

1 Funktionale Anforderungen

1.1 User Stories

1.1.1 Systemanforderung

- 3 Benutzergruppen: IT-Manger, Hausmeister, Mieter
- IDs zur Festlegung der Zugriffsrechte
- Client-Server Kommunikationsprotokoll

1.1.2 Temperatursteuering:

- Mieter kann Temperatur aus zulässigem Bereich auswählen und einstellen (für seine Wohneinheit)
 - Priorität: Muss
 - Aufwand: 1 Tag
 - Details: Jeder Benutzer, der Anfragen an das System stellt, hat eine eigene ID. Anhand der ID wird ermittelt, welche Parameter er verstellen darf
 - Anfrage zur Änderung der Temperaturparameter mit Benutzer-ID
 - Auswahl des Raumes
 - Abfrage des Sollwerts Temperatur
 - Abfrage der IST-Temperatur
 - Neueingabe der gewünschten Temperatur
 - Überprüfung, ob Temperatur im zulässigen Bereich
 - Evtl. Korrektur auf Max- oder Min-Wert
 - Rückmeldung, dass geänderte Parameter übernommen wurden
- Admin kann einen zulässigen Temperaturbereich festlegen
 - Priorität: Muss
 - Aufwand: 1 Stunde
 - Details:
 - Der Temperaturbereich wird in zwei Variablen nach dem unten beschriebenen Format gespeichert.
 - Temperatur-Format: Die Temperatur wird in °C plus ein Dezimal gespeichert, z.B. 20,1°C wird als 201 in eine 32-bit signed integer gespeichert.
 - Der Server überprüft, ob die eingegebene Temperatur einer Anfrage innerhalb des Bereichs liegt.
 - Bei Angabe einer nicht zulässige Temperatur wird eine Fehlermeldung erzeugt.
 - Bei Angabe einer zulässigen Temperatur wird eine Systemantwort gesendet.
- Das Leitsystem reguliert die Temperatur nach einem vordefinierten Zeitintervall
 - Priorität: Muss
 - Aufwand: 2 Stunden
 - Details:
 - Die aktuelle Temperatur wird abgefragt
 - Die aktuelle Temperatur wird mit dem Sollwert verglichen
 - Falls die Abweichung zwischen Ist- und Sollwert größer als eine vordefinierte Toleranz ist, dann werden die Aktoren angesteuert, um die Temperatur zu regeln.
- Admin kann den Temperaturverlauf über den Tag festlegen (Nachtabsenkung)
 - Priorität: sollte

- Aufwand: 1 Stunde
- Details: Es gibt die Betriebsarten Tages- und Nachtbetrieb. Die Temperatur für die Nachtabenkung wird relativ zur Tagestemperatur geändert. Die Werte gelten für das ganze Gebäude.
 - Anfrage zur Änderung der Temperaturkurven mit Benutzer ID
 - Änderung des Wertes für die Nachtabenkung

1.1.3 Benutzer Verwaltung

- Admin kann neue Benutzer anlegen
 - Priorität: Muss
 - Aufwand: 2 Stunde
 - Details: Alle Mieter, sowie Hausmeister und Admin sind mit Ihrer ID in einer zentralen Datenbank gespeichert
 - Vorname char 2..10 Zeichen
 - Name char 2..30 Zeichen
 - Rolle enum
 - WohnungsID 32-bit unsigned integer

1.2 Fachklassendiagramm

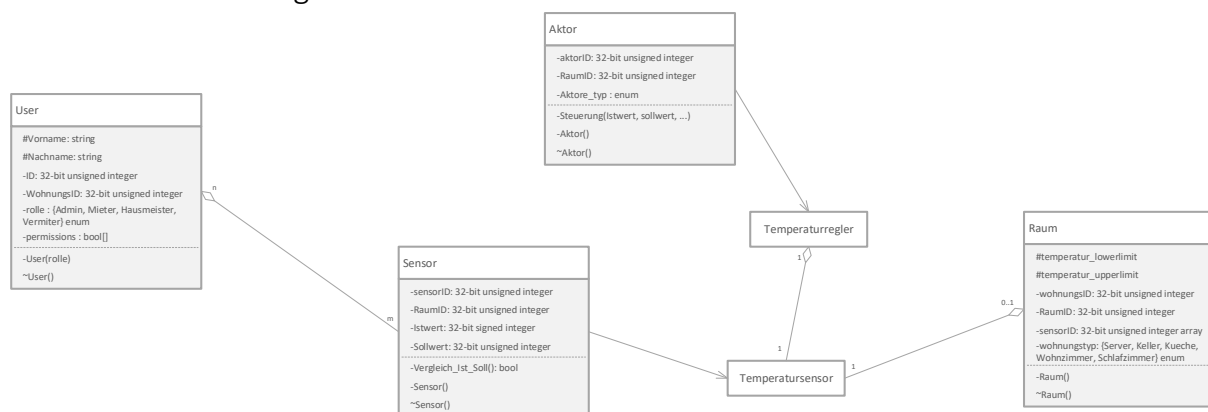


Abbildung 1. Fachklassendiagramm des Leitsystems

1.3 Zu verwendende Dateien

1.3.1 Logdateien (.log)

Die log Dateien speichern Information über die Ereignisse (Events), die im Leitsystem auftreten, z.B. die Umstellung eines Sensorarbeitspunkts, die Erstellung eines neuen Benutzers.

Es wird zwischen drei Arten von Ereignissen unterschieden: Benutzer-, System- und Fehlerereignissen. Jede log Zeile besteht aus Datum- und Zeitstempel, Eventtyp, sowie Event spezifischen Parametern. User-events beinhalten die WohnungsID, aus der das Event erzeugt wurde, sowie Informationen über die ausgeführte Funktion (Funktionsname und ggf. Funktionsparameter). System-Events werden für systemübergreifende Änderung und Abfragen reserviert, die von einem Administrator durchgeführt werden können. Fehler, die auf dem Leitsystem auftreten, werden in der Logdatei dokumentiert, wo die Fehlernummer, -text, und -quelle beinhaltet ist.

Beispiel:

```

2020-02-02 22:52:31 - user - WohnungsID:12345 - newTemp:189;prevTemp:210
2020-02-02 22:53:31 - system - newSoftwareUpdateDistribution:1.1.0.0
2020-02-02 22:54:31 - error - errorCode:2 - errorInfo:"Connection Timeout" -
errorSource:WohnungsID:12345;UserID:78;SensorID:912

```

Abbildung 2. Beispiel einer Logdatei des Gebäude Leitsystem

1.3.2 Konfigurationsdateien (.ini)

Die Konfigurationsdateien beinhalten Parameter, die das System und die Systemkomponente definieren. Es werden drei Konfigurationsdateien erstellt: eine für die Systemeigenschaften und Abhängigkeiten von übergeordneten Komponenten (system.ini), eine für die Räume und eine für die Geräte. Jede Konfigurationsdatei besteht aus Abschnitten, die durch eckige Klammern eingeleitet werden.

System (inkl. Wohnungsdaten), Räume, Geräte (inkl. Sensoren und Aktoren)

```

; _____system.ini_____
[information]
version = "1.0"

[system]
sync_invertvall_ms = 2000
wohnung_N = 20
raum_N = 64
benutzer_N = 45
geraete_N = 250
temp_toleranz = 5
temp_max = 260
temp_min = -100

[wohnung_12345]
wohnungsID = 12345
raumIDs = "584;342;122;754"
userIDs = "572;424;878"

; _____wohnungsraume.ini_____
[raeume]
raumTypen = "Kueche;Wohnzimmer;Badezimmer;Schlafzimmer;Keller"

[raum_584]
raumID = 584
raumTyp = "Kueche"
gerateIDs = "4721;8726;7841;4722;8727;7843"

; _____geraete.ini_____
[geraete]
geraetTypen = "sensor;aktor;tuer;fenster"

[sensoren]
sensorTypen = "Temperatur;Licht;Druck"

[geraet_4721]
geraetID = 4721
geraetTyp = "Sensor"
sensorTyp = "Temperatur"

```

Abbildung 3. Beispieldateien für die Konfigurationsdaten des Gebäude Leitsystems

1.4 Schnittstelle zwischen Steuerapplikation und Leitsystem

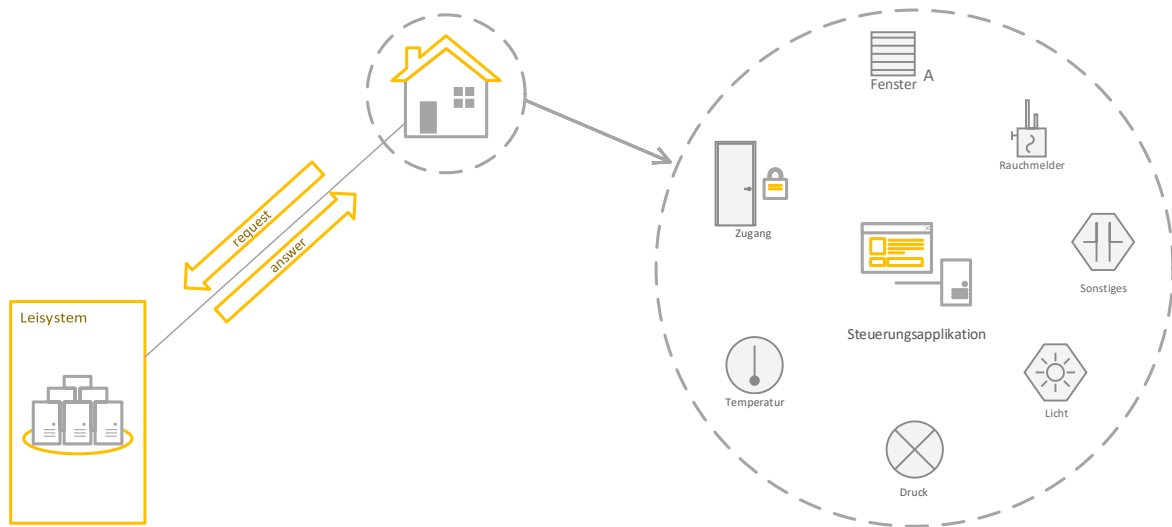


Abbildung 4. Darstellung der Abhängigkeiten zwischen Leitsystem, Steuerapplikation und den Peripheriegeräten

Die Kommunikation zwischen der Steuerapplikation und dem Leitsystem erfolgt auf Basis von Zeichenketten, die aus den folgenden vier Sub-strings geschrieben werden, die durch ein Semikolon getrennt werden:

<request/answer>	besagt, ob es sich um eine Abfrage oder eine Antwort handelt
<sourceID>	kann entweder Server oder eine ClientID sein, z.B. WohnungsID, oder RaumID
<function>	Name der jeweiligen Funktion, die aufgerufen wurde, oder die aufgerufen werden soll
"<parameters/return>"	Dieser Parameter wird in Anführungszeichen gesetzt, damit mehrere Funktionsparametern durch ein Trennzeichen getrennt werden können. Im Falle einer Antwort wird das Ergebnis von der aufgerufenen Funktion geschrieben.

2 Nicht funktionale Anforderungen:

- Die Synchronisationsrate muss so eingestellt werden, dass der Server nicht im Leerlaufzustand belastet wird.
 - Priorität: Muss
 - Aufwand: 1 Stunde
 - Details:
 - Die Synchronisationsrate muss einstellbar sein.
 - Das System befindet sich im Leerlaufzustand, wenn keine Benutzer-Events erzeugt werden. Regelungsgrößen, wie z.B. die Temperatur werden weiterhin in diesem Zustand gesteuert und geregelt.
- Das System soll modular aufgebaut werden.
 - Priorität: Sollte
 - Aufwand: 1 Stunde
 - Details:
 - Es sollen Objektklassen erstellt werden, die die Vererbung und Überschreibung von bestehenden Funktionen ermöglichen.
- Das Leitsystem soll die Konfigurationsdaten für jede Wohnung in eine oder mehrere Dateien speichern.
 - Priorität: Muss
 - Aufwand: 1 Stunden
 - Details:
 - Die Daten sollen u.a. WohnungsID, RaumID, SensorID, AktorID beinhalten (siehe Abschnitt 1.3.2)
- Das Leitsystem erzeugt log Dateien
 - Priorität: Muss
 - Aufwand: 1 Stunde
 - Details:
 - Das Leitsystem schreibt in einer lokal gespeicherten Logdatei folgende Daten: (siehe Abschnitt 1.3.1)
 - Event Typ: Benutzer-, Fehler- oder System-Event
 - BenutzerID
 - Zeit- und Datumstempel
 - Event spezifische Parameter
 - Die Logdateien werden das .log Format verwenden
 - Jede 90 Tagen werden die Logdateien gelöscht.