

IHK-Essen
MEO FI Anw. 1 (AP T2V1)
Abschlussprüfung Sommer 2022

Fachinformatiker/Fachinformatikerin (VO 2020)
Fachrichtung:Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Prototyp zur Überwachung der Luftqualität

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Abgabedatum: Essen, den 04.05.2022

Prüfungsbewerber:

Norbert Heimsath
Mergelstraße 25
45478 Mülheim an der Ruhr
norbert@heimsath.org
0178 341 7997

Ausbildungsbetrieb:

IT-Akademie Dr. Heuer GmbH
Konrad-Zuse-Straße 2b
44801 Bochum
office@drheuer.de
0234 33855860

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Verzeichnisse

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Projektumfeld.....	1
1.2	Projektziel.....	1
1.3	Projektbegründung.....	1
1.4	Projektschnittstellen.....	1
1.5	Projektabgrenzung.....	1
2	Projektplanung.....	2
2.1	Projektphasen.....	2
2.2	Abweichungen vom Projektantrag.....	2
2.3	Ressourcenplanung.....	2
2.4	Entwicklungsprozess.....	3
3	Analysephase.....	4
3.1	Ist-Analyse.....	4
3.1.1	Wo sollten die Sensoren platziert werdend?.....	4
3.1.2	Welche Grenzwerte sind sinnvoll?.....	4
3.1.3	Wieviele Messpunkte werden benötigt?.....	4
3.2	Wirtschaftlichkeitsanalyse.....	5
3.2.1	Kommerzielle Sensor Systeme.....	5
3.2.2	Eigenbau System.....	6
3.2.3	Fazit.....	6
3.3	Anwendungsfälle.....	7
3.4	Qualitätsanforderungen.....	7
3.5	Lastenheft/Fachkonzept.....	7
4	Entwurfsphase.....	7
4.1	Zielplattform.....	7
4.2	Architekturdesign.....	7
4.3	Entwurf der Benutzeroberfläche.....	8
4.4	Datenmodell.....	8
4.5	Geschäftslogik.....	8
4.6	Maßnahmen zur Qualitätssicherung.....	9
4.7	Pflichtenheft/Datenverarbeitungskonzept.....	9
5	Implementierungsphase.....	9
5.1	Implementierung der Datenstrukturen.....	9
5.2	Implementierung der Benutzeroberfläche.....	9
5.3	Implementierung der Geschäftslogik.....	9

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Verzeichnisse

6	Abnahmephase.....	10
7	Einführungsphase.....	10
8	Dokumentation.....	10
9	Fazit.....	10
9.1	Soll-/Ist-Vergleich.....	10
9.2	Lessons Learned.....	11
9.3	Ausblick.....	11
	Eidesstattliche Erklärung.....	13
	Anhang.....	i
A1	Spiralmodell nach Boehm.....	i
A2	Sensor Prototypen Eigenbau.....	i
A3	Kommerzielle Sensoren.....	ii
A4	Anwendungsfall Diagramm.....	iv
A5	Detaillierte Zeitplanung.....	v
A6	Lastenheft (Auszug).....	vi
A7	Use-Case-Diagramm.....	vii
A8	Pflichtenheft (Auszug).....	vii
A9	Datenbankmodell.....	x
A10	Ereignisgesteuerte Prozesskette.....	xi
A11	Oberflächenentwürfe.....	xii
A12	Screenshots der Anwendung.....	xiii
A13	Entwicklerdokumentation (Auszug).....	xv
A14	Testfall und sein Aufruf auf der Konsole.....	xvi
A15	Klasse: ComparedNaturalModuleInformation.....	xvii
A16	Klassendiagramm.....	xix
A17	Lastenheft.....	xx

Abbildungsverzeichnis

Abbildung 1:	Verkleinertes Mockup der Applikation.....	11
Abbildung 2:	Anwendungsfall Diagramm Sensor Webapplikation.....	iii
Abbildung 3:	Mockup Datentabelle mit Suche.....	iv
Abbildung 4:	Mockup Dashboard.....	iv

Tabellenverzeichnis

Tabelle 1:	Grobe Zeitplanung.....	2
Tabelle 2:	Kosten Sensorensystem Kauflösung.....	5
Tabelle 3:	Kosten für Sensorensystem Eigenbau.....	6

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Verzeichnisse

Tabelle 4: Entscheidungsmatrix Max. Wert = 6.....	10
Tabelle 5: Soll-/Ist-Vergleich.....	14
Tabelle 6: Sensor Eigenbau 3 Prototypen.....	i
Tabelle 7: Kommerzielle Sensorsysteme.....	ii
Tabelle 8: Detaillierte Zeitplanung.....	x

Verzeichnis der Listings

Testklasse.....	xiii
Klasse ComparedNaturalModuleInformation.....	xiii

Abkürzungsverzeichnis

API	Application Programming Interface
CSV	Comma Separated Values
EPK	Ereignisgesteuerte Prozesskette
ERM	Entity Relationship Model
GUI	Graphical User Interface
HTML	Hypertext Markup Language
LAN	Local Area Network
MVC	Model View Controller
PHP	PHP Hypertext Preprocessor
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SVN	Subversion
VLAN	Virtuelles LAN
XML	Extensible Markup Language

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Einleitung**1 Einleitung**

Die folgende Projektdokumentation schildert den Ablauf des IHK-Abschlussprojektes, welches der Autor im Rahmen ihrer Weiterbildung/Externenprüfung zum Fachinformatikerin Fachrichtung Anwendungsentwicklung durchgeführt hat.

1.1 Projektumfeld

Die Dr. Heuer IT-Akademie ist eine Schulungseinrichtung für IT-Berufe mit ca. 160 Schülern in 12 Schulungsräumen an 2 Standorten. Die IT-Akademie bietet seit 2005 an Standort Bochum und seit 2013 am Standort Düsseldorf, Umschulungen und Weiterbildungen im IT-Bereich an. Die Dr. Heuer GmbH ist auch gleichzeitig Auftraggeber da diese Arbeit als Weiterbildung/Externenprüfung durchgeführt wird.

1.2 Projektziel

Spätestens seit Corona hat sich herausgestellt das eine Überwachung der Raumluftqualität in Schulungsräumen und Büros angebracht ist, auch wäre es aus Energiespargründen durchaus sinnvoll, diese Überwachung möglicherweise auf andere Parameter wie Temperatur auszuweiten. Das Ziel ist dementsprechend der Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen. Das Kernstück soll hierbei eine Webanwendung darstellen, die eine Verwaltung des WLAN-basierten Sensornetzes und der zugehörigen Messdaten anbietet.

1.3 Projektbegründung

Derzeit finden die Messungen stichprobenartig und manuell statt, das ist zwar grundsätzlich funktional, aber es erfordert regelmäßigen Personalaufwand und deckt auch nicht wirklich den Tagesverlauf ab. Eine gekaufte Lösung ist nach ersten Recherchen sehr kostenintensiv, so das eine eigene Lösung praktikabel erscheint.

1.4 Projektschnittstellen

Grundsätzlich sind folgende Schnittstellen zu beachten

- Die als Auftraggeber genehmigt das Projekt und stellt die Mittel zur Verfügung und nimmt die Anwendung ab.
- Einige privilegierte Mitarbeiter der IT-Akademie verwalten das Sensornetz und werden benachrichtigt wenn Sollwerte deutlich abweichen.
- Die Dozenten der Dr. Heuer GmbH werden vom System benachrichtigt wenn Sensorwerte nicht im Soll Bereich liegen
- Die Hauptanwendung empfängt Daten von den einzelnen Raumsensoren und speichert diese in einer Datenbank

1.5 Projektabgrenzung

Folgende Punkte sind nicht Teil des Projektes. Da sie völlig den Rahmen von 80 Stunden sprengen würden.

- Die weitere detaillierte Absicherung des Webserver.
- Zusammenbau weiterer Sensoren außer der 3 Testgeräte.
- Einrichten eines vom normalen Betrieb der Akademie abgegrenzten WLAN
- Konfiguration des Routers (DHCP und dergleichen) und der Switches zu aufspannen eines VLAN
- Zur Verfügung stellen von Webhooks für MS Teams, SMS Portalen und SMTP Server zum Mailversand
- Beschaffung der Hardware obliegt der Fachabteilung.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Einleitung

2 Projektplanung

2.1 Projektphasen

Das Projekt musste im Zeitraum vom 15.03.2022 – 05.05.2022 stattfinden, einem Zeitraum der sich für Externe Prüflinge mitten in der Vorbereitung der Prüfungen für Teil 1 (30.03.) und Teil 2 (04.05). Damit musste das Projekt immer um die Vorbereitungskurse herum ausgeführt werden oder Vorbereitungskurse mussten ausfallen. Alles in allem eine sehr schwierige Situation.

Die generelle Planung des Projektes und seiner Schritte wird konventionell durchgeführt, die Implementierung erfolgt im **Spiralmodell**. Dadurch ist es möglich das im Ablauf weitere Punkte hinzukommen, die diese Planung deutlich verändern. Des weiteren werden die Phasen auf die einzelnen Iterationen aufgeteilt und nicht stur nacheinander abgehandelt. Weitere Details siehe 2.4 Entwicklungsprozess @todo [Link here](#)

Tabelle 1 zeigt die grobe Zeitplanung.

Eine detailliertere Zeitplanung ist in Tabelle 8 in Anhang A11 zu sehen. @todo

Projektphase	Geplante Zeit
Analysephase	12 h
Entwurfsphase	12 h
Implementierungsphase	37 h
Abnahme und Deployment	3 h
Dokumentation	13 h
Pufferzeit	3 h
Summe	80 h

Tabelle 1: Grobe Zeitplanung

2.2 Abweichungen vom Projektantrag

Die Pufferzeit ist separat ausgewiesen, da sie nicht nur der Implementierung dient sondern generell zur Verfügung steht.

2.3 Ressourcenplanung

Alle genutzten Ressourcen sind im Anhang @todo A.2: Verwendete Ressourcen auf S. ii zusammengefasst. Dies beinhaltet sowohl Hard- und Software -ressourcen als auch das benötigte Personal. Bei der Auswahl der verwendeten Software wurde darauf geachtet, dass diese kostenfrei (z.B. als Open Source) zur Verfügung steht. Damit sollen die anfallende Projektkosten möglichst gering gehalten werden.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Projektplanung**2.4 Entwicklungsprozess**

Der Ablauf des Entwicklungsprozesses war durch die Dr. Heuer GmbH vorgegeben. Die generelle Planung des Projektes und seiner Teilkomponenten sollte konventionell durchgeführt werden und die Implementierung sollte mit Hilfe des **Spiralmodells(nach Boehm)** stattfinden. In der Planung sollten als einzelne Schritte identifiziert und spezifiziert werden, wobei die Reihenfolge der Ausführung und weitere Details in der Implementierung Schritt für Schritt sich in den Iterationen der Spirale ergeben.

