr>
<tr>
<td>StatisticDataInformationController (Schnittstelle)</td>
<td></td>
</tr>
<tr>
<td>StatisticDataReportController (Schnittstelle)</td>
<td></td>
</tr>
<tr>
<td>StatisticDataSelectionController (Schnittstelle)</td>
<td></td>
</tr>
<tr>
<td>StatisticDataTupleController (Schnittstelle)</td>
<td></td>
</tr>
</tbody>
</table>
9.2.2 «AccessCoordination»-Komponente


Die folgende Tabelle und das Package Diagramm geben einen Überblick, welche Teile der «AccessCoordination»-Komponente getestet werden:
Abbildung 4: Package Diagramm «AccessController» (ohne «Models» Package)
<table>
<thead>
<tr>
<th>Wird getestet (Typ)</th>
<th>Wird nicht explizit getestet (Typ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>API-Controllers (Klassen)</td>
<td>Configuration (Klassen)</td>
</tr>
<tr>
<td>Helpers (Klassen)</td>
<td>AccessCoordinationDb (SQLite-Datenbank)</td>
</tr>
<tr>
<td>Services (Klassen)</td>
<td>AccessCoordinationDbContext (Klassen)</td>
</tr>
<tr>
<td>StatisticDataActuatorsController (Schnittstelle)</td>
<td>EloVarDbContext (Klassen)</td>
</tr>
<tr>
<td>StatisticDataInformationController (Schnittstelle)</td>
<td>Models (Klassen)</td>
</tr>
<tr>
<td>StatisticDataQueryParameterController (Schnittstelle)</td>
<td>UI-Controllers (Klassen)</td>
</tr>
<tr>
<td>StatisticDataTupleController (Schnittstelle)</td>
<td>Views (cshtml-Dateien)</td>
</tr>
<tr>
<td>StatisticDataTupleController (Schnittstelle)</td>
<td>AccountController (Schnittstelle)</td>
</tr>
<tr>
<td>ActuatorsController (Schnittstelle)</td>
<td>ManageController (Schnittstelle)</td>
</tr>
<tr>
<td>QueryController (Schnittstelle)</td>
<td>HomeController (Schnittstelle)</td>
</tr>
<tr>
<td></td>
<td>DataTupleSyncController (Schnittstelle)</td>
</tr>
<tr>
<td></td>
<td>DataTupleImportController (Schnittstelle)</td>
</tr>
</tbody>
</table>
9.4 Teststufen


Die Teststrategie sieht vor, dass die Software «Statistik-Daten-Auswertung» gemäss den aus der Norm IEEE 829 bekannten Teststufen geprüft wird.

Abbildung 5: Allgemeines V-Modell mit Teststufen. Source: [8]
### 9.4.1 Unit- / Komponententests

| Beschreibung | - Es werden einzelne Module und Funktionen der Komponenten getestet, die selbstständig entwickelt wurden.  
| Testziele | - Bestätigung, dass Module und Funktionen sich so verhalten, wie vom Entwickler angedacht. Unittests helfen dem Entwickler beim Überprüfen seiner Arbeit.  
| - Es sind automatisierte Tests | - Sollen für die Regression verwendet werden können.  
| Implementierung | - White-Box-Test  
| Unittest | - Unittests haben keine Abhängigkeiten zu anderen Units oder Komponenten  
| - kurze Laufzeit | - ASP.NET Core 2.0, Microsoft NET Test Sdk, xunit  
| Implementierung | - White-Box-Test  
| - kurze bis mittlere Laufzeit | - ASP.NET Core 2.0, Microsoft NET Test Sdk, xunit, Moq/Moq4, EF Core 2.x In-Memory Database  
| Teststartkriterien | - Funktion, Algorithmus oder Komponente wurde durch den Entwickler getestet und freigegeben.  
| - Das Testobjekt erfüllt gemäss Entwickler die Anforderungen (oder eine Teilmenge davon). |  
| Testendkriterien | - Sämtliche Unit- / Komponententests werden erfolgreich ausgeführt.  
| Testabbruchkriterien | - Die Entwicklung eines einzelnen Tests wird unter-/abgebrochen, falls es nicht möglich ist, die Funktion für einen Test zu isolieren.  
| - Die Entwicklung des Tests wird abgebrochen, wenn das Verhältnis zwischen Testlaufzeit und Zweckdienlichkeit durch den Testentwickler nicht positiv beurteilt werden kann. |  
| Abnahmekriterien | - Sämtliche automatisierte Tests können ohne Benutzerinteraktion ausgeführt werden, und sämtliche Prüfkriterien werden erfüllt.  
| Ausführung | - Microsoft Visual Studio 2017, .NET Core 2.0 Runtime, xunit.runner.visualstudio  
| Dokumentation | - Testbeschreibung im Quellcode  
| - Protokoll/Auswertung im Testreport |  
| Ressourcen/Personal | - Entwickler (vgl. Projektleitdokumentation) |
## 9.4.2 Integrationstests

