



Diplomarbeit

Entomophagie

Untertitel der Arbeit

Imst, 20. Dezember 2017

Eingereicht von

Leonid Hammer

Verantwortlich für IT: HTML, CSS, BWL: Kaufvertrag

Kevin Glatz

Verantwortlich für IT: SQL, C# BWL: Kaufvertrag

Tobias Haslwanter

Verantwortlich für IT: HTML, CSS, BWL: Kaufvertrag

Florian Tipotsch

Verantwortlich für IT: SQL, C# BWL: Kaufvertrag

Eingereicht bei

Stefan Stolz und Nina Margreiter

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbst verfasst und keine anderen als die angeführten Behelfe verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht. Ich bin damit einverstanden, dass meine Arbeit öffentlich zugänglich gemacht wird.

Ort, Datum

Leonid Hammer

Kevin Glatz

Tobias Haslwanter

Florian Tipotsch

Abnahmeerklärung

Hiermit bestätigt der Auftraggeber, dass das übergebene Produkt dieser Diplomarbeit den dokumentierten Vorgaben entspricht. Des Weiteren verzichtet der Auftraggeber auf unentgeltliche Wartung und Weiterentwicklung des Produktes durch die Projektmitglieder bzw. die Schule.

Ort, Datum

Thorsten Schwerte

Vorwort

z. B. Hinweise, wie das bearbeitete Thema gefunden wurde oder Dank für die Betreuung (Kooperationspartner/in, Betreuer/innen, Sponsoren) etc.

Abstract (Deutsch)

(ca. $\frac{1}{2}$ bis max. 2 Seiten) Kurzbeschreibung von Aufgabenstellung und Problemlösung.

Abstract (Englisch)

(ca. ½ bis max. 2 Seiten)

Inhaltsverzeichnis

Abbildungsverzeichnis	10
Tabellenverzeichnis	11
Quelltexte	12
1. Einleitung	15
2. Projektmanagement	16
2.1. Metainformationen	16
2.1.1. Team	16
2.1.2. Betreuer	16
2.1.3. Partner	16
2.1.4. Ansprechpartner	16
2.2. Vorerhebungen	16
2.2.1. Projektzieleplan	16
2.2.2. Projektumfeld	17
2.2.3. Risikoanalyse	17
2.3. Pflichtenheft	17
2.3.1. Zielbestimmung	17
2.3.2. Produkteinsatz und Umgebung	17
2.3.3. Funktionalitäten	18
2.3.4. Testszenarien und Testfälle	18
2.3.5. Liefervereinbarung	18

2.4.	Planung	19
2.4.1.	Projektstrukturplan	19
2.4.2.	Meilensteine	19
2.4.3.	Gant-Chart	19
2.4.4.	Abnahmekriterien	19
2.4.5.	Pläne zur Evaluierung	19
2.4.6.	Ergänzungen und zu klärende Punkte	19
3.	Vorstellung des Produktes	20
4.	Eingesetzte Technologien	21
4.1.	Technologie für Webapp	22
4.2.	Yii2	22
4.2.1.	Was ist Yii	22
4.2.2.	Alternativen für Yii	22
4.2.3.	Warum haben wir uns für Yii2 entschieden	23
4.3.	Gas-Sensoren	24
4.3.1.	MQ Gas Sensoren	24
4.3.2.	Adafruit CCS811	25
5.	Problemanalyse	26
5.1.	USE-Case-Analyse	26
5.2.	Domain-Class-Modelling	27
5.3.	User-Interface-Design	27
6.	Systementwurf	28
6.1.	Architektur	28
6.1.1.	Design der Komponenten	28
6.1.2.	Benutzerschnittstellen	29
6.1.3.	Datenhaltungskonzept	29
6.1.4.	Konzept für Ausnahmebehandlung	29
6.1.5.	Sicherheitskonzept	29
6.1.6.	Design der Testumgebung	30
6.1.7.	Design der Ausführungsumgebung	30

6.2. Detailentwurf	30
7. Implementierung	32
7.1. Handy-App	33
8. Deployment	34
9. Tests	35
9.1. Systemtests	35
9.2. Akzeptanztests	35
10. Projektevaluation	36
11. Benutzerhandbuch	37
12. Betriebswirtschaftlicher Kontext	38
13. Zusammenfassung	39
Literaturverzeichnis	40
A. Anhang-Kapitel	41
A.1. Anhang-Section	41