Die erste Schwierigkeit die sich hierbei ergab, war das es mehrere Definitionen des Spiralmodells gibt. Vor allem überraschend war, das die verbreitete deutsche Fassung, die auch in die Wikipedia Einzug gefunden hat leider nicht der original Definition von Boehm entspricht. So beginnt sie mit dem 2. Quadranten und verzichtet auf eine Definition der Ziele für die Erste Iteration.

Aus diesem Grunde habe ich die Originalgrafik noch einmal in Anhang A1 Spiralmodell nach Boehm Seite i eingefügt

Das Spiralmodell ist ein iteratives Vorgehensmodell in dem alle Entwicklungsphasen mehrfach spiralförmig durchlaufen werden. Die Phasen sind:

1. Beschreiben der Rahmenbedingungen, festlegen der Ziele und Alternativen.
2. Risikoanalyse, Evaluierung der Alternativen, beseitigen der Risiken such Analysen, Simulationen oder Prototyping.
3. Zwischenprodukt Realisieren und Überprüfen.
4. Analyse des jetzigen und Planung des nächsten Zyklus.

Die generelle vor weggehende Planung war mir in diesem Projekt trotzdem wichtig damit ich einen groben Überblick hatte wohin es überhaupt gehen soll. Da es sich bei dem Projekt um einen Prototypen handelt war es jedoch von Vorteil einen fast schon agilen Prozess für die Ausführung zu nutzen, da Probleme und Abweichungen absolut zu erwarten waren.

Die kontinuierliche Überprüfung macht abschließende Tests nicht überflüssig, aber reduziert deren Aufwand drastisch. Auch Probleme mit dem User Interface oder zusätzlich Wünsche können direkt einfließen.

In der Implementierung werden Schritte 1-4 immer wieder durchgegangen und die Zwischenergebnisse festgehalten.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Analysephase**3 Analysephase****3.1 Ist-Analyse**

Zurzeit ist überhaupt kein Sensornetz vorhanden. Es werden einfach nur periodische Messungen mit Handgeräten durchgeführt. Aus den Messwerten werden unter Berücksichtigung der Raumgröße und aktueller Belegung mit Hilfe von statistischen Verfahren, die Zeit Intervalle festgelegt, die für eine ausreichende Belüftung notwendig sind. Dieses Verfahren ist zwar grundsätzlich funktional, aber es erfordert regelmäßigen Personalaufwand für Messungen, die Übernahme der Messwerte und Berechnungen.

Vorhanden sind die Option, ein akademieweites WLAN aufzuspannen, Kabelschächte unter den Fensterbänken (LAN und Strom) und einige bodennahe Steckdosen in den Räumen. Zu beachten ist, dass die Sensoren auf eine Stromversorgung angewiesen sind und die Netzwerkschwitches nicht über Power over Ethernet (POE) verfügen.

Als zentraler Server soll ein Raspberry Pi dienen, der auch schon vorhanden, aber nicht eingerichtet ist. Alle weiteren Hardware- Komponenten müssen neu angeschafft werden.

3.1.1 Wo sollten die Sensoren platziert werdend?

Da CO₂ schwerer als Luft ist herrscht die höchste Konzentration in Bodennähe, dementsprechend macht es Sinn die Sensoren nicht zu hoch anzubringen. Knapp über dem Boden könnten die Sensoren allerdings durch aufgewirbelten Staub oder Feuchtigkeit verunreinigt werden. Dementsprechend sollte eine Anbringung irgendwo zwischen der Gesichts- und der Knie-höhe eines sitzenden Menschen eigentlich optimal sein.

3.1.2 Welche Grenzwerte sind sinnvoll?

Das Bundesamt für Umweltschutz hat in seiner Fortbildung für den Öffentlichen Gesundheitsdienst 2019 (Anforderungen an Lüftungskonzeptionen in Bildungsgebäuden – Aktuelle Empfehlungen) Empfohlen in Innenräumen mit dem Co₂ Wert unter 1000 ppm zu bleiben. Kurzfristig wären auch Werte bis 1500 ppm vertretbar, dann müsse aber unbedingt gehandelt werden. Dementsprechend wurden die Grenzwerte für diese Projekt mit 1000 ppm und 1500 ppm festgelegt.

3.1.3 Wieviele Messpunkte werden benötigt?

Die DHG besitzt an beiden Standorten zusammen insgesamt 12 Schulungsräume und 6 Büros. Pro Schulungsraum werden 2 Sensoren benötigt, pro Büro nur einer. Da in allen Gebäuden Betondecken installiert benötigen wir bei den Kommerziellen Systemen mindestens 4 Basisstationen(2 pro Etage), da diese Systeme alle ein eigenes WLAN aufspannen um mit Ihren Sensoren zu kommunizieren. Das heißt bei den Kommerziellen Systemen benötigen wir 30 Sensoren und mindestens 4 Basisstationen. Da unser Eigenbau normales WLAN nutzt benötigen wir dort nur die 30 Sensoren.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Analysephase**3.2 Wirtschaftlichkeitsanalyse**

Zuerst habe ich mich auf die Suche nach käuflichen Raumluft Kontrollsystemen umgeschaut. Danach wurden die Prototypen für die Testsensoren festgelegt und zum Schluss ein Vergleich durchgeführt. Die Kosten für geleistete Arbeitsstunden sind von der DHG mit 43,29 € / Stunde angegeben worden.

3.2.1 Kommerzielle Sensor Systeme

Bei den kommerziellen Sensor Systemen war es relativ schwierig, Systeme zu finden, die zumindest im weitesten Sinne den Anforderungen des Pflichtenheftes entsprechen.

Die 3 Systeme, die ausgemacht werden konnten, habe ich in Anhang A3 Kommerzielle Sensoren Seite II zusammengefasst.

Entschieden habe ich mich für das Modell Monitoring CA-M Kohlendioxid von Clean-air-engineering, das pro Sensor mit 547,40 €, pro Basisstation mit 458,15 € und für die Management Cloud mit mindestens 199,38 € / Jahr zu Buche schlägt. Die Geräte müssen weiterhin immer noch an der Wand befestigt werden, die Basisstationen eingerichtet und mit den Sensoren verbunden werden. Auch muss die Firewall dazu eingerichtet werden, den Datenverkehr zwischen den Basisstationen und der Cloud zu erlauben. Der insgesamt von mir geschätzte Arbeitsaufwand beträgt etwa 10 Stunden. Die Kosten für die Cloud berechne ich auf 2 Jahre, das entspricht dem üblichen Garantiezeitraum für elektronische Geräte.

Tabelle 2: Kosten Sensorensystem Kauflösung

Posten	Preis	Anzahl	Gesamt
Sensor	547,40 €	30	16422,00 €
Basisstation	458,15 €	min. 4	1832,60 €
Cloud	Min. 199,38 €	2	398,76 €
Arbeitsstunden	43,29 €	10	432,29 €
Summe			19085,65 €

Vorteile:

- Kaufen, Installieren und Fertig(hoffentlich)
- Gerätegarantie und möglicherweise ein wenig Support vom Anbieter

Nachteile:

- Defekte Geräte können nicht vor Ort repariert werden. Nach Ende der Garantiefrist bleibt nur der komplette Austausch der Einzelgeräte.
- Möglicherweise gibt es nach ein paar Jahren keine Austauschgeräte mehr, da der Hersteller ein neues System benutzt. Dann muss das ganze Netz ersetzt werden.
- Wenn man alle Funktionen nutzen möchte, ist man von der Cloud abhängig und den Preis für die Vollversion (Enterprise Level) gibt es nur auf Anfrage.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Analysephase**3.2.2 Eigenbau System**

Es sollten 3 Sensor Prototypen Zusammengestellt werden werden alle mit Unterschiedlichen Ausstattungen.

Die detaillierte Übersicht finden Sie im Anhang A2 Sensor Prototypen Eigenbau Seite i

Für die Berechnung nehmen wir hier die teuerste und die günstigste Variante. Und als Basisstation dient ein Raspberry PI Minicomputer der mit Gehäuse 81,90 € gekostet hat. Da sich auch Dieser genauso wie die NodeMCU boards nahtlos in die bestehende WLAN Architektur einfügen, wird nur eine Basisstation benötigt.

Tabelle 3: Kosten für Sensorensystem Eigenbau

Posten	Preis	Anzahl	Gesamt V1	Gesamt V2
Sensor	29,18 € / 61,24 €	30	875,40 €	1837,20
Basistation	81,90 €	1	81,90 €	81,90 €
Arbeitszeit	43,29 €	80	3463,20	3463,20 €
Summe			4420,50 €	5382,30 €

Vorteile:

- Alle Sensoren können repariert werden.
- Sensortypen können erweitert und Verbessert werden.
- Auch die Kommunikationsschnittelle kann angepasst werden.
- Die Applikation kann beliebig erweitert werden
- Die Einzelteile für Sensoren werden eher günstiger als teurer.

Nachteile:

- Mehr Arbeitszeit benötigt.
- Fachkräfte werden benötigt.

3.2.3 Fazit

Es ergibt sich sofort eine Beträchtliche Preisdifferenz:

Bei Variante 3: 19085,65 € - 5382,30 € = **13703,35 €**

und

Bei Variante 1: 19085,65 € - 4420,50 € = **14665,15 €**

Das entspricht **338** bzw. **316** weiteren Entwicklerstunden die im Vergleich zur Kommerziellen Lösung noch zur Verfügung stehen.

Weitere Vorteile:

- Reparaturen an den Sensoren sind problemlos möglich also max. 23,40 € statt 547,40 € für einen neuen Sensor.
- Die Cloud Kosten fallen komplett weg.

Die Entscheidung fällt eindeutig zu Gunsten der Eigenbaulösung aus!