| Beschreibung | - Es wird das Zusammenspiel von zwei oder mehreren Komponenten getestet. Es wird dabei aber nicht das ganze System integriert und getestet.  
|             | - Die Komponenten werden via deren Schnittstellen getestet. |
| Testziele   | - Sollen gewährleisten, dass komplette Abläufe auch über Komponenten hinweg korrekte Ergebnisse liefern.  
|             | - Sollen für die Regression verwendet werden können. |
| Implementierung | - Black-Box-Test  
|                 | - mittlere Laufzeit  
|                 | - ASP.NET Core 2.0, Microsoft NET Test Sdk, Microsoft.AspNetCore.TestHost, xunit, EF Core 2.x In-Memory Database  
|                 | TestHost ist ein In-Memory Webhost speziell für Integrationstests. Er kann schneller aufgesetzt und wieder abgebaut werden, da die Kommunikation nicht über Sockets geht: [https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests](https://docs.microsoft.com/en-us/aspnet/core/test/integration-tests) |
| Teststartkriterien | - Das Zusammenspiel der beteiligten Komponente wurde durch den Entwickler getestet und freigegeben.  
|                   | - Das Testobjekt erfüllt gemäss Entwickler die Anforderungen (oder eine Teilmenge davon).  
|                   | - Unit- und Komponententest der beteiligten Komponenten (vgl. 9.4.1) sind erfolgreich abgenommen. |
| Testendkriterien | - Sämtliche Integrationstests werden erfolgreich ausgeführt. |
| Testabbruchkriterien | - Die Entwicklung eines Tests wird unter-/abgebrochen, falls es nicht möglich ist, die Funktion für einen Test zu isolieren.  
|                     | - Die Entwicklung des Tests wird abgebrochen, wenn das Verhältnis zwischen Testlaufzeit und Zweckdienlichkeit durch den Testentwickler nicht positiv beurteilt werden kann. |
| Abnahmekriterien | - Sämtliche automatisierte Tests können ohne Benutzerinteraktion ausgeführt werden, und sämtliche Prüfkriterien werden erfüllt. |
| Ausführung | - Microsoft Visual Studio 2017, .NET Core 2.0 Runtime, xunit.runner.visualstudio |
| Dokumentation | - Testbeschreibung im Quellcode  
|               | - Protokoll/Auswertung im Testreport |
| Ressourcen/Personal | - Entwickler (vgl. Projektleitdokumentation) |
### 9.4.3 Automatisierte Systemtests

| Beschreibung | - Es wird das gesamte System gegen die funktionalen Anforderungen getestet. (Nicht-funktionale Anforderungen werden in den automatisierten Systemtests aus Zeitgründen nicht getestet.)  
- Das System wird via dessen Systemschnittstellen getestet. |
| Testziele | - Sollen gewährleisten, dass komplette Szenarien / Use Cases die erwarteten Ergebnisse liefern.  
- Überprüfung ob und welche funktionalen Anforderungen das System erfüllt.  
- Sollen für die Regression verwendet werden können. |
| Implementierung | - Black-Box-Test  
- lange Laufzeit  
- ASP.NET Core 2.0, Microsoft NET Test Sdk, Microsoft.AspNetCore.TestHost, xunit, EF Core 2.x In-Memory Database, Micr0soft Kestrel Web Server |
| Teststartkriterien | - Das Zusammenspiel der beiden beteiligten Komponenten «AccessCoordination» und «DataPersistence» wurde durch den Entwickler getestet und freigegeben.  
- Das Testobjekt erfüllt gemäss Entwickler die Anforderungen (oder eine Teilmengen davon).  
- Unit-, Komponenten und Integrationstest der beteiligten Komponenten (vgl. 9.4.1/9.4.2) sind erfolgreich abgenommen. |
| Testendkriterien | - Sämtliche Systemtests werden erfolgreich ausgeführt. |
| Testabbruchkriterien | - Die Entwicklung des Tests wird abgebrochen, wenn das Verhältnis zwischen Testlaufzeit und Zweckdienlichkeit durch den Testentwickler nicht positiv beurteilt werden kann. |
| Abnahmekriterien | - Sämtliche automatisierte Tests können ohne Benutzerinteraktion ausgeführt werden, und sämtliche Prüfkriterien werden erfüllt. |
| Ausführung | - Microsoft Visual Studio 2017, .NET Core 2.0 Runtime, xunit.runner.visualstudio |
| Dokumentation | - Testbeschreibung im Quellcode  
- Protokoll/Auswertung im Testreport |
| Ressourcen/Personal | - Entwickler (vgl. Projektleitungsdokumentation) |
### 9.4.4 Manuelle Systemtests

