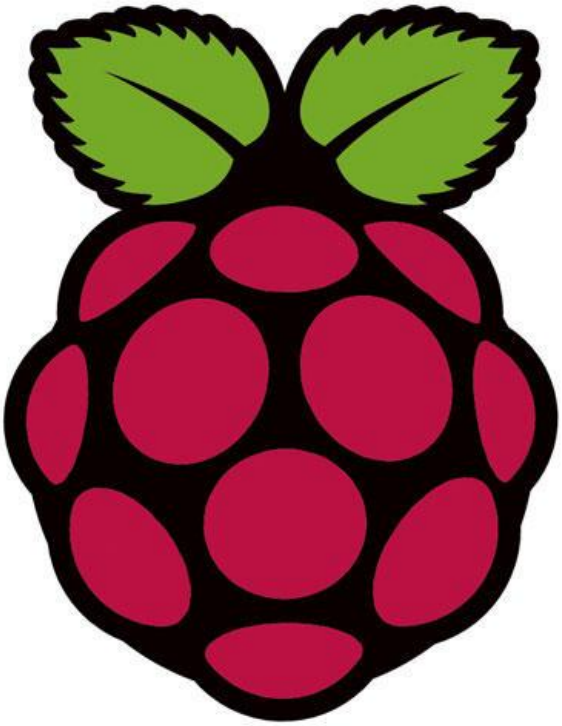

Wetterstation mit Raspberry Pi

Version: 0.7

Datum: 16.11.14

Projektteam: Andreas Hasler / David Daniel

Inhaltsverzeichnis

1 Informationen zum Dokument

1.1 Zweck des Dokuments

Dieses Dokument beinhaltet die Projektdokumentation zum Projekt *Wetterstation*, welches im Zuge des 9. Semesters im Fach Embedded Systems und Hardware Hacking an der FFHS umgesetzt wurde.

1.2 Versionskontrolle

Ausgabe	Datum	Autor	Bemerkungen
0.1	27.09.2014	Andreas Hasler	Initialversion
0.2	28.09.2014	Andreas Hasler	Anpassungen Anforderungen und Terminplan
0.3	29.09.2014	Andreas Hasler	Anpassungen Anforderungen
0.4	10.10.2014	Andreas Hasler	Use-Cases hinzugefügt
0.5	25.10.2014	Andreas Hasler	Grobentwurf / Lösungsfindung / Hardware Entwurf
0.6	16.11.2014	David Daniel	Software Entwurf (Steuerung) / Qualitätssicherung / Testing
0.7	16.11.2014	Andreas Hasler	Entwurf Webseite

1.3 Referenzierte Dokumente

Dokument / Bemerkungen
Präsenz Block 2 (27.09.2014) mit der Aufgabenstellung auf Seite 11

2 Projektdefinition

Mit dem Raspberry Pi soll eine Wetterstation erstellt werden, welche Wetterdaten (Luftdruck, Temperatur, Feuchtigkeit und Lichtstärke) ermittelt und auf einem Display alternierend darstellt. Zusätzlich sollen die Wetterdaten auf dem Raspberry Pi in einer Datenbank persistent abgespeichert werden, so dass die aktuellsten Daten Online eingesehen werden können.