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Analysephase**3.3 Anwendungsfälle**

Dieses Projekt beinhaltet 3 Haupt Anwendungsfälle

1. Ein jeder einem Klassenraum zugeordneter Dozent wird gewarnt wenn der Co2-gehalt der Atemluft im Raum über die festgelegten Grenzen steigt. Des Weiteren hat er die Möglichkeit die allgemeine Übersicht (Dashboard) über alle Räume ohne Login aufzurufen.
2. Ein als Benutzer eingetragener „Manager“ wird benachrichtigt wenn der zuständige Dozent die Warnung ignoriert und der Co2 Gehalt über den 2. Grenzwert steigt. Auch kann er die Übersicht aufrufen. Weiterhin kann ein Manager sich anmelden und sämtliche Objekte (Dozenten,Sensoren, Räume...) erstellen und verwalten. Löschen kann er nicht, nur deaktivieren. Darüber hinaus kann er Daten exportieren.
3. Der Administrator der Anwendung wird nicht benachrichtigt. Er kann allerdings ebenfalls alles was der Manager kann und darüber hinaus die gesamte Anwendung verwalten, erweitern und verändern.

Um dies weiter zu verdeutlichen, habe ich Anhang A4 Anwendungsfall Diagramm Seite iii eingefügt.

3.4 Qualitätsanforderungen

Da es sich um einen Prototypen handelt sind die Qualitätsanforderungen nicht übermäßig hoch.

- Übersichtliche Darstellung der Daten.
- Datentabellen sollten sortierbar und durchsuchbar sein.
- Die Anwendung soll zuverlässig über lange Zeit funktionieren und robust sein.
- Die Optik sollte dem Corporate Design entsprechen
- Programmcode sollte unbedingt Objektorientiert sein
- Der Code sollte den Vorgaben des Frameworks entsprechen.
- In Code Kommentare werden in ausreichender Form erwartet.
- Die Sensoren werden in einem fortgesetzten Projekt noch weiterentwickelt, insofern sollen diese nur generell funktionieren und Testdaten liefern.
- Die Datenübertragung sollte wenn möglich verschlüsselt erfolgen.

3.5 Lastenheft/Fachkonzept

Das Lastenheft befindet sich aus Platzgründen in einem separaten Anhang. @todo

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Analysephase

4 Entwurfsphase**4.1 Zielplattform****4.1.1 Server**

Als Serverrechner ist ein Raspberry PI minirechner vorgegeben, dieser muss neu eingerichtet werden. Das standard Betriebssystem für diese Rechner ist Raspian ein Leichtgewichtiger Klon der Debian Linux Distribution. Im Zweifelsfalle können auch original Debian Pakete installiert werden. Folgende Features müssen installiert werden:

- Apache Webserver mit verschlüsselter Datenübertragung (HTTPS)
- Ein selbstsigniertes Zertifikat für HTTPS
- Ein Datenbanksystem
- PHP
- Remote Zugriff
- X-Window System
- Arduino IDE

Es wird nur eine grundsätzliche Einrichtung erwartet, den Rest übernimmt der Fachbereich.

4.1.2 Sensoren

Für die Sensoren werden einzelne elektronische Sensor- und Anzeige- Bauteile verwendet die gemeinsam mit einem ESP8266 Mikroprozessorboard auf eine Platine gesteckt oder gelötet werden. Der ESP8266 ist günstig (ca. 4€) und wird in so gut wie allen Bastelprojekten zum Thema Sensorik verwendet. Eine Alternative wäre ein original Arduino, der aber wesentlich teurer ist (ab ca. 20€) und auch keine bessere Funktionalität bietet. Ein ESP8266 lässt sich in verschiedenen Programmiersprachen programmieren (MicroPython, C++ , LUA), leider stellte sich schon bei der Recherche für die Sensor Komponenten heraus, das eine vollständige Treiberunterstützung für alle Komponenten nur mit der Arduino IDE (C++) möglich ist.

4.2 Architekturdesign**4.2.1 Sensor Platzierung**

Zuerst musste einmal geklärt werden wo die Sensoren überhaupt platziert werden konnten damit festgelegt werden konnte ob die Datenübertragung per Lan oder Wlan stattfinden konnte.

Büros und Klassenräume in der DHG sind praktisch immer gleich aufgebaut. Eine große Stromleiste unter den Fenstern versorgt alle Arbeitsplätze mit Strom und LAN Anschlüssen. Wireless Access Points versorgen weiterhin alle Räume mit einem Akademieweiten WLAN. Die LAN Switches sind leider nicht in der Lage POE(Power over Ethernet zu liefern).

- Wie wir schon vorher festgestellt haben ist die optimale Höhe irgendwo zwischen Knie und Gesichtshöhe einer sitzenden Person. Damit fällt die Decke als Platz auch schon aus.
- Eine Positionierung am Fenster ist nicht sinnvoll, da beim Lüften sofort wieder optimale Luftqualität gegeben ist obwohl die Luft noch nicht den Rest des Raumes erreicht hat.
- An den Innenseiten der Räume befinden sich mehrere Steckdosen auf Kniehöhe, die als Stromversorgung dienen könnten allerdings keine Anschlüsse für LAN. Die Steck-

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Entwurfsphase

dosen dürfen allerdings nicht vollständig blockiert werden, da Sie vom Reinigungspersonal benötigt werden.

Daraus ergibt sich eigentlich zwangsläufig folgende Konstellation:

Für die Stromversorgung werden die Steckdosen an den Rauminnenseiten genutzt und die Datenübertragung findet über WLAN statt. Die Sensoren werden neben oder an den Steckdosen montiert und sind damit auch ungefähr in der richtigen Höhe für die Messungen.

4.2.2 Programmiersprache

Da ich seit vielen Jahren Webanwendungen mit PHP schreibe fiel die Entscheidung relativ leicht. Zumal es keine gute Idee ist eine Projektarbeit in Sprachen auszuführen mit denen man nur wenig vertraut ist. Weiterhin bin ich in PHP mit verschiedenen Frameworks vertraut. Vor allem da der Umfang des Projektes relativ groß ist, wäre es ziemlich sinnfrei alle Teilelemente vollständig selbst zu programmieren. Bleibt also noch die Auswahl eines passenden Frameworks als Grundlage.

4.2.3 Framework

Da das ganze Projekt auf einem Raspian Minirechner (RASPI) laufen soll ist die Menge die an Ressourcen zur Verfügung steht eher gering. Damit fallen Umgebungen wie Zend Framework oder Symfony schon aus da der Bedarf an Rechenleistung und Speicher eher für größere Server geeignet ist. Leider ist es in Mode gekommen solche Anforderungen öffentlich freizugeben, so das ich mich da nur auf Erfahrungswerte verlassen kann.

Bleiben hier also noch leichtgewichtige Frameworks zur Auswahl. In meinem Fall sind das Laravel und Processwire. Beide bringen einen guten Funktionsumfang mit und sind nicht all zu schwerfällig. Hier ein direkter Vergleich:

Processwire	Laravel
Processwire(PW) ist ein kleines Framework/CMS das relativ ungewöhnliche Wege geht. die Datenbank ist im Normalfall völlig gekapselt was SQL injections praktisch unmöglich macht. Grundsätzliche Dinge wie Benutzer- und Rechteverwaltung, ein Admin Panel, eine Modulschnittstelle viele weitere nützliche Kleinigkeiten machen das schnelle Entwickeln von neuen Applikationen angenehm einfach. Dazu kommt ein Hook System, das es erlaubt an unglaublich vielen Stellen der Hauptapplikation seinen eigenen Code einfach einzuhängen. Leider ist Processwire nicht optimal wenn es um die Continuous Integration geht das Updaten einer Applikation ist etwas sperrig. PW wird allerdings auch eher für kleinere Projekte genutzt.	Laravell ist ein allgemeines Framework für Webapplikationen grundsätzlich kann man damit alles machen was man will. Die Dokumentation ist gut und man findet sehr viele von Benutzern erstellte Erweiterungen. Diese Vielfalt ist gleichzeitig auch einer der haupt Nachteile. Für Alles gibt es Erweiterungen nur Interagieren die nicht immer alles völlig reibungslos miteinander. Teilweise wird der Support eingestellt und die Erweiterungen fallen aus. Teilweise werden bei neuen Laravel Versionen Dinge verändert und Module werden disfunktional. Als Entwickler ist man ständig damit beschäftigt eine Applikation auf dem Laufenden zu halten. Für unsere kleine Anwendung ist das alles eher kontraproduktiv.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Entwurfsphase**Woraus legen wir bei der Auswahl besonderen Wert?**

Da das Projekt trotz ziemlichen Umfangs in 80h nebst Planung und Dokumentation abgeschlossen sein muss legen wir natürlich besonderen Wert darauf möglichst schnell voranzukommen. Sparsame Ressourcennutzung ist ein weiterer wichtiger Punkt. Aufgrund der knappen Zeit ist es fast unmöglich bei auftretenden Problemen mit langen Recherchen eine Lösung zu finden, deswegen ist gute Dokumentation/Support sehr wichtig. Testing ist grundsätzlich nicht unwichtig. Auch die Vertrautheit mit einem System ist als Projektüberlegung wichtig, da es den ganzen Prozess beschleunigt und Probleme und die dazugehörigen Workarounds schon bekannt sind. Der Basis Funktionsumfang begrenzt die Menge an Standardlösungen die noch gefunden werden müssen. Wir wollen ja nicht ständig das Rad neu erfinden. Synchronisation und Rollout werden zwar erst wirklich wichtig wenn der Prototyp in Serie gehen sollte aber auch das sollte berücksichtigt werden. Den gesamt möglichen Funktionsumfang lasse ich hier außen vor, da die Applikation einen eng umgrenzten Einsatzbereich hat. Ein weiterer wirklich wichtiger Punkt ist die Frage wie sicher und auch wie zukunftssicher ist die Applikation, da wir ein System haben das möglicherweise selten mit aktuellen Patches versehen wird und obendrein auch nicht möchten, das beim Update dann irgendwelche Probleme auftreten. Und zu guter Letzt ist Sicherheit natürlich immer ein wichtiges Thema.