| Beschreibung                          | - Es wird das gesamte System gegen die funktionalen Anforderungen getestet. Die manuellen Systemtests ergänzen die automatisierten Systemtests.  
|                                       | - Das System wird nur via Webinterface geprüft. |
| Testziele                            | - Sollen gewährleisten, dass komplette Szenarien / Use Cases die erwarteten Ergebnisse liefern. Überprüfung ob und welche funktionalen Anforderungen das System erfüllt.  
|                                       | - Es werden primär die Szenarien geprüft, für die es keine automatisierten Systemtests gibt. |
| Implementierung                      | - Black-Box-Test  
| Teststartkriterien                   | - Das Zusammenspiel der beiden beteiligten Komponenten «AccessCoordination» und «DataPersistence» wurde durch den Entwickler getestet und freigegeben.  
|                                       | - Das System erfüllt gemäss Entwickler die Anforderungen.  
|                                       | - Unit-, Komponenten, Integrations und automatisierte Systemtests der beteiligten Komponenten (vgl. 9.4.1 / 9.4.2 / 9.4.3) sind erfolgreich abgenommen.  
|                                       | - Der Testfall für den jeweiligen manuellen Test ist freigegeben.  
|                                       | - Das System ist betriebsbereit und mit Testdaten aufdatiert. |
| Testendkriterien                     | - Sämtliche Tests gemäss Testfall sind durchgeführt.  
|                                       | - Befunde sind im Protokoll schriftlich festgehalten. |
| Testabbruchkriterien                 | - Das System ist nicht betriebsbereit, oder der Betrieb kann nicht für die Dauer des Tests aufrechtgehalten werden. |
| Abnahmekriterien                    | - Sämtliche Befunde wurden beurteilt, ob eine Nachbearbeitung notwendig ist. |
| Ausführung                          | - Mozilla Firefox |
| Dokumentation                       | - Testfall-Beschreibung  
|                                       | - Testprotokoll (Protokoll/Auswertung im Testreport) |
| Ressourcen/Personal                  | Entwickler (vgl. Projektleitdokumentation) |
### 9.4.5 Abnahmetest

| Beschreibung | - Es wird zum Ende der MAS-Arbeit eine Bestandesaufnahme gemacht. Dieser Test wird nur einmal durchgeführt.  
|              | - Das System wird nur via Webinterface geprüft. |
| Testziele    | - Soll zeigen, welche geforderten funktionalen Anforderungen das System erfüllt und welche nicht. |
| Implementierung | - Black-Box-Test  
|              | - Das System wird in der Produktivumgebung getestet. |
| Teststartkriterien | - Aufgrund der Rahmenbedingungen dieses Projektes als MAS-Arbeit wird am Ende der Arbeit ein Abnahmetest/Schlussbewertung durchgeführt.  
|              | - Der Testplan für den Abnahmetest ist freigegeben.  
|              | - Das System ist betriebsbereit. |
| Testendkriterien | - Sämtliche Tests gemäss Testplan sind durchgeführt.  
|              | - Befunde sind im Protokoll schriftlich festgehalten. |
| Testabbruchkriterien | - keine |
| Abnahmekriterien | - Sämtliche Befunde wurden schriftlich festgehalten. |
| Dokumentation | - Testplan mit Testfällen  
|              | - Testprotokoll (Protokoll/Auswertung im Testreport) |
| Ressourcen/Personal | - Auftraggeber, Entwickler (vgl. Projektleitdokumentation) |
### 9.5 Übersicht Testobjekte und Testtyp

<table>
<thead>
<tr>
<th>Testobjekt</th>
<th>Testtyp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Informelle Reviews</td>
</tr>
<tr>
<td>Informelle Reviews</td>
<td></td>
</tr>
<tr>
<td>Unitests</td>
<td></td>
</tr>
<tr>
<td>Komponententests</td>
<td></td>
</tr>
<tr>
<td>Integrationstests</td>
<td></td>
</tr>
<tr>
<td>Automatisierte Systemtests</td>
<td></td>
</tr>
<tr>
<td>Manuelle Systemtests</td>
<td></td>
</tr>
<tr>
<td>Abnahmetests</td>
<td></td>
</tr>
</tbody>
</table>