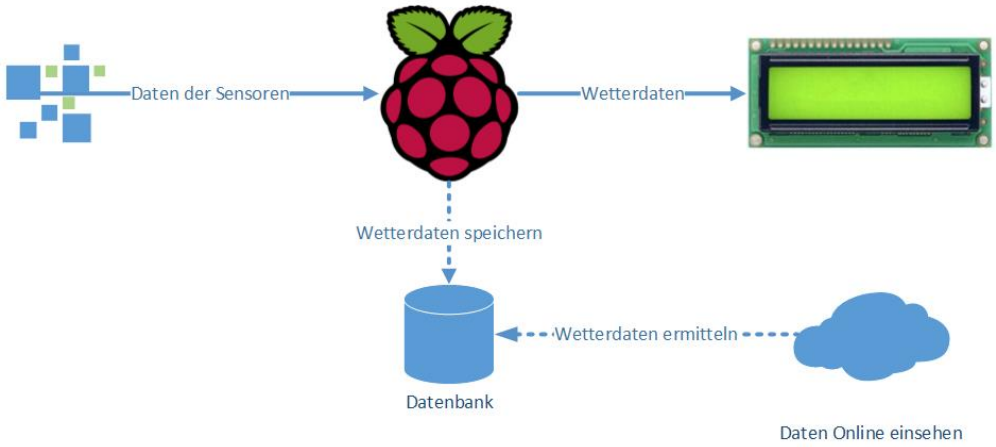

Abbildung 1: Projektidee (Skizze)

3 Anforderungen

Nachfolgend werden die funktionalen sowie die nicht funktionalen Anforderungen an das System beschrieben. Bei den funktionalen Anforderungen handelt es sich ausschliesslich um Muss-Anforderungen.

3.1 Funktionale Anforderungen

- Die Wetterdaten (Luftdruck, Temperatur, Feuchtigkeit und Lichtstärke) sind mittels Sensoren zu ermitteln.
- Die ermittelten Wetterdaten sind persistent in einer Datenbank abzuspeichern.
- Des Weiteren sind die ermittelten Wetterdaten auf einem Display auszugeben.
- Die aktuellsten Wetterdaten müssen Online eingesehen werden können.

3.2 Nicht funktionale Anforderungen

- Die Projekt muss am 14.12.2014 (inkl. Dokumentation) abgeschlossen sein
- Das Projekt muss mittels Präsentation am 20.12.2014 anlässlich der 5. Präsenz vorgestellt werden.
- Die Signal- und Datenverarbeitung hat auf dem Raspberry Pi zu erfolgen.

4 Kontextdiagramm

Nachfolgend wird das Kontextdiagramm des Projekts (inkl. den Kann-Zielen) darstellt:

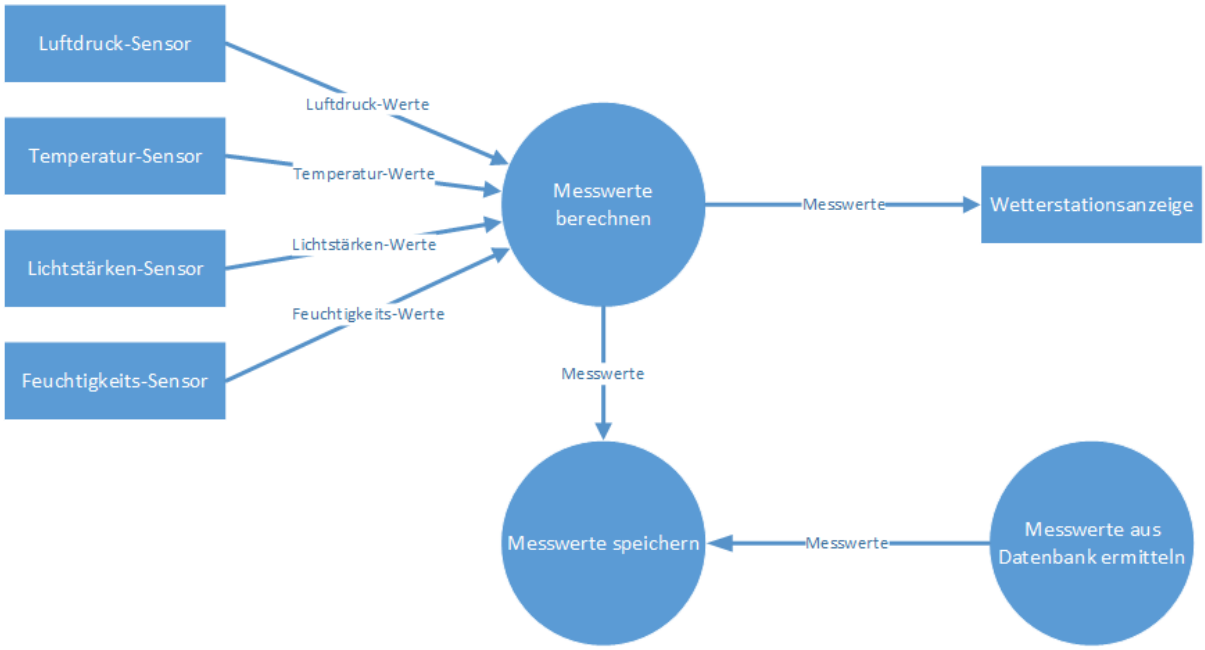

Abbildung 2: Kontextdiagramm des Projekts

Die Werte der einzelnen Sensoren werden ermittelt und in einer zentral berechnet / umgerechnet. Anschliessend werden die Daten an die Anzeige weitergeleitet. Zudem werden die Messwerte nach der Berechnung persistent abgespeichert, damit die Daten Online abgefragt werden können.

5 Terminplan

Bezeichnung	Termin
Projektskizze erstellt	**05.10.2014**
Anforderungen / Kontextdiagramm / Terminplan	**12.10.2014**
Use-Cases erstellen / verifizieren	19.10.2014
Lösungsentwürfe erstellen (Grobentwurf) / Lösungsfindung	26.10.2014
Schaltungsentwurf / Softwareentwurf / Testkonzept	**17.11.2014**
Schaltung / Hardware umsetzen	22.11.2014
Software implementieren (Ermittlung Messwerte, Weitergabe der Messwerte an den LCD-Bildschirm)	07.12.2014
Applikationstest und Abnahme	15.12.2014
Projektdokumentation finalisieren	**15.12.2014**
Präsentation anlässlich Präsenz 5	**20.12.2014**

Die Meilensteine (Abgaben in moodle) sind Fett markiert und sind zwingend einzuhalten.

6 Use Cases

6.1 Diagramm

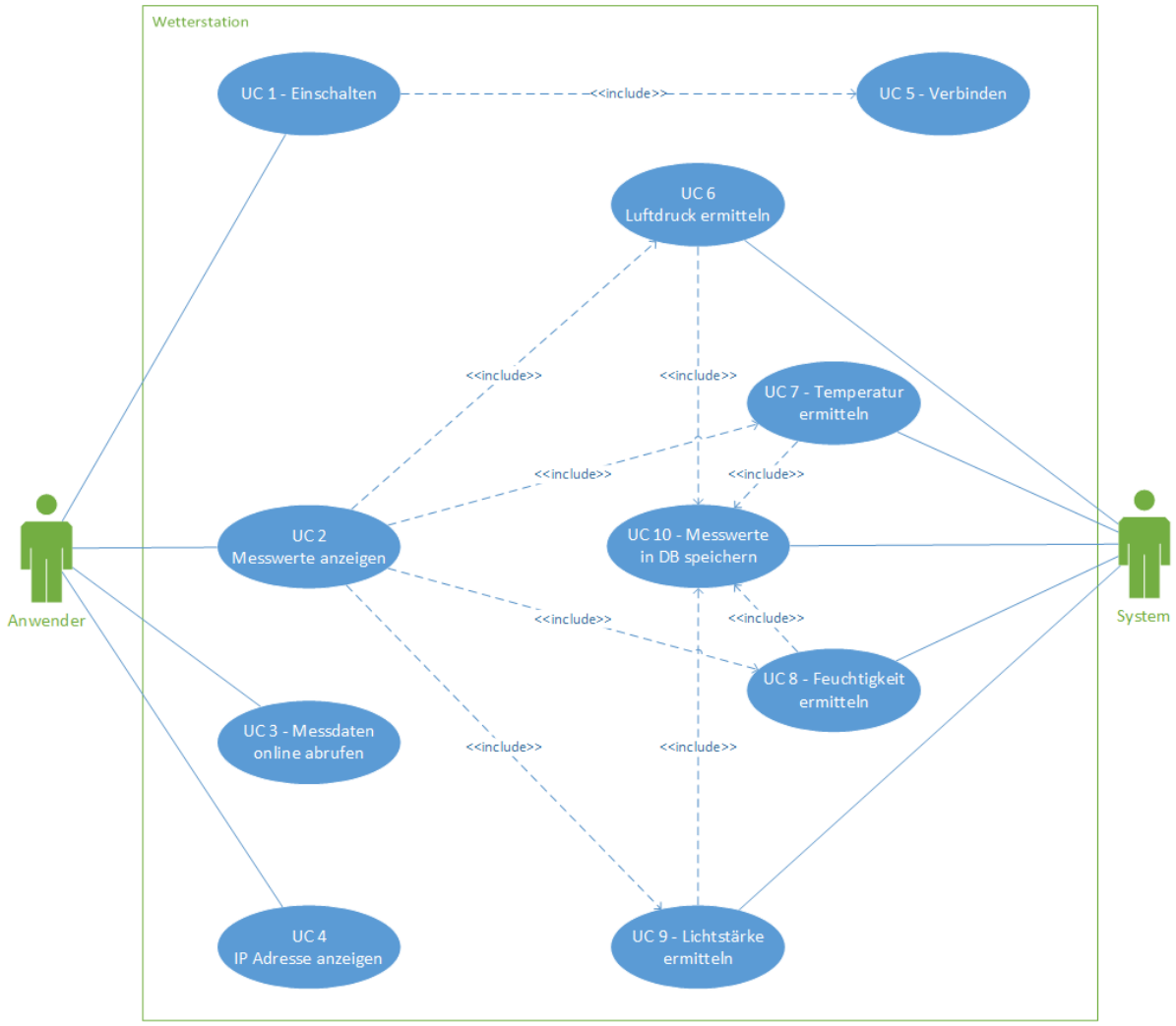

Abbildung 3: Use Cases

6.2 Beschreibungen

UC 1 - Einschalten	
Beschreibung	Die Wetterstation einschalten
Stakeholder	Anwender
Uses	*UC 5 - Verbinden*
Vorbedingungen	Die Wetterstation ist noch nicht eingeschaltet
Nachbedingungen	Die Wetterstation ist eingeschaltet
Ablauf	Verbindung Wetterstation / 230V Steckdose mit Netzteil

UC 2 - Messwerte anzeigen	
Beschreibung	Die unterschiedlichen Messwerte sollen auf dem LCD Display der Wetterstation angezeigt werden
Stakeholder	Anwender
Uses	*UC 6 - Luftdruck ermitteln, UC 7 - Temperatur ermitteln, UC 8 - Feuchtigkeit ermitteln, UC 9 - Lichtstärke ermitteln*
Vorbedingungen	• Wetterstation eingeschaltet • Messwerte durch die Sensoren ermittelt
Nachbedingungen	Messwerte werden auf dem LCD-Display angezeigt
Ablauf	1. Anwender betätigt den Schalter 1 am LCD-Display (gilt nur wenn der Schalter 2, 3 oder 4 zuvor betätigt wurde, ansonsten werden die Messwerte standardmässig anzeigt) 2. Messwerte werden auf dem LCD-Display angezeigt (Pro LCD-Reihe ein Messwert)

UC 3 – Messdaten online abrufen	
Beschreibung	Die Messdaten dem Anwender Online zur Verfügung stellen
Stakeholder	Anwender
Vorbedingungen	• Messdaten in der Datenbank vorhanden • IP-Adresse des Raspberry Pi bekannt
Nachbedingungen	Die Messwerte konnten Online ermittelt werden
Ablauf	1. Anwender verbindet sich mittels der bekannten IP-Adresse mit der Schnittstelle auf dem Raspberry Pi

	2. Messwerte werden aus der Schnittstelle auf Grund der Angabe des Datumbereichs (resp. des aktuellen Wertes) ausgelesen

UC 4 - IP Adresse anzeigen	
Beschreibung	Die IP-Adresse des Raspberry Pi wird auf dem LCD-Display angezeigt (für Fernwartung oder Zugriffe auf die Online-Schnittstelle)
Stakeholder	Anwender
Vorbedingungen	Wetterstation eingeschaltet
Nachbedingungen	IP-Adresse des Raspberry Pi wird auf dem LCD-Display dargestellt
Ablauf	1. Anwender betätigt den Schalter 2 am LCD- 2. IP-Adresse wird auf dem LCD-Display dargestellt

UC 5 - Verbinden	
Beschreibung	Das Raspberry Pi verbindet sich beim Systemstart mit der Hardware zur Ermittlung der Messwerte.
Stakeholder	*System (Raspberry Pi)*
Vorbedingungen	• Wetterstation eingeschaltet • Hardware zur Ermittlung der Messwerte an das Raspberry Pi angeschlossen und bereit
Nachbedingungen	Das System ist mit der Hardware zur Ermittlung der Messwerte verbunden
Ablauf	1. Verbindung mit der Hardware zur Ermittlung der Messwerte aufbauen (IP-Verbindung). a. Bei einem Kommunikationsfehler soll dieser auf dem LCD Display ausgeben werden. b. Kann die Verbindung hergestellt werden, kann mit der Ermittlung der Messwerte begonnen werden.

UC 6 - Luftdruck ermitteln	
Beschreibung	Der aktuelle Luftdruck wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• Wetterstation eingeschaltet

	• Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. Validieren des ermittelten Messwertes a. Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 4. Ausgabe des Messwertes auf dem LCD-Display (UC 2) 5. Speicherung des Messwertes in der DB (UC 9)

UC 7 - Temperatur ermitteln	
Beschreibung	Der aktuelle Temperatur wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• Wetterstation eingeschaltet • Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. Validieren des ermittelten Messwertes a. Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 4. Ausgabe des Messwertes auf dem LCD-Display (UC 2) 5. Speicherung des Messwertes in der DB (UC 9)

UC 8 - Feuchtigkeit ermitteln	
Beschreibung	Die aktuelle Feuchtigkeit wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• Wetterstation eingeschaltet • Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. Validieren des ermittelten Messwertes a. Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 4. Ausgabe des Messwertes auf dem LCD-Display (UC 2) 5. Speicherung des Messwertes in der DB (UC 9)

UC 9 - Lichtstärke ermitteln	
Beschreibung	Die aktuelle Lichtstärke wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• Wetterstation eingeschaltet • Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. Validieren des ermittelten Messwertes

	a. Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 4. Ausgabe des Messwertes auf dem LCD-Display (UC 2) 5. Speicherung des Messwertes in der DB (UC 9)

UC 10 - Messwerte in DB speichern	
Beschreibung	Die ermittelten Messwerte in die Datenbank speichern
Stakeholder	*System (Raspberry Pi)*
Vorbedingungen	• Messwerte wurden von den entsprechenden Sensoren ermittelt • Datenbank auf dem Raspberry Pi verfügbar
Nachbedingungen	Messwerte wurden in der Datenbank hinterlegt
Ablauf	Messwerte werden in der Datenbank abgespeichert (bei einem allfälligen Zugriffsfehler wird der Fehler nicht nach aussen populiert).

7 Grobentwurf

7.1 Hardware

Bei der Umsetzung der Hardware-Schaltung gibt es 2 mögliche Varianten, wie die elektronischen Bauteile (Sensoren und LCD-Display) mit dem Raspberry Pi (und damit mit der Steuerung) verbunden werden können:

- Direkte Anbindung
- Indirekte Anbindung über einen Master-Baustein

7.1.1 Direkte Anbindung

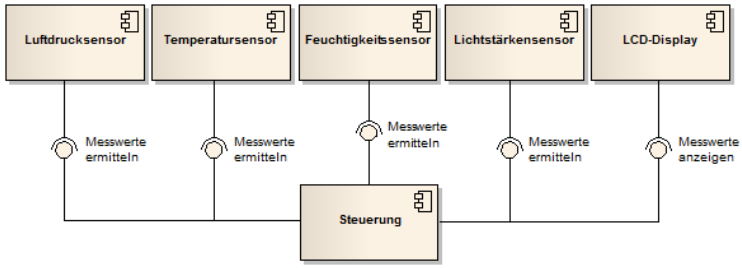

Abbildung 4: Direkte Anbindung der elektronischen Bauteile an das Raspberry Pi / die Steuerung

Bei der direkten Anbindung werden alle Sensoren sowie der LCD-Display direkt mit dem Raspberry Pi verbunden. Jedes elektronische Bauteil benötigt aus diesem Grunde eine separate Daten- sowie Stromzuleitung.

7.1.2 Indirekte Anbindung

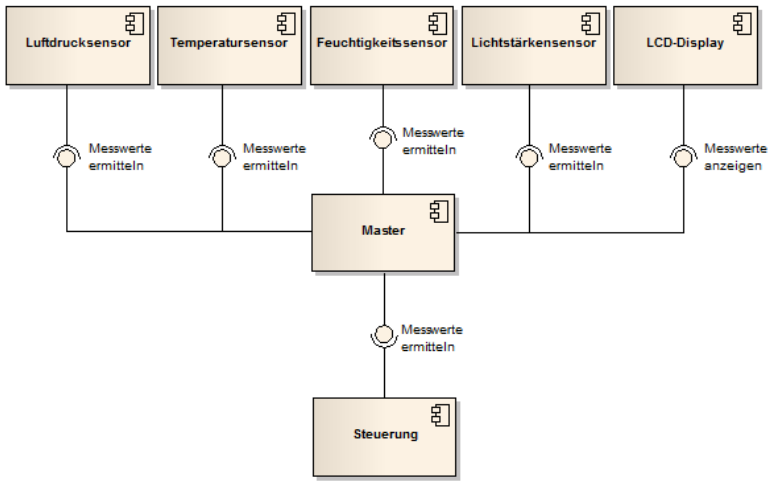

Abbildung 5: Indirekte Anbindung der elektronischen Bauteile an das Raspberry Pi / die Steuerung

Bei der indirekten Anbindung werden die Sensoren wie das LCD-Display an einen Master angeschlossen, welcher wiederum mit dem Raspberry Pi verbunden ist. Die

Stromversorgung erhalten die Bauteile durch den Master. Die Daten werden vom/zum Raspberry Pi über den Master gesandt.

7.2 Steuerung / Online-Schnittstelle

Die Steuerung ermittelt die Messwerte von den Sensoren und gibt diese innerhalb eines bestimmten Intervalls an das LCD-Display sowie an die Datenbank weiter. Die Steuerung wird nach dem Startvorgang des Raspberry Pi automatisch gestartet (kein manueller Eingriff nötig), so dass die Messwerte umgehend ermittelt und gespeichert werden. Damit die Daten von anderen Personen eingesehen werden können, wird eine Online-Schnittstelle definiert, welche die Daten gegen aussen zur Verfügung stellt.

Für die Umsetzung der Steuerung haben wir uns für C++ als Sprache entschieden. Von den auf Raspberry Pi verfügbaren Sprachen ist bei C++ die grösste Erfahrung vorhanden. Bei der Datenbank zur persistenten Speicherung der Messwerte haben wir uns für SQLite entschieden.

Die Online-Schnittstelle kann auf 3 unterschiedliche Arten umgesetzt werden:

- Webseite (PHP)
- RESTful Webservice (PHP)
- Smartphone-App (Windows Phone 8) im Zusammenspiel mit einem RESTful Webservice (PHP)

7.2.1 Webseite