Eigenschaft	Gewichtung	Processwire		Laravel	
Ressourcenbedarf	4	6	24	4	16
Rapid Development	6	5	30	4	24
Basis Funktionalität	3	5	15	2	6
Testing	2	2	4	5	10
Vertrautheit	3	4	12	3	9
Sicherheit	5	5	25	3	15
Update/Wartung	4	5	20	4	16
Dokumentation/ Support	4	5	20	4	16
Sync und Rollout	2	3	6	5	10
Nutzwert			156		122

Tabelle 4: Entscheidungsmatrix Max. Wert = 6

4.2.4 Fazit

Trotz der Mächtigkeit von Laravel halte ich Processwire für diese Projekt für das besser geeignete System.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

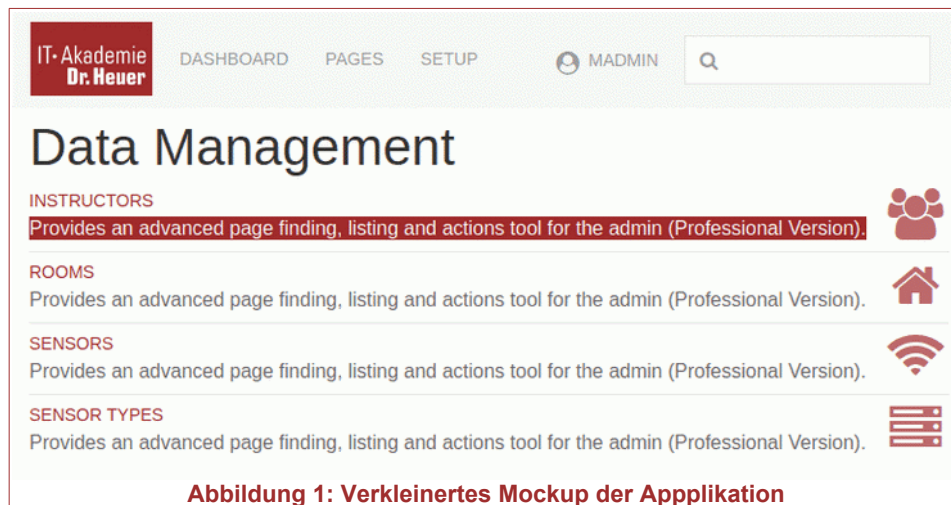
Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Entwurfsphase**4.2.5 Datenbanksystem**

Da Processwire in Zusammenarbeit mit MariaDB oder MySQL die besten Resultate liefert wähle ich einfach aus persönlicher Vorliebe MariaDB aus. Zumal das auch die unter Debian/Raspian bevorzugte DB ist und sich damit auf den geplanten Minirechner gut einrichten lässt.

4.3 Entwurf der Benutzeroberfläche

Processwire kommt von Hause aus mit einer auf dem CSS Framework Uikit basierenden Oberfläche die sich mit Hilfe eines Modular integrierbaren Compilers für die Stylesheet-Sprache „Less“ vollständig konfigurieren und anpassen lässt.



Fürs Mockup habe ich eine Test Applikation farblich etwas angepasst und mit dem Grafikprogramm GIMP noch ein wenig nachbearbeitet. Weitere Mockups finden sich in Anhang A5 Mockups der Anwendung Seite iv

4.4 Datenmodell

- Entwurf/Beschreibung der Datenstrukturen (z.B. ERM und/oder Tabellenmodell, XML-Schemas) mit kurzer Beschreibung der wichtigsten (!) verwendeten Entitäten.

Beispiel

In Anhang Fehler: Verweis nicht gefunden wird ein ERM dargestellt, welches lediglich Entitäten, Relationen und die dazugehörigen Kardinalitäten enthält.

4.5 Geschäftslogik

- Modellierung und Beschreibung der wichtigsten (!) Bereiche der Geschäftslogik (z.B. mit Komponenten-, Klassen-, Sequenz-, Datenflussdiagramm, Programmablaufplan, Struktogramm, EPK).
- Wie wird die erstellte Anwendung in den Arbeitsfluss des Unternehmens integriert?

Beispiel

Ein Klassendiagramm, welches die Klassen der Anwendung und deren Beziehungen untereinander darstellt, kann im Anhang Fehler: Verweis nicht gefunden eingesehen werden.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Entwurfsphase

Die EPK in Anhang Fehler: Verweis nicht gefunden zeigt den grundsätzlichen Ablauf beim Einlesen eines Moduls.

4.6 Maßnahmen zur Qualitätssicherung

- Welche Maßnahmen werden ergriffen, um die Qualität des Projektergebnisses (siehe Kapitel 3.4) zu sichern (z.B. automatische Tests, Anwendertests)?
- Ggfs. Definition von Testfällen und deren Durchführung (durch Programme/Benutzer).

4.7 Pflichtenheft/Datenverarbeitungskonzept

- Auszüge aus dem Pflichtenheft/Datenverarbeitungskonzept, wenn es im Rahmen des Projekts erstellt wurde.

Beispiel

Ein Beispiel für das auf dem Lastenheft (siehe Kapitel 3.5) aufbauende Pflichtenheft ist im Anhang Fehler: Verweis nicht gefunden zu finden.

5 Implementierungsphase

@todo KANN ICH DAS SO MACHEN ????

Da die Implementierung im Spiralsystem durchgeführt wurde werde ich die Iterationen hier Quadrantenweise durchgehen. Noch einmal zur Erinnerung:

1. Festlegen der Ziele und Alternativen
2. Beurteilung von alternativen und Risikoanalyse
3. Entwicklung und Test
4. Analyse und Planung des nächsten Zyklus

5.1 Implementierung des Servers (soll 3h / ist 5h)

1. Raspberry, Raspbian OS installieren mit Apache Webserver mit HTTPs, MariaDB, PHP 8, Processwire CMS, Arduino IDE , X-Windows, VNC Server, SSH, VIM als lokale Entwicklungsumgebung, Setup mit Monitor Maus und Tastatur. (Geplant 3h)
2. Bei Unverträglichkeiten oder Fehlenden Softwarepaketen ist mit zusätzlichem Zeitaufwand zu rechnen. Deswegen habe ich mich für ein Original Raspbian OS entschieden da Debian meiner Erfahrung nach bei der Installation mehr Probleme bereitet. Hardware Probleme wären auch möglich aber es steht Reservehardware zur Verfügung
3. Zuerst Image auf Speicherkarte schreiben und Rechner starten hier trat das erste Problem auf, nämlich das die Karte nicht mehr richtig funktionierte leider war ich da schon halb fertig. Paketinstallation verlief teilweise etwas zickig. Ich musste PHP aus Fremddquellen installiert werden und einige Module(z.b. BcMath) mussten händisch nachinstalliert werden. Die Qualität von Raspbian OS im generellen entspricht nicht der Stabilität von Debian, alles verlief sehr mühsam.
Als abschließender Test wurde Processwire installiert, der Selbsttest bei der Installation und eine manuelle Überprüfung verliefen Problemlos.
4. Habe Mich bei dem Zeitaufwand deutlich überschätzt so das hier schon 5h anfielen. Leider machte mein Internetprovider wirklich Probleme mit dem heimischen Serverbetrieb so das Ich mich entschlossen habe im nächsten Schritt auf einem Öffentlichen Webserver auszuweichen.
Im nächsten Schritt wird die Applikation erneut installiert, ans Corporate Design angepasst , alle Notwendigen Module @todo Anhang PW module ? werden eingespielt

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Implementierungsphase

und konfiguriert die vorgesehenen Datenstrukturen in PW angelegt und die Formulare und Datentabellen konfiguriert.

5.2 Grundanwendung, Benutzeroberfläche

- 1.
2. Ggfs. Beschreibung des Corporate Designs und dessen Umsetzung in der Anwendung.

Beispiel

Screenshots der Anwendung in der Entwicklungsphase mit Dummy-Daten befinden sich im Anhang Fehler: Verweis nicht gefunden.

5.3 Implementierung der Geschäftslogik

- Beschreibung des Vorgehens bei der Umsetzung/Programmierung der entworfenen Anwendung.
- Ggfs. interessante Funktionen/Algorithmen im Detail vorstellen, verwendete Entwurfsmuster zeigen.
- Quelltextbeispiele zeigen.
- Hinweis: Es wird nicht ein lauffähiges Programm bewertet, sondern die Projektdurchführung. Dennoch würde ich immer Quelltextausschnitte zeigen, da sonst Zweifel an der tatsächlichen Leistung des Prüflings aufkommen können.

Beispiel

Die Klasse `ComparedNaturalModuleInformation` findet sich im Anhang Fehler: Verweis nicht gefunden.

6 Abnahmephase

- Welche Tests (z.B. Unit-, Integrations-, Systemtests) wurden durchgeführt und welche Ergebnisse haben sie geliefert (z.B. Logs von Unit Tests, Testprotokolle der Anwender)?
- Wurde die Anwendung offiziell abgenommen?

Beispiel

Ein Auszug eines Unit Tests befindet sich im Anhang Fehler: Verweis nicht gefunden. Dort ist auch der Aufruf des Tests auf der Konsole des Webserverns zu sehen.

7 Einführungsphase

- Welche Schritte waren zum Deployment der Anwendung nötig und wie wurden sie durchgeführt (automatisiert/manuell)?
- Wurden Ggfs. Altdaten migriert und wenn ja, wie?
- Wurden Benutzerschulungen durchgeführt und wenn ja, Wie wurden sie vorbereitet?

8 Dokumentation

- Wie wurde die Anwendung für die Benutzer/Administratoren/Entwickler dokumentiert (z.B. Benutzerhandbuch, API-Dokumentation)?
- Hinweis: Je nach Zielgruppe gelten bestimmte Anforderungen für die Dokumentation (z.B. keine IT-Fachbegriffe in einer Anwenderdokumentation verwenden, aber auf jeden Fall in einer Dokumentation für den IT-Bereich).

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Dokumentation**Beispiel**

Ein Ausschnitt aus der erstellten Benutzerdokumentation befindet sich im Anhang Fehler: Verweis nicht gefunden.

Die Entwicklerdokumentation wurde mittels *PHPDoc* automatisch generiert. Ein beispielhafter Auszug aus der Dokumentation einer Klasse findet sich im Anhang Fehler: Verweis nicht gefunden.

9 Fazit**9.1 Soll-/Ist-Vergleich**

- Wurde das Projektziel erreicht und wenn nein, warum nicht?
- Ist der Auftraggeber mit dem Projektergebnis zufrieden und wenn nein, warum nicht?
- Wurde die Projektplanung (Zeit, Kosten, Personal, Sachmittel) eingehalten oder haben sich Abweichungen ergeben und wenn ja, warum?
- Hinweis: Die Projektplanung muss nicht strikt eingehalten werden. Vielmehr sind Abweichungen sogar als normal anzusehen. Sie müssen nur vernünftig begründet werden (z.B. durch Änderungen an den Anforderungen, unter-/überschätzter Aufwand).

Beispiel (verkürzt)

Wie in Tabelle 5 zu erkennen ist, konnte die Zeitplanung bis auf wenige Ausnahmen eingehalten werden.