#### 9.5.1 Statische Objekte (3.1)

<table>
<thead>
<tr>
<th>Testobjekt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projektantrag</td>
</tr>
<tr>
<td>Projektleitdokumentation</td>
</tr>
<tr>
<td>Anforderungsspezifikation</td>
</tr>
<tr>
<td>Architekturdokumentation</td>
</tr>
<tr>
<td>Testkonzept</td>
</tr>
<tr>
<td>Systemtest-Testspezifikation</td>
</tr>
<tr>
<td>Testbericht</td>
</tr>
<tr>
<td>Installations- und Betriebsdokumentation</td>
</tr>
<tr>
<td>Benutzerdokumentation</td>
</tr>
<tr>
<td>Statisches Software-Produkt</td>
</tr>
</tbody>
</table>

#### 9.5.2 DataPersistence (3.2.4)

<table>
<thead>
<tr>
<th>Testobjekt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration (Klassen)</td>
</tr>
<tr>
<td>Models (Klassen)</td>
</tr>
<tr>
<td>Controller (Klassen)</td>
</tr>
<tr>
<td>Helpers (Klassen)</td>
</tr>
<tr>
<td>Services (Klassen)</td>
</tr>
<tr>
<td>StatisticDataActuatorsController (Schnittstelle)</td>
</tr>
<tr>
<td>StatisticDataAlarmController (Schnittstelle)</td>
</tr>
<tr>
<td>StatisticDataInformationController (Schnittstelle)</td>
</tr>
<tr>
<td>StatisticDataReportController (Schnittstelle)</td>
</tr>
<tr>
<td>StatisticDataSelectionController (Schnittstelle)</td>
</tr>
<tr>
<td>StatisticDataTupleController (Schnittstelle)</td>
</tr>
<tr>
<td>DataPersistenceDb (SQLite-Datenbank)</td>
</tr>
<tr>
<td>DataPersistenceDbContext (Klassen)</td>
</tr>
</tbody>
</table>
### 9.5.3 AccessCoordination (3.2.4)

| Configuration (Klassen) | Models (Klassen) | API-Controllers (Klassen) | Helpers (Klassen) | Services (Klassen) | StatisticDataActuatorsController (Schnittstelle) | StatisticDataInformationController (Schnittstelle) | StatisticDataQueryParameterController (Schnittstelle) | StatisticDataTupleController (Schnittstelle) | AccessCoordinationDb (SQLite-Datenbank) | AccessCoordinationDbContext (Klassen) | EloVarDbContext (Klassen) | UI-Controllers (Klassen) | Views (cshtml-Dateien) | AccountController (Schnittstelle) | ManageController (Schnittstelle) | HomeController (Schnittstelle) | DataTupleSyncController (Schnittstelle) | DataTupleImportController (Schnittstelle) | ActuatorsController (Schnittstelle) | QueryController (Schnittstelle) |
|-------------------------|-----------------|--------------------------|-------------------|--------------------|------------------------------------------------|------------------------------------------------|------------------------------------------------|------------------------------------------------|-----------------------------|--------------------------------|-----------------------------|--------------------------|---------------------------|--------------------------|-----------------------------|-----------------------------|-----------------------------|--------------------------------|-----------------------------|-----------------------------|-----------------------------|

| Expliziter Testfall |

| Impliziter Testfall, d.h. der Testfall betrifft das Testobjekt, der Testfall ist aber nicht explizit für das Testobjekt |

### 10 Zeit-/Arbeitsplan

Statistik-Daten Auswertung

Masterarbeit HSR-MAS-SE 2016

Testbericht
Inhaltsverzeichnis

Zu diesem Dokument ........................................................................................................ 3
1.1 Zweck ......................................................................................................................... 3
1.2 Gültigkeit .................................................................................................................... 3
1.3 Verfügbarkeit des Dokumentes .................................................................................. 3

2 Weiterführende Informationen ..................................................................................... 4
2.1 Definitionen, Akronym und Abkürzungen ................................................................. 4
2.2 Ergänzende Dokumente ............................................................................................ 4
2.3 Prozessbezogene Dokumente ..................................................................................... 4
2.4 Versionshistorie ......................................................................................................... 4
2.5 Verteilung ................................................................................................................... 4