Abbildungsverzeichnis

Tabellenverzeichnis

Quelltexte

Einleitende Bemerkungen

Notationen

Beschreibung wie Code, Hinweise, Zitate etc. formatiert werden

1. Einleitung

2. Projektmanagement

2.1. Metainformationen

2.1.1. Team

2.1.2. Betreuer

2.1.3. Partner

2.1.4. Ansprechpartner

2.2. Vorerhebungen

2.2.1. Projektzieleplan

Projektziele-Hierarchie - SMART

2.2.2. Projektumfeld

- Identifikation der Stakeholder
- Charakterisierung der Stakeholder
- Maßnahmen
- Grafische Darstellung des Umfeldes

2.2.3. Risikoanalyse

- Risikomatrix

2.3. Pflichtenheft

2.3.1. Zielbestimmung

- Projektbeschreibung
- IST-Zustand
- SOLL-Zustand
- NICHT-Ziele (Abgrenzungskriterien)

2.3.2. Produkteinsatz und Umgebung

- Anwendungsgebiet
- Zielgruppen

- Betriebsbedingungen
- Hard-/Softwareumgebung

2.3.3. Funktionalitäten

- MUSS-Anforderungen
 - Funktional
 - Nicht-funktional
- KANN-Anforderungen
 - Funktional
 - Nicht-funktional

2.3.4. Testszenarien und Testfälle

- Beschreibung der Testmethodik
- Testfall 1
- Testfall 2
- ...

2.3.5. Liefervereinbarung

- Lieferumfang
- Modus
- Verteilung(Deployment)

2.4. Planung

2.4.1. Projektstrukturplan

2.4.2. Meilensteine

2.4.3. Gant-Chart

2.4.4. Abnahmekriterien

2.4.5. Pläne zur Evaluierung

2.4.6. Ergänzungen und zu klärende Punkte

3. Vorstellung des Produktes

Vorstellung des fertigen Produktes anhand von Screenshots, Bildern, Erklärungen.

4. Eingesetzte Technologien

- (Löschen wenn ihr alle Technologien habt) Technologien die aus dem Unterricht bekannt sind, nur nennen und deren Einsatzzweck im Projekt beschreiben, nicht die Technologien selbst.
- Technologien die aus dem Unterricht nicht bekannt sind, im Detail beschreiben incl. deren Einsatz im Projekt
- Fokus auf eingesetzten Frameworks

4.1. Technologie für Webapp

- PHP - Für Handyapp
- Html - Für Handyapp
- MySql - Für Datenbanken
- Yii2 - Für Handyapp

4.2. Yii2

4.2.1. Was ist Yii

Yii ist ein high Performance PHP Framework welches vorallem für die Entwicklung im Web2.0 eingesetzt wird. Web 2.0 fördert die user aktiv im Web mitzumachen. Diese können eigenen Beiträge erstellen und diese auf der Website anzeigen lassen. Web2

4.2.2. Alternativen für Yii

Yii kann sehr weitreichend eingesetzt werden. Mit dem richtigen Wissen und Fähigkeiten kann man alles was mit einer PHP Seite möglich ist ganz einfach in Yii2 umsetzen. Dabei gibt es auch viele Vorteile:

- CRUD-Creator
- Model Generator
- Einfache implementierung von HTML Formulare

Allerdings sind Frameworks nicht Administrationsfreundlich da sie sehr viel Vorkentniss erfordern um diese richtig zu implementieren. Einfacher zu implementieren sind CMS Systeme. Es gibt sehr viele Große CMS Systeme zum Beispiel:

- Joomla
- Wordpress
- Drupal
- Contao

Diese haben wir auch schon im Unterricht besprochen und damit Websiten erstellt. Vorteile sind vorallem die einfache implementierung und rasche einrichtung einer Website. Auch SEO wird von den CMS Systemen vereinfacht. Nachteile sind allerdings oft eingeschränkte möglichkeiten und grenzen die das CMS setzt.

Es gibt auch noch viele andere Frameworks zum Beispiel:

4.2.3. Warum haben wir uns für Yii2 entschieden

Der Hauptgrund waum wir uns gegen CMS Systeme entschieden haben sind die eingeschränkten möglichkeiten die wir damit hätten. Bei Yii2 können wir die gesamte Website nach unseren Bedarf zusammenstellen und auch so bearbeiten wie wir es wollen. Es war uns auch wichtig das wir nach modernen Entwurfsmustern arbeiten (hier MVC).

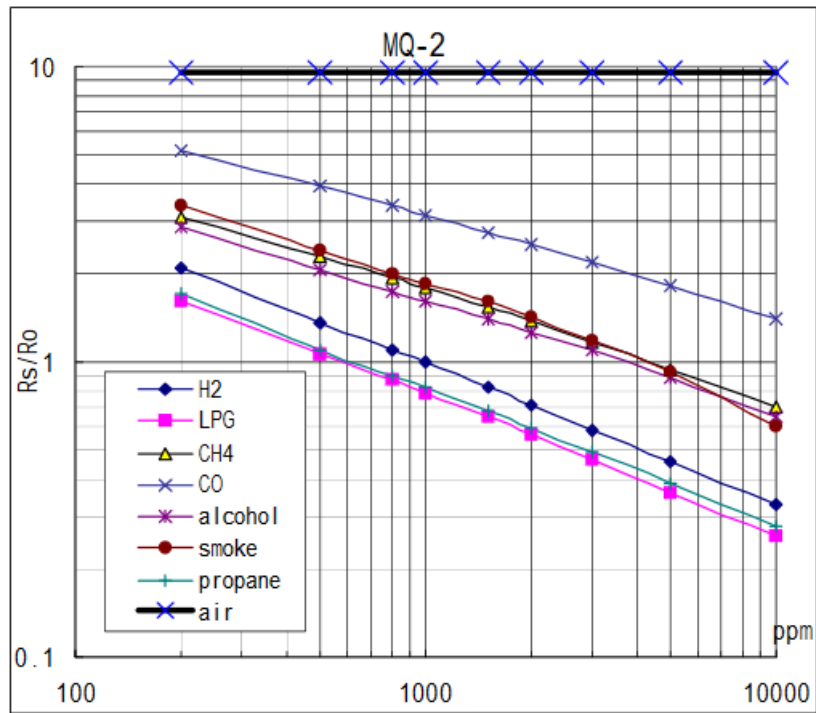
4.3. Gas-Sensoren

4.3.1. MQ Gas Sensoren

Es gibt mehrer MQ Gas Sensoren zum Beispiel:

- MQ2 Methane, Butane, LPG, smoke
- MQ3 Alcohol, Ethanol, smoke
- MQ4 Methane, CNG Gas
- MQ5 Natural gas, LPG
- MQ6 LPG, butane gas
- MQ7 Carbon Monoxide
- MQ8 Hydrogen Gas
- MQ9 Carbon Monoxide, flammable gasses
- Mehr gibt es auf der Website: Gas

In der Schule haben wir den MQ2 zur verfügung stehend werden wir auch von der Schule den Adafruit CCS811 bereitgestellt bekommen. Wir bedanken uns dafür oftmals.



Datasheet Gas

4.3.2. Adafruit CCS811

5. Problemanalyse

5.1. USE-Case-Analyse

- UseCases auf Basis von Benutzerzielen identifizieren:
 - Benutzer eines Systems identifizieren
 - Benutzerziele identifizieren (Interviews)
 - Use-Case-Liste pro Benutzer definieren
- UseCases auf Basis von Ereignissen identifizieren:
 - Externes Event triggert einen Prozess
 - zeitliches Event triggert einen Prozess (Zeitpunkt wird erreicht)
 - State-Event (Zustandsänderung im System triggert einen Prozess)
- Werkzeuge:
 - USE-Case-Beschreibungen (textuell, tabellarisch)
 - USE-Case-Diagramm
 - Aktivitätsdiagramm für den Use-Case (Interaktion zwischen Akteur und System abbilden)
 - System-Sequenzdiagramm (Spezialfall eines Sequenzdiagramms: Nur 1 Akteur und 1 Objekt, das Objekt ist das komplette System, es geht um die Input/Output Requirements, die abzubilden sind)

5.2. Domain-Class-Modelling

- "Dinge" (Rollen, Einheiten, Geräte, Events etc.) identifizieren, um die es im Projekt geht
- ER-Modellierung oder Klassendiagramme
- Zustandsdiagramme (zur Darstellung des Lebenszyklus von Domain-Klassen darstellen)

5.3. User-Interface-Design

- Mockups
- Wireframes

6. Systementwurf

6.1. Architektur

6.1.1. Design der Komponenten

Darstellung und Beschreibung der Systemarchitektur;

- statische Zerlegung des Systems in seine physischen Bestandteile (Komponenten, Komponentendiagramm)
- (textuelle) Beschreibung des dynamischen Zusammenwirkens aller Komponenten
- (textuelle) Beschreibung der Strategie für die Architektur, d. h. wie die Architektur in Statik und Dynamik funktionieren soll.
- Verwendung von Referenzarchitekturen bzw. Architekturmustern (als Schablonen, z.B. MVC, Plugin, Pipes and Filters)
 - MVC
 - Schichten
 - Pipes
 - Request Broker
 - Service-Oriented

6.1.2. Benutzerschnittstellen

- Design des UIs
- Dialoge, Dialogsteuerung, Ergonomie, Gestaltung, Eingabeüberprüfungen

6.1.3. Datenhaltungskonzept

- Design der Datenbank (ER-Modell)
- Design des Zugriffs auf diese Daten (Datenhaltungskonzept)
- Caching, Transaktionen

6.1.4. Konzept für Ausnahmebehandlung

- Systemweite Festlegung, wie mit Exceptions umgegangen wird
- Exceptions sind primär aus den Bereichen UI, Persistenz, Workflow-Management

6.1.5. Sicherheitskonzept

Beschreibung aller sicherheitsrelevanten Designentscheidungen

- Design der Security-Elemente
- Design von Safety-Elementen (Fehlertoleranz, Verfügbarkeit etc.)

6.1.6. Design der Testumgebung

- wie wird getestet (Unit-Testing, Integrationstesting, Systemtests, Akzeptanztests)
- Testumgebung, Testprozess, Teststrategie, Testmethoden, Testfälle

6.1.7. Desing der Ausführungsumgebung

- Deployment (DevOps)
- Betrieb (besonders Hoch- und Hertunerfahren der Anwendung)

6.2. Detailentwurf

Design jedes einzelnen USE-Cases

- Design-Klassendiagramme vom Domain-Klassendiagramm ableiten (incl. detaillierter Darstellung und Verwendung von Vererbungshierarchichen, abstrakten Klassen, Interfaces)
- Sequenzdiagramme vom System-Sequenz-Diagramm ableiten
- Aktivitätsdiagramme
- Detaillierte Zustandsdiagramme für wichtige Klassen

Verwendung von CRC-Cards (Class, Responsibilities, Collaboration) für die Klassen

- um Verantwortlichkeiten und Zusammenarbeit zwischen Klassen zu definieren und
- um auf den Entwurf der Geschäftslogik zu fokussieren

Design-Klassen für jeden einzelnen USE-Case können z.B. sein:

- UI-Klassen
- Data-Access-Klassen
- Entity-Klassen (Domain-Klassen)
- Controller-Klassen
- Business-Logik-Klassen
- View-Klassen

Optimierung des Entwurfs (Modularisierung, Erweiterbarkeit, Lesbarkeit):

- Kopplung optimieren
- Kohäsion optimieren
- SOLID
- Entwurfsmuster einsetzen

7. Implementierung

Detaillierte Beschreibung der Implementierung aller Teilkomponenten der Software entlang der zentralsten Use-Cases:

- GUI-Implementierung
- Controllerlogik
- Geschäftslogik
- Datenbankzugriffe

Detaillierte Beschreibung der Teststrategie (Testdriven Development):

- UNIT-Tests (Funktional)
- Integrationstests

Zu Codesequenzen:

- kurze Codesequenzen direkt im Text (mit Zeilennummern auf die man in der Beschreibung verweisen kann)
- lange Codesequenzen in den Anhang (mit Zeilennummer) und darauf verweisen (wie z.B. hier ??)

7.1. Handy-App

Für unser Projekt erstellen wir eine Handy-App mit der man die Daten seiner eigenen Zuchtkammer anzeigen lassen kann. Wir haben geplant das man sich mit der Seriennummer der Box Registrieren kann und dann am Handy über eine Web-App alle Daten anzeigen lassen kann. Folgende Daten sollte man auslesen können:

- Sauerstoff
- Luftfeuchtigkeit
- Gewicht
- Futtermenge
- ungefähre Zeit bis zu Reife

8. Deployment

- Umsetzung der Ausführungsumgebung
- Deployment
- DevOps-Thema

9. Tests

9.1. Systemtests

Systemtests aller implementierten Funktionalitäten lt. Pflichtenheft

- Beschreibung der Teststrategie
- Testfall 1
- Testfall 2
- Testfall 3
- ...

9.2. Akzeptanztests

10. Projektevaluation

siehe Projektmanagement-Unterricht

11. Benutzerhandbuch

falls im Projekt gefordert

12. Betriebswirtschaftlicher Kontext

BW-Teil

13. Zusammenfassung

- Etwas längere Form des Abstracts
- Detaillierte Beschreibung des Outputs der Arbeit

Literaturverzeichnis

[Gas] MQ Gas sensors. . – URL <https://playground.arduino.cc/Main/MQGasSensors>. – Zugriffsdatum: 2017-12-20

[Datasheet] MQ Gas sensors. . – URL <https://www.mouser.com/ds/2/321/605-00008-MQ-2-Datasheet-370464.pdf>. – Zugriffsdatum: 2017-12-20

[Web2] Web 2.0. . – URL https://en.wikipedia.org/wiki/Web_2.0. – Zugriffsdatum: 2017-12-20

A. Anhang-Kapitel

A.1. Anhang-Section

Testtext