Phase	Geplant	Tatsächlich	Differenz
Analyse	9 h	10 h	+1 h
Entwurf	20 h	20 h	
Implementierung	30 h	27 h	-3 h
Abnahme	1 h	1 h	
Einführung	1 h	1 h	
Dokumentation	9 h	11 h	+2 h
Gesamt	70 h	70 h	

Tabelle 5: Soll-/Ist-Vergleich

9.2 Lessons Learned

- Was hat der Prüfling bei der Durchführung des Projekts gelernt (z.B. Zeitplanung, Vorteile der eingesetzten Frameworks, Änderungen der Anforderungen)?

9.3 Ausblick

- Wie wird sich das Projekt in Zukunft weiterentwickeln (z.B. geplante Erweiterungen)?

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Fazit

Literaturverzeichnis

ISO9126-1: ISO/IEC 9126-1, Software-Engineering – Qualität von Software-Produkten – Teil
1: Qualitätsmodell, 2001

Literaturverzeichnis

ISO9126-1: ISO/IEC 9126-1, Software-Engineering – Qualität von Software-Produkten – Teil
1: Qualitätsmodell, 2001

Empfehlungen des Bundesumweltamtes:

<https://www.bfr.bund.de/cm/343/anforderungen-an-lueftungskonzeptionen-in-bildungsgebaeuden-aktuelle-empfehlungen.pdf>

@todo Einträge ins literaturverzeichniss

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Eidesstattliche Erklärung

Eidesstattliche Erklärung

Ich, Norbert Heimsath, versichere hiermit, dass ich meine Dokumentation zur betrieblichen Projektarbeit mit dem Thema

Prototyp zur Überwachung der Luftqualität – Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Essen, den 04.05.2022

NORBERT HEIMSATH

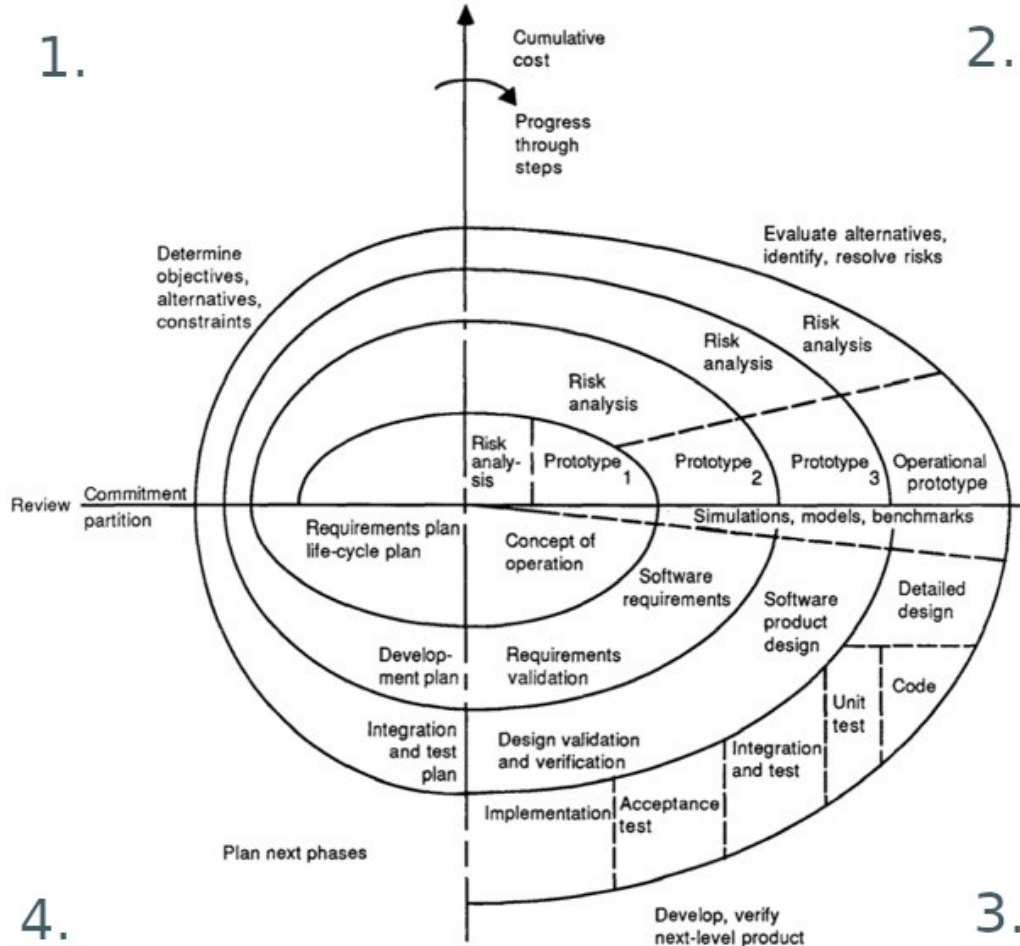
PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Anhang

Anhang

A1 Spiralmodell nach Boehm



A2 Sensor Prototypen Eigenbau

Übersicht über die 3 geplanten Prototypen mit Preisen und Komponenten. Es Sollte 3 verschiedene Preisstufen berücksichtigt werden. Bei der Recherche hatte ich überdies den Eindruck das auch die teuren Kommerziellen Systeme zur Co2 Messung ebenfalls eins der beiden Sensormodelle benutzen. Zu diesem Schluss bin ich gekommen weil sich die Eichverfahren, Einbrennzeiten und Benutzungshinweise zu 100% mit denen der Dokumentationen der einzelnen Sensoren decken.

Tabelle 6: Sensor Eigenbau 3 Prototypen

System	Preis	Variante 1	Variante 2	Variante 3
NodeMCU Mainboard	4,66 €	4,66 €	4,66 €	4,66 €
Platine	1,60 €	1,60 €	1,60 €	1,60 €
revolt USB Netzteile & Smartphone-Halterung	11,50 €	11,50 €	11,50 €	11,50 €
Gehäuse mit Klarsicht-	4,50 €	4,50 €	4,50 €	4,50 €

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Anhang

deckel				
Diverse Kleinteile	2,50 €	2,50 €	2,50 €	2,50 €
MH-Z19C Co2 Sensor	23,40 €		23,40 €	23,40 €
MQ-135 CO2/Luftqualität Sensor	2,66 €	2,66 €		
HT22 Temperatur- Feuchtigkeitssensor	7,99 €			7,99 €
LED- Ampel	1,83 €		1,83 €	
Beeper	1,76 €	1,76 €	1,76 €	1,76 €
LCD Anzeige	3,33 €			3,33 €
Summe		29,18 €	51,75 €	61,24 €

A3 Kommerzielle Sensoren

Generell kranken die kommerziellen Systeme immer wieder unter unklaren Spezifikationen. API's werden nicht offengelegt oder stehen erst zur Verfügung wenn man gekauft hat. Features werden nur vage beschrieben und Kosten für Clouddienste müssen separat recherchiert und natürlich auch bezahlt werden.

Übertragungswege für Warnungen außerhalb von Email und SMS sind nicht vorgesehen, meist werden nicht alle SMS Portale unterstützt. Oft werden Produktserien einfach eingestellt so dass man bei teilweisem Ausfall ein komplett neues System benötigt.

Wenn die Verfügbarkeit besser wäre wäre die Entscheidung auf den BMC gefallen, leider ist diese so schlecht das ich mich für den **Clean-air-engineering Monitoring CA-M** entschieden habe.

Tabelle 7: Kommerzielle Sensorsysteme

	Virtenio Preon	BMC	Clean-air-engineering
Name	Cube Co2 Ampel	RTR-576 Funk Datenlogger	Monitoring CA-M Kohlendioxid
Kosten	179,00€ / Stück	528,00 € / Stück	547,40 € / Stüc
Weitere Kosten	Wandhalterung 25,00€ / Stück	Netzteil 22,80 € / Stück Wandhalterung 27,28 € / Stück	
Cloud	Cloud für 24 Monate 45,60 € / Pro Gerät ?	Derzeit kostenlos.	Basis Version 199,38 € / Jahr Enterprise Version nur auf Anfrage.
Link	Virtenio.com	BMC.de	Clean-air-engineering.de
Bemerk.	Da Kalibrierung notwendig	Schlechte Verfügbar-	Sensor Takte von kürzer

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Anhang

	handelt es sich wahrscheinlich um den billigsten am Markt verfügbaren Sensor, Kosten ca. 3,50€. Alarmwerte können nicht eingestellt werden. Reparatur ist nicht vorgesehen.	keit in Europa. Reparatur ist nicht vorgesehen.	als 2 Stunden erfordern Enterprise Version der Cloud. Reparatur ist nicht vorgesehen.
Eignung	Nur bedingt geeignet, da Features der Cloud unklar und ohne Kommerziellen Cloud Anbieter keine zentrale Datenerfassung. Kalibrierung ist notwendig also ist in regelmäßigen Abständen mit manuelle Wartungsarbeiten zu rechnen.	In Europa ist es im Moment wirklich schwierig die Basisstationen zu erhalten damit ist dieses System für den Einsatz hier nur bedingt brauchbar.	Generell gut , nur sehr teuer. Leider haben die Sensoren keine Ampel oder Anzeige. Für lokale Datenerfassung ist immer die Enterprise version der Cloud Software erforderlich, diese kann dann aber auch lokal installiert werden. Preis leider nur auf Anfrage.
Basis	keine	ca. 350 € (Dollarkurs)	458,15 €

A4 Anwendungsfall Diagramm

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

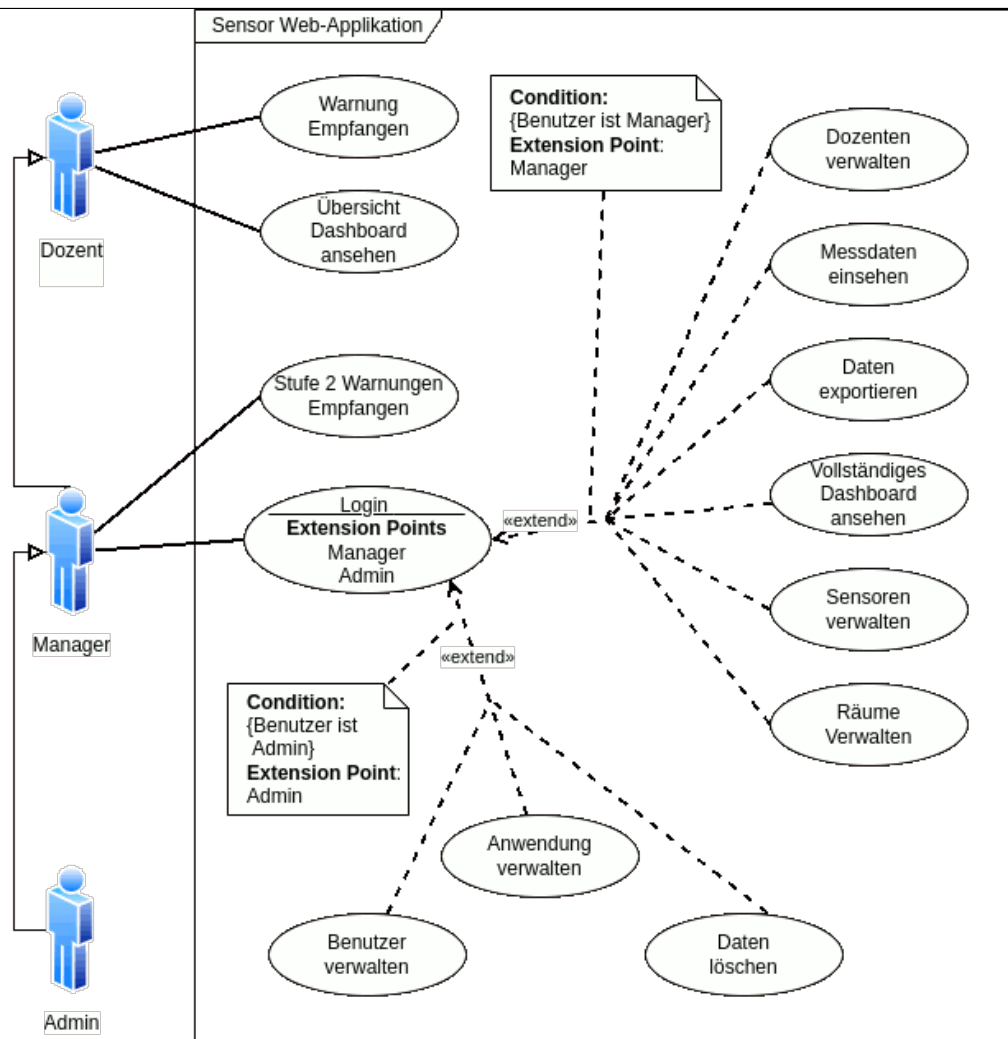
Anhang

Abbildung 2: Anwendungsfall Diagramm Sensor Webapplikation

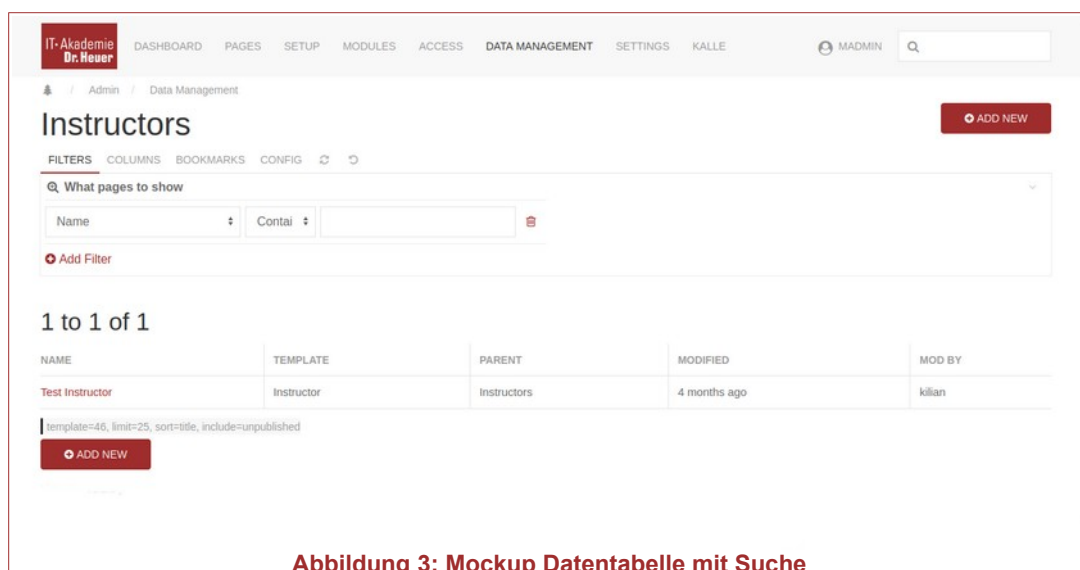
A5 Mockups der Anwendung

Abbildung 3: Mockup Datentabelle mit Suche

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Anhang

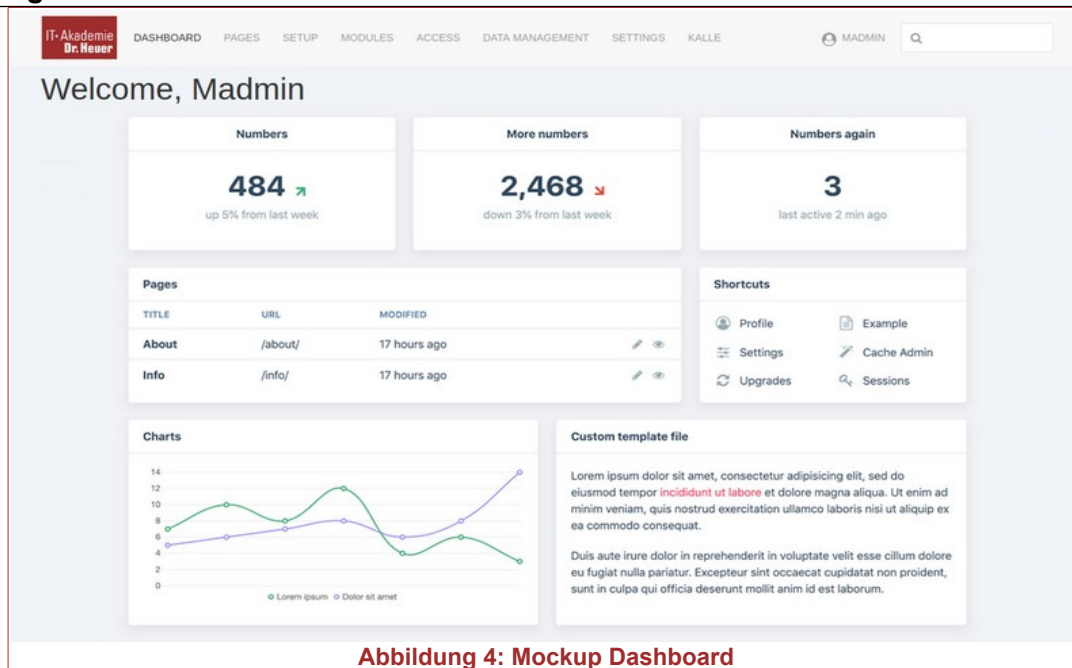
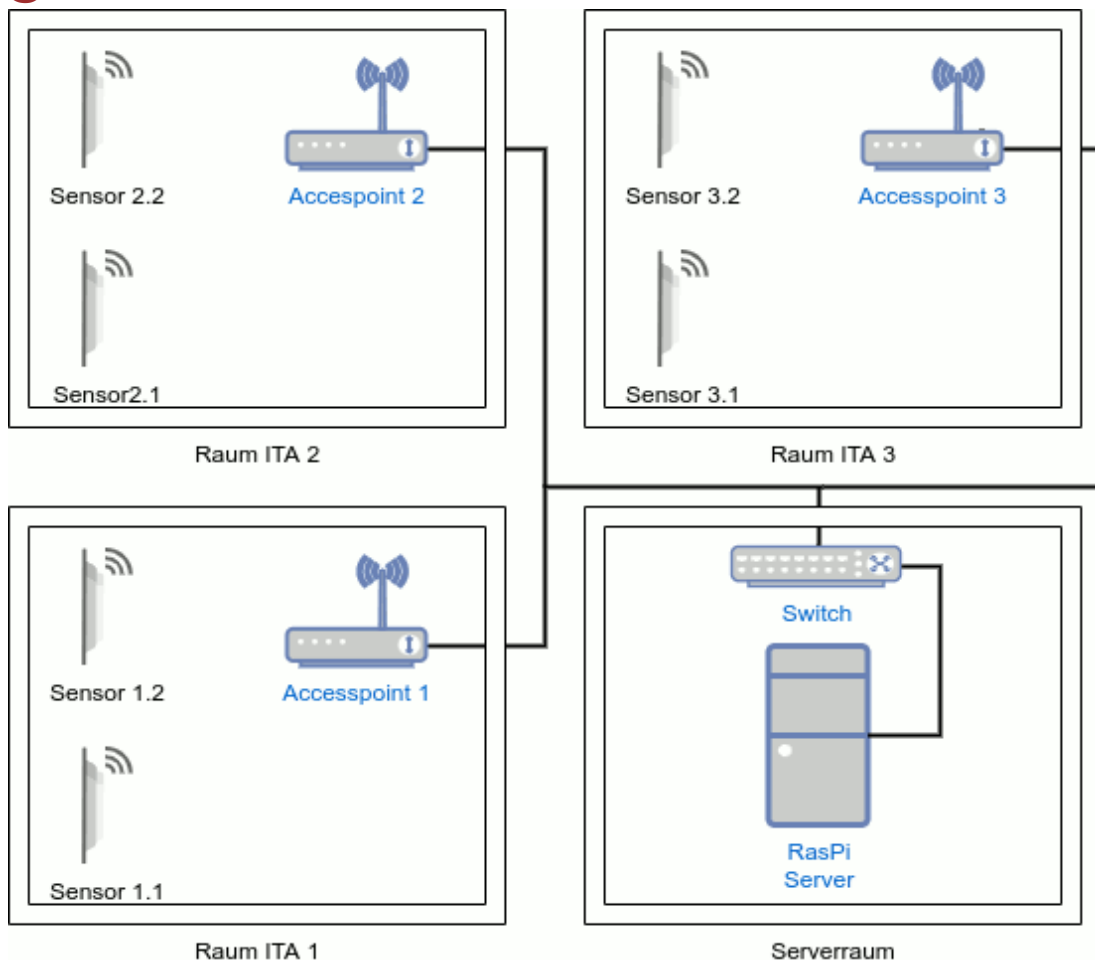


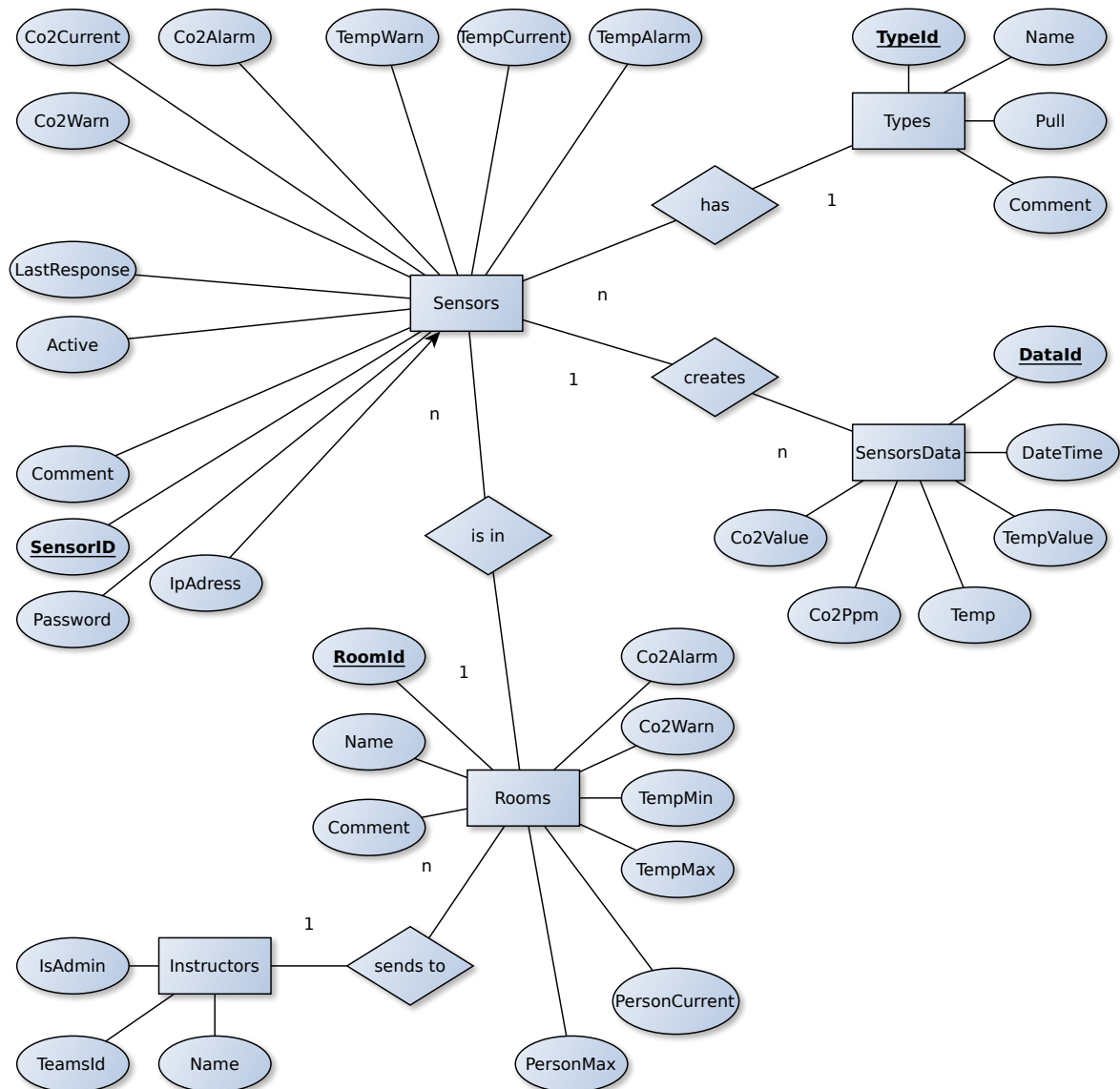
Abbildung 4: Mockup Dashboard

A6 @todo Wie nenne ich das hier



PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

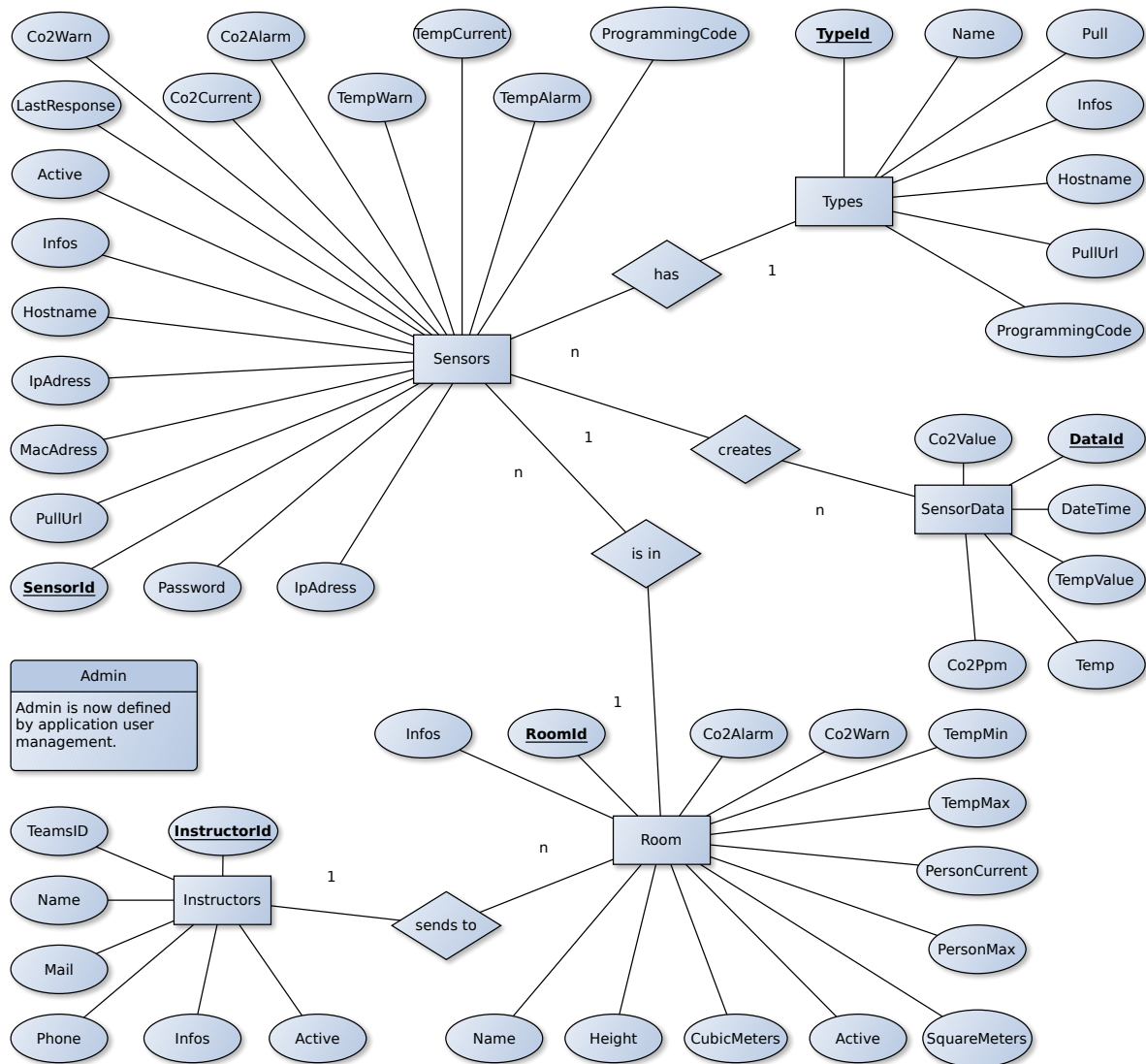
Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Anhang**A7 Datenbank ER Diagramm Entwurf****A8 Datenbank ER Diagramm nach Änderungen**

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Anhang

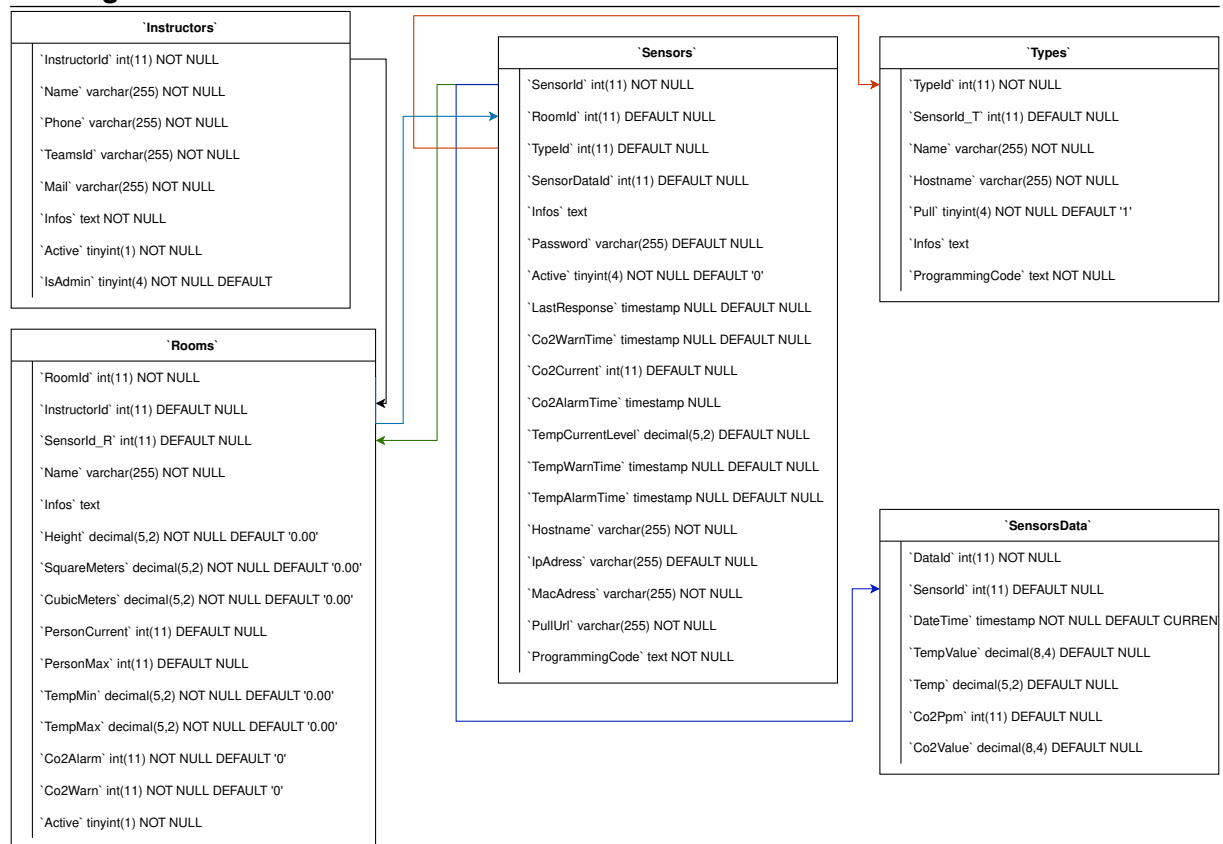


A9 Symbolische Datenbank Struktur nach Änderungen

Wie schon erwähnt speichert Processwire alle Daten sozusagen als Objekt Blackbox. Diese Datenstruktur zeigt aber wie die Daten Abgebildet werden.

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität in den Schulungsräumen der IT-Akademie.

Anhang



A10 !!!!Ab hier nur noch Beispielzeugs aus der Vorlage!!!!

A11 Detaillierte Zeitplanung

Analysephase		9 h
1. Analyse des Ist-Zustands		3 h
1.1. Fachgespräch mit der EDV-Abteilung		1 h
1.2. Prozessanalyse		2 h
2. „Make or buy“-Entscheidung und Wirtschaftlichkeitsanalyse		1 h
3. Erstellen eines Use-Case-Diagramms		2 h
4. Erstellen des Lastenhefts mit der EDV-Abteilung		3 h
Entwurfsphase		20 h
1. Prozessentwurf		3 h
2. Datenbankentwurf		3 h
2.1. ER-Modell erstellen		2 h

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Anhang

2.2. Konkretes Tabellenmodell erstellen	1 h
3. Erstellen von Datenverarbeitungskonzepten	4 h
3.1. Verarbeitung der CSV-Daten	1 h
3.2. Verarbeitung der SVN-Daten	1 h
3.3. Verarbeitung der Sourcen der Programme	2 h
4. Benutzeroberflächen entwerfen und abstimmen	2 h
5. Erstellen eines UML-Komponentendiagramms der Anwendung	4 h
6. Erstellen des Pflichtenhefts	4 h
Implementierungsphase	30 h
1. Anlegen der Datenbank	1 h
2. Umsetzung der HTML-Oberflächen und Stylesheets	5 h
3. Programmierung der PHP-Module für die Funktionen	23 h
3.1. Import der Modulinformationen aus CSV-Dateien	2 h
3.2. Parsen der Modulquelltexte	3 h
3.3. Import der SVN-Daten	2 h
3.4. Vergleichen zweier Umgebungen	4 h
3.5. Abrufen der von einem zu wählenden Benutzer geänderten Module	3 h
3.6. Erstellen einer Liste der Module unter unterschiedlichen Aspekten	5 h
3.7. Anzeigen einer Liste mit den Modulen und geparsten Metadaten	3 h
3.8. Erstellen einer Übersichtsseite für ein einzelnes Modul	1 h
4. Nächtlichen Batchjob einrichten	1 h
Abnahmetest der Fachabteilung	1 h
1. Abnahmetest der Fachabteilung	1 h
Einführungsphase	1 h
1. Einführung/Benutzerschulung	1 h
Erstellen der Dokumentation	9 h
1. Erstellen der Benutzerdokumentation	2 h

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Anhang

2. Erstellen der Projektdokumentation	6 h
3. Programmdokumentation	1 h
3.1. Generierung durch <i>PHPdoc</i>	1 h
Gesamt	70 h

Tabelle 8: Detaillierte Zeitplanung

A12 Lastenheft (Auszug)

@todo Lastenheft Separat ? Oder Wie bekomme ich das hier rein ????

Es folgt ein Auszug aus dem Lastenheft mit Fokus auf die Anforderungen:

Die Anwendung muss folgende Anforderungen erfüllen.

1. Verarbeitung der Moduldaten
 - 1.1. Die Anwendung muss die von Subversion und einem externen Programm bereitgestellten Informationen (z.B. Source-Benutzer, -Datum, Hash) verarbeiten.
 - 1.2. Auslesen der Beschreibung und der Stichwörter aus dem Sourcecode.
2. Darstellung der Daten
 - 2.1. Die Anwendung muss eine Liste aller Module erzeugen inkl. Source-Benutzer und -Datum, letztem Commit-Benutzer und -Datum für alle drei Umgebungen.
 - 2.2. Verknüpfen der Module mit externen Tools wie z.B. Wiki-Einträgen zu den Modulen oder dem Sourcecode in Subversion.
 - 2.3. Die Sourcen der Umgebungen müssen verglichen und eine schnelle Übersicht zur Einhaltung des allgemeinen Entwicklungsprozesses gegeben werden.
 - 2.4. Dieser Vergleich muss auf die von einem bestimmten Benutzer bearbeiteten Module eingeschränkt werden können.
 - 2.5. Die Anwendung muss in dieser Liste auch Module anzeigen, die nach einer Bearbeitung durch den gesuchten Benutzer durch jemand anderen bearbeitet wurden.
 - 2.6. Abweichungen sollen kenntlich gemacht werden.
 - 2.7. Anzeigen einer Übersichtsseite für ein Modul mit allen relevanten Informationen zu diesem.
3. Sonstige Anforderungen
 - 3.1. Die Anwendung muss ohne das Installieren einer zusätzlichen Software über einen Webbrowser im Intranet erreichbar sein.
 - 3.2. Die Daten der Anwendung müssen jede Nacht bzw. nach jedem SVN-Commit automatisch aktualisiert werden.
 - 3.3. Es muss ermittelt werden, ob Änderungen auf der Produktionsumgebung vorgenommen wurden, die nicht von einer anderen Umgebung kopiert wurden. Diese Modul-liste soll als Mahnung per E-Mail an alle Entwickler geschickt werden (Peer Pressure).
 - 3.4. Die Anwendung soll jederzeit erreichbar sein.
 - 3.5. Da sich die Entwickler auf die Anwendung verlassen, muss diese korrekte Daten liefern und darf keinen Interpretationsspielraum lassen.
 - 3.6. Die Anwendung muss so flexibel sein, dass sie bei Änderungen im Entwicklungsprozess einfach angepasst werden kann.

Zielbestimmung

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Anhang**1. Musskriterien**

- 1.1. Modul-Liste: Zeigt eine filterbare Liste der Module mit den dazugehörigen Kerninformationen sowie Symbolen zur Einhaltung des Entwicklungsprozesses an
 - In der Liste wird der Name, die Bibliothek und Daten zum Source und Kompilat eines Moduls angezeigt.
 - Ebenfalls wird der Status des Moduls hinsichtlich Source und Kompilat angezeigt. Dazu gibt es unterschiedliche Status-Zeichen, welche symbolisieren in wie weit der Entwicklungsprozess eingehalten wurde bzw. welche Schritte als nächstes getan werden müssen. So gibt es z. B. Zeichen für das Einhalten oder Verletzen des Prozesses oder den Hinweis auf den nächsten zu tätigen Schritt.
 - Weiterhin werden die Benutzer und Zeitpunkte der aktuellen Version der Sourcen und Kompilate angezeigt. Dazu kann vorher ausgewählt werden, von welcher Umgebung diese Daten gelesen werden sollen.
 - Es kann eine Filterung nach allen angezeigten Daten vorgenommen werden. Die Daten zu den Sourcen sind historisiert. Durch die Filterung ist es möglich, auch Module zu finden, die in der Zwischenzeit schon von einem anderen Benutzer editiert wurden.
- 1.2. Tag-Liste: Bietet die Möglichkeit die Module anhand von Tags zu filtern.
 - Es sollen die Tags angezeigt werden, nach denen bereits gefiltert wird und die, die noch der Filterung hinzugefügt werden könnten, ohne dass die Ergebnisliste leer wird.
 - Zusätzlich sollen die Module angezeigt werden, die den Filterkriterien entsprechen. Sollten die Filterkriterien leer sein, werden nur die Module angezeigt, welche mit einem Tag versehen sind.
- 1.3. Import der Moduldaten aus einer bereitgestellten CSV-Datei
 - Es wird täglich eine Datei mit den Daten der aktuellen Module erstellt. Diese Datei wird (durch einen Cronjob) automatisch nachts importiert.
 - Dabei wird für jedes importierte Modul ein Zeitstempel aktualisiert, damit festgestellt werden kann, wenn ein Modul gelöscht wurde.
 - Die Datei enthält die Namen der Umgebung, der Bibliothek und des Moduls, den Programmtyp, den Benutzer und Zeitpunkt des Sourcecodes sowie des Kompilats und den Hash des Sourcecodes.
 - Sollte sich ein Modul verändert haben, werden die entsprechenden Daten in der Datenbank aktualisiert. Die Veränderungen am Source werden dabei aber nicht ersetzt, sondern historisiert.
- 1.4. Import der Informationen aus Subversion (SVN). Durch einen „post-commit-hook“ wird nach jedem Einchecken eines Moduls ein PHP-Script auf der Konsole aufgerufen, welches die Informationen, die vom SVN-Kommandozeilentool geliefert werden, an NatInfo übergibt.
- 1.5. Parsen der Sourcen
 - Die Sourcen der Entwicklungsumgebung werden nach Tags, Links zu Artikeln im Wiki und Programmbeschreibungen durchsucht.
 - Diese Daten werden dann entsprechend angelegt, aktualisiert oder nicht mehr gesetzte Tags/Wikiartikel entfernt.
- 1.6. Sonstiges
 - Das Programm läuft als Webanwendung im Intranet.
 - Die Anwendung soll möglichst leicht erweiterbar sein und auch von anderen Entwicklungsprozessen ausgehen können.
 - Eine Konfiguration soll möglichst in zentralen Konfigurationsdateien erfolgen.

Produkteinsatz

PROTOTYP ZUR ÜBERWACHUNG DER LUFTQUALITÄT

Entwurf und Bau eines Prototyps zur Überwachung der Luftqualität
in den Schulungsräumen der IT-Akademie.

Anhang

1. Anwendungsbereiche
 - 1.1. Die Webanwendung dient als Anlaufstelle für die Entwicklung. Dort sind alle Informationen für die Module an einer Stelle gesammelt. Vorher getrennte Anwendungen werden ersetzt bzw. verlinkt.
2. Zielgruppen
 - 2.1. NatInfo wird lediglich von den Natural-Entwicklern in der EDV-Abteilung genutzt.
3. Betriebsbedingungen
 - 3.1. Die nötigen Betriebsbedingungen, also der Webserver, die Datenbank, die Versionsverwaltung, das Wiki und der nächtliche Export sind bereits vorhanden und konfiguriert. Durch einen täglichen Cronjob werden entsprechende Daten aktualisiert, die Webanwendung ist jederzeit aus dem Intranet heraus erreichbar.