3 Testdurchführung ......................................................................................................... 5
3.1 Übersicht Testobjekte ............................................................................................... 5
3.2 Übersicht Testtyp ....................................................................................................... 5
3.2.1 Statische Objekte .................................................................................................. 5
3.2.2 DataPersistence .................................................................................................... 6
3.2.3 AccessCoordination ......................................................................................... 6
3.3 Änderung .................................................................................................................. 7
3.3.1 AccessCoordination ......................................................................................... 7

4 Testresultate ................................................................................................................. 7
4.1 Reviews ..................................................................................................................... 7
4.1.1 Statische Objekte .................................................................................................. 7
4.2 Automatisierte Tests ............................................................................................... 8
4.2.1 DataPersistence .................................................................................................. 8
4.2.2 AccessCoordination ......................................................................................... 8
4.3 Manuelle Systemtests ............................................................................................ 9
4.3.1 Nachprüfung ...................................................................................................... 9

5 Abnahmetest .............................................................................................................. 10
5.1 Testfälle ..................................................................................................................... 10
5.2 System Under Test ................................................................................................. 10
5.3 Bemerkung .............................................................................................................. 10

6 Bewertung .................................................................................................................. 10

7 Nicht gelöste Problem ............................................................................................... 10

8 Anhang ....................................................................................................................... 11
8.1 Testprotokolle .......................................................................................................... 11
Zu diesem Dokument

1.1 Zweck

Dieser Testbericht ist eine Zusammenfassung, Bestandesaufnahme und Beurteilung der Testresultate am Ende der MAS-Arbeit.

1.2 Gültigkeit

Dieses Dokument ist über die ganze Projektdauer gültig.

1.3 Verfügbarkeit des Dokumentes

Dieses Dokument befindet sich im Projektverzeichnis: 12_Testaktivitaeten
2 Weiterführende Informationen

2.1 Definitionen, Akronyme und Abkürzungen

Sämtliche Definitionen, Akronyme und Abkürzungen werden zentral in einem Projektglossar dokumentiert. Das Projektglossar wird kontinuierlich angepasst.
Das Projektglossar befindet sich im Projektverzeichnis: 09_Glossar

2.2 Ergänzende Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Testkonzept: «Statistik-Daten Auswertung, Testkonzept», VER003</td>
</tr>
</tbody>
</table>

2.3 Prozessbezogene Dokumente

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Beschreibung</th>
</tr>
</thead>
</table>

2.4 Versionshistorie

(nicht öffentlich)

2.5 Verteilung

<table>
<thead>
<tr>
<th>Ver.</th>
<th>Name</th>
<th>Rolle / Titel / Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>002</td>
<td>MAS-Arbeit</td>
<td>MAS-Arbeit</td>
</tr>
</tbody>
</table>
3 Testdurchführung

3.1 Übersicht Testobjekte

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Objekt</th>
<th>Version</th>
<th>Prüfung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Projektantrag</td>
<td>VER011</td>
<td>Nein</td>
</tr>
<tr>
<td>2</td>
<td>Projektleitdokumentation</td>
<td>VER005</td>
<td>Ja</td>
</tr>
<tr>
<td>3</td>
<td>Anforderungspezifikation</td>
<td>VER006</td>
<td>Ja</td>
</tr>
<tr>
<td>4</td>
<td>Architekturdokumentation</td>
<td>VER006</td>
<td>Ja</td>
</tr>
<tr>
<td>5</td>
<td>Testkonzept</td>
<td>VER003</td>
<td>Ja</td>
</tr>
<tr>
<td>6</td>
<td>Systemtest-Testspezifikation</td>
<td>VER004</td>
<td>Ja</td>
</tr>
<tr>
<td>7</td>
<td>Testbericht</td>
<td>VER002</td>
<td>Ja</td>
</tr>
<tr>
<td>8</td>
<td>Installations- und Betriebsdokumentation</td>
<td>Keine Versionierung in Confluence</td>
<td>Ja</td>
</tr>
<tr>
<td>9</td>
<td>Benutzerdokumentation</td>
<td>Keine Versionierung in Confluence</td>
<td>Ja</td>
</tr>
<tr>
<td>10</td>
<td>AccessCoordination.exe</td>
<td>v1.0.0.0</td>
<td>Ja</td>
</tr>
<tr>
<td>11</td>
<td>DataPersistence.exe</td>
<td>v1.0.0.0 (Nachprüfung mittels v1.0.0.1)</td>
<td>Ja</td>
</tr>
</tbody>
</table>

3.2 Übersicht Testtyp

Gemäss Testkonzept Kapitel: 9.5

<table>
<thead>
<tr>
<th>Testobjekt</th>
<th>Testtyp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Informelle Reviews</td>
</tr>
<tr>
<td>Projektantrag</td>
<td></td>
</tr>
<tr>
<td>Projektleitdokumentation</td>
<td></td>
</tr>
<tr>
<td>Anforderungspezifikation</td>
<td></td>
</tr>
<tr>
<td>Architekturdokumentation</td>
<td></td>
</tr>
<tr>
<td>Testkonzept</td>
<td></td>
</tr>
<tr>
<td>Testfall (manuelle Systemtests)</td>
<td></td>
</tr>
<tr>
<td>Testbericht</td>
<td></td>
</tr>
<tr>
<td>Installations- und Betriebsdokumentation</td>
<td></td>
</tr>
<tr>
<td>Benutzerdokumentation</td>
<td></td>
</tr>
</tbody>
</table>

3.2.1 Statische Objekte

Projektantrag
Projektleitdokumentation
Anforderungspezifikation
Architekturdokumentation
Testkonzept
Testfall (manuelle Systemtests)
Testbericht
Installations- und Betriebsdokumentation
Benutzerdokumentation
### 3.2.2 DataPersistence

<table>
<thead>
<tr>
<th>Configuration (Klassen)</th>
<th>Models (Klassen)</th>
<th>Controller (Klassen)</th>
<th>Helpers (Klassen)</th>
<th>Services (Klassen)</th>
<th>StatisticDataActuatorsController (Schnittstelle)</th>
<th>StatisticDataAlarmController (Schnittstelle)</th>
<th>StatisticDataInformationController (Schnittstelle)</th>
<th>StatisticDataReportController (Schnittstelle)</th>
<th>StatisticDataSelectionController (Schnittstelle)</th>
<th>StatisticDataTupleController (Schnittstelle)</th>
<th>DataPersistenceDb (SQLite-Datenbank)</th>
<th>DataPersistenceDbContext (Klassen)</th>
</tr>
</thead>
</table>

### 3.2.3 AccessCoordination

| Configuration (Klassen)          | Models (Klassen)    | API-Controllers (Klassen) | Helpers (Klassen) | Services (Klassen) | StatisticDataActuatorsController (Schnittstelle) | StatisticDataInformationController (Schnittstelle) | StatisticDataQueryParameterController (Schnittstelle) | StatisticDataTupleController (Schnittstelle) | AccessCoordinationDb (SQLite-Datenbank) | AccessCoordinationDbContext (Klassen) | EloVarDbContext (Klassen) | UI-Controllers (Klassen) | Views (cshtml-Dateien) | AccountController (Schnittstelle) | ManageController (Schnittstelle) | HomeController (Schnittstelle) | DataTupleSyncController (Schnittstelle) | DataTupleImportController (Schnittstelle) | ActuatorsController (Schnittstelle) | QueryController (Schnittstelle) |
|---------------------------------|--------------------|--------------------------|-------------------|-------------------|------------------------------------------------|------------------------------------------------|------------------------------------------------|------------------------------------------------|---------------------------------|-----------------------------|---------------------------------|-----------------------------|--------------------------|--------------------------|-----------------------------|---------------------------------|-----------------------------|-----------------------------|---------------------------------|-----------------------------|-----------------------------|

---

**Status:** RELEASE  
**Version:** 002  
**Classification:** PUBLIC
### 3.3 Änderung

#### 3.3.1 AccessCoordination

<table>
<thead>
<tr>
<th>Services (Klassen)</th>
<th>Test</th>
<th>Test</th>
<th>Test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>StatisticDataActuatorsController (Schnittstelle)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StatisticDataInformationController (Schnittstelle)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StatisticDataQueryParameterController (Schnittstelle)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StatisticDataTupleController (Schnittstelle)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


### 4 Testresultate

#### 4.1 Reviews

<table>
<thead>
<tr>
<th>Testobjekt</th>
<th>Testtyp</th>
<th>Resultat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informelle Reviews</td>
<td>Unitests</td>
<td>Integrationstests</td>
</tr>
<tr>
<td>Projektantrag</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Projektledokumentation</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Anforderungsspezifikation</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Architekturdokumentation</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Testkonzept</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Testfall (manuelle Systemtests)</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Testbericht</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Installations- und Betriebsdokumentation</td>
<td></td>
<td>Review</td>
</tr>
<tr>
<td>Benutzerdokumentation</td>
<td></td>
<td>Review</td>
</tr>
</tbody>
</table>

**4.1.1 Statische Objekte**

<table>
<thead>
<tr>
<th>Projekteintrag</th>
<th>Resultat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review</td>
<td></td>
</tr>
<tr>
<td>Projektledokumentation</td>
<td>Review</td>
</tr>
<tr>
<td>Anforderungsspezifikation</td>
<td>Review</td>
</tr>
<tr>
<td>Architekturdokumentation</td>
<td>Review</td>
</tr>
<tr>
<td>Testkonzept</td>
<td>Review</td>
</tr>
<tr>
<td>Testfall (manuelle Systemtests)</td>
<td>Review</td>
</tr>
<tr>
<td>Testbericht</td>
<td>Review</td>
</tr>
<tr>
<td>Installations- und Betriebsdokumentation</td>
<td>Review</td>
</tr>
<tr>
<td>Benutzerdokumentation</td>
<td>Review</td>
</tr>
</tbody>
</table>
# 4.2 Automatisierte Tests

<table>
<thead>
<tr>
<th>Testobjekt</th>
<th>Testtyp</th>
<th>Resultat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informelle Reviews</td>
<td>Unitests</td>
<td>Komponententests</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## 4.2.1 DataPersistence

| Configuration (Klassen) | | | | |
| Models (Klassen) | | | | |
| Controller (Klassen) | | | | |
| Helpers (Klassen) | | | | |
| Services (Klassen) | | | | |
| StatisticDataActuatorsController (Schnittstelle) | | | | |
| StatisticDataAlarmController (Schnittstelle) | | | | |
| StatisticDataInformationController (Schnittstelle) | | | | |
| StatisticDataReportController (Schnittstelle) | | | | |
| StatisticDataSelectionController (Schnittstelle) | | | | |
| StatisticDataTupleController (Schnittstelle) | | | | |
| DataPersistenceDb (SQLite-Datenbank) | | | | |
| DataPersistenceDbContext (Klassen) | | | | |

### 4.2.1.1 DataPersistence

| Configuration (Klassen) | | | | |
| Models (Klassen) | | | | |
| Controller (Klassen) | | | | |
| Helpers (Klassen) | | | | |
| Services (Klassen) | | | | |
| StatisticDataActuatorsController (Schnittstelle) | | | | |
| StatisticDataAlarmController (Schnittstelle) | | | | |
| StatisticDataInformationController (Schnittstelle) | | | | |
| StatisticDataReportController (Schnittstelle) | | | | |
| StatisticDataSelectionController (Schnittstelle) | | | | |
| StatisticDataTupleController (Schnittstelle) | | | | |
| DataPersistenceDb (SQLite-Datenbank) | | | | |
| DataPersistenceDbContext (Klassen) | | | | |

## 4.2.2 AccessCoordination

| Configuration (Klassen) | | | | |
| Models (Klassen) | | | | |
| API-Controllers (Klassen) | | | | |
| Helpers (Klassen) | | | | |
| Services (Klassen) | | | | |
| StatisticDataActuatorsController (Schnittstelle) | | | | |
| StatisticDataAlarmController (Schnittstelle) | | | | |
| StatisticDataInformationController (Schnittstelle) | | | | |
| StatisticDataQueryParameterController (Schnittstelle) | | | | |
| StatisticDataTupleController (Schnittstelle) | | | | |
| AccessCoordinationDb (SQLite-Datenbank) | | | | |
| AccessCoordinationDbContext (Klassen) | | | | |
| AccessCoordinationDbContext (Klassen) | | | | |
| EloVarDbContext (Klassen) | | | | |
| UI-Controllers (Klassen) | | | | |
| Views (cshtml-Dateien) | | | | |
### 4.3 Manuelle Systemtests

<table>
<thead>
<tr>
<th>Testobjekt</th>
<th>Testcase Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual-UC10 UC10</td>
<td>UC10: Statistik-Daten-Selektion bekannt geben</td>
</tr>
<tr>
<td>Manual-UC20 UC20</td>
<td>UC20: Statistik-Daten-Report bekannt geben</td>
</tr>
<tr>
<td>Manual-UC40 UC40</td>
<td>UC40: Statistik-Daten-Alarmierung bekannt geben</td>
</tr>
<tr>
<td>Manual-UC11 UC11</td>
<td>UC11: Alle Statistik-Daten-Abfrage-Parametersets bekannt geben</td>
</tr>
<tr>
<td>Manual-UC12 UC12</td>
<td>UC12: Statistik-Daten-Abfrage-Parameterset entgegennnehmen</td>
</tr>
<tr>
<td>Manual-UC30 UC30</td>
<td>UC30: Statistik-Daten-Tupel entgegennnehmen</td>
</tr>
<tr>
<td>Manual-UC31 UC31</td>
<td>UC31: Modifiziertes Statistik-Daten-Tupel entgegennnehmen</td>
</tr>
<tr>
<td>Manual-UC50 UC50</td>
<td>UC50: Statistik-Daten-Aktuator bekannt geben</td>
</tr>
</tbody>
</table>

#### 4.3.1 Nachprüfung

5 Abnahmetest

Für den Abnahmetest wurden die Testfälle der manuellen Systemtests verwendet. Im Gegensatz zu den Systemtests wurden die Testfälle aber auf dem Pilot-System mit produktiven Datenbanken ausgeführt.

5.1 Testfälle

<table>
<thead>
<tr>
<th>Testobjekt</th>
<th>Erwartung</th>
<th>Ergebnis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual-UC10</td>
<td>UC10: Statistik-Daten-Selektion bekannt geben</td>
<td>grün</td>
</tr>
<tr>
<td>Manual-UC20</td>
<td>UC20: Statistik-Daten-Report bekannt geben</td>
<td>grün</td>
</tr>
<tr>
<td>Manual-UC40</td>
<td>UC40: Statistik-Daten-Alarmierung bekannt geben</td>
<td>grün</td>
</tr>
<tr>
<td>Manual-UC11</td>
<td>UC11: Alle Statistik-Daten-Abfrage-Parametersets bekannt geben</td>
<td>n/a</td>
</tr>
<tr>
<td>Manual-UC12</td>
<td>UC12: Statistik-Daten-Abfrage-Parameterset entgegennehmen</td>
<td>grün</td>
</tr>
<tr>
<td>Manual-UC30</td>
<td>UC30: Statistik-Daten-Tupel entgegennehmen</td>
<td>grün</td>
</tr>
<tr>
<td>Manual-UC31</td>
<td>UC31: Modifiziertes Statistik-Daten-Tupel entgegennehmen</td>
<td>grün</td>
</tr>
<tr>
<td>Manual-UC50</td>
<td>UC50: Statistik-Daten-Aktuator bekannt geben</td>
<td>grün</td>
</tr>
</tbody>
</table>

5.2 System Under Test

Release-Paket: StatisticDataAnalysisService_RC_20180823.zip

<table>
<thead>
<tr>
<th>AccessCoordination</th>
<th>DataPersistence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version: 1.0.0.0</td>
<td>Version: 1.0.0.1</td>
</tr>
<tr>
<td>ApiVersion: v1</td>
<td>ApiVersion: v1</td>
</tr>
</tbody>
</table>

5.3 Bemerkung

Grund: Beim Produktivsystem ist es nicht möglich zu überprüfen, ob das System alle Statistik-Daten-Abfrage-Parametersets darstellt, die in der Datenbank gespeichert sind.

6 Bewertung

Das System wird für den Pilot-Betrieb freigegeben.

7 Nicht gelöste Probleme

Das Testframework für automatisierte Tests ist zuwenig ausgereift und muss erweitert werden. Diese technische Schuld ist bereits in der Architekturdokumentation festgehalten.

<table>
<thead>
<tr>
<th>TS08 Testframework</th>
<th>Unittest haben zu viel repetitiven Code. Das Framework sollte dahingehend erweitert werden, das mehrfach verwendete Funktionen zentral verwaltet werden.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS09 Testframework</td>
<td>Das Testframework muss um eine Möglichkeit erweitert werden, dass Testdaten zentral erzeugt werden können. Momentan müssen sie oft on-the-fly erzeugt werden. Test sollten parametrisierbar sein, so dass systematisch erzeugte Testdaten verwendet werden können.</td>
</tr>
</tbody>
</table>
8 Anhang

8.1 Testprotokolle

8.1.1 Reviews (4.1)

Zugehörige Review Dokumente befinden sich in einem separaten Unterverzeichnis im jeweiligen Verzeichnis des freigegebenen Dokuments.

8.1.2 Automatisierte Tests (4.2)

Ausgabe Xunit für DataPersistence

```
8.1.1 Reviews (4.1)

Zugehörige Review Dokumente befinden sich in einem separaten Unterverzeichnis im jeweiligen Verzeichnis des freigegebenen Dokuments.

8.1.2 Automatisierte Tests (4.2)

Ausgabe Xunit für DataPersistence
```