Konzept

Die Applikation “Smart Room” soll als digitaler Zwilling eines Raumes dienen, dessen Raumausstattung, wie etwa Türen, Fenster, Beleuchtung und Ventilatoren, manuell lenkbar ist. Zu dieser Raumausstattung gehören Sensoren, die, die Anzahl der Personen in einem Raum sowie den CO2-Wert als auch die Raumtemperatur erkennen können. Neben den manuellen Funktionen, wie etwa die Funktion, Türen zu öffnen und zu schließen, soll die Applikation auch sicherstellen, dass beim Auftreten gewisser Bedingungen, wie etwa die Anwesenheit mehr als 0 Personen in einem Raum, die Raumausstattung automatisch dementsprechend reagiert. Zudem dient die Applikation als Anzeige und graphische Visualisierung gewisser Daten, wie etwa vorhandene Räume, Raumausstattung, Temperaturen, CO2-Werte und Anzahl der Personen. Die Informationen über die Räume werden entweder manuell angelegt oder mittels einer .CSV-Datei importiert, die dann in eine Datenbank verwandelt wird, auf die die Applikation Zugriff hat.

Vorgehensweise

Zur Umsetzung der Applikation wird in einer Gruppe von drei Mitgliedern, bestehend aus Abir Sikder, Stefan Pilgerstorfer und Nuray Seker eine Java-Applikation entwickelt. Im Rahmen der agilen Softwareentwicklung gibt für dieses Vorhaben 3 Sprints, wobei die Sprints mit Tools wie ZenHub verwaltet werden.

Technologische Voraussetzungen

Die Applikation ist als ein Java-Programm konzipiert und soll auf Windows, Mac und Linux laufen. Dafür reicht es, die Applikation herunterzuladen oder mit einem Datenträger zu importieren.

Funktionale Anforderungen

* Basic
* Importieren der Daten aus einer CSV-Datei in eine ER-Datenbank
* Räume erstellen, bearbeiten und löschen
* Informationen über Räume anzeigen lassen
  + ID
  + Größe
  + Anzahl der Türe
  + Anzahl der Fenster
  + Anzahl der Lampen
  + Anzahl der Ventilatoren
* Verfügbare Räume anzeigen lassen und als .csv speichern
* Anzeigen der CO2-Werte, Temperaturen und die Anzahl der Personen in Echt-Zeit durch Liniendiagramme
* Anzeigen der Informationen der jeweiligen Räume
* Hinzufügen zufällige Werte (CO2, Temperatur, Anzahl der Personen) für Räume
* Visuelle Darstellung der CO2-Werte, Temperaturen, Lichter, Ventilatoren, Fenster und Türen für alle Räume
* Remote Control
* Türen sperren und entsperren
* Beleuchtung ein- und ausschalten
* Fenster öffnen und schließen
* Ventilatoren ein- und ausschalten
* Security
* Alarm aussenden und Türen öffnen, falls Raum über 70 Grad ist
* Energy Saving
* Einschalten der Beleuchtungen, wenn Personen im Raum anwesend sind und Ausschalten, wenn keine mehr im Raum sind
* Ausschalten laufender Geräte, wenn keine Personen im Raum anwesend sind
* Air Quality
* Fenster öffnen und Ventilatoren einschalten, wenn CO2-Werte zu hoch
* Die Räume in der Applikation bekommen je nach CO2-Werten eigene Farben
  + Grün, bei < 800 ppm
  + Gelb, bei > 800 ppm und < 1000 ppm
  + Rot, bei > 1000 ppm

Die Funktionen werden in den Use-Cases genauer beschrieben

Nicht-Funktionale Anforderungen der Applikation

Qualitäts-Anforderungen

Die Funktionalitäten des Projekts “Smart Room” sind erfüllt, wenn alle Funktionen, die in den Use-Cases beschrieben wurden, ausführbar sind und die Applikation in 95% der Fälle ohne Störungen funktioniert. Bei den restlichen 5%, werden Fehlerbehebungen oder Wartungsarbeiten durchgeführt.

Um die eventuelle Erweiterung bzw. Anpassung der Funktionalitäten in der Zukunft zu ermöglichen, beschränken wir uns auf gängige, zukunftssichere Programmiersprachen. Grundsätzlich soll das User-Interface über eine intuitive und leichte Bedienung verfügen, sodass es zu keiner langwierigen Einschulung führt.

Anforderungen für die Erstellungsphase

Das Projekt soll inkrementell erfolgen. Am Anfang des Projekts soll das Entwicklungsteam ein Konzept ausarbeiten. Danach werden ein UML-Diagramm und ein ER-Diagramm erstellt um später den Ablauf bzw. die funktionalen Anforderungen sichtbar zu machen.

Die Implementierung soll erst dann gestartet werden, wenn die obigen Anforderungen erfüllt worden sind.

Anforderungen an die Sicherheit

Die Daten der jeweiligen Räume sollen in Datenbanken gespeichert sein. Zugang zu den persönlichen Daten hat allein nur der System Administrator.

Softwarearchitektur

Die Programmiersprache für die Applikation ist JAVA SE 11, wobei für die grafische Benutzeroberfläche JAVA FX 11 verwendet wird. Zur Verwaltung der Datenbank wird \_\_\_\_\_\_ verwendet.

Im Rahmen der JavaFX-Konzeption wird das MVC-Modell, auch “Model, View, Control”-Modell genannt, verwendet. Dabei wird die Programmstruktur im “Model” definiert, welches dann die Ansicht bzw. View informiert, die dann den Controller über Benutzereingabe informiert. Der Controller benachrichtigt und manipuliert im Anschluss das Model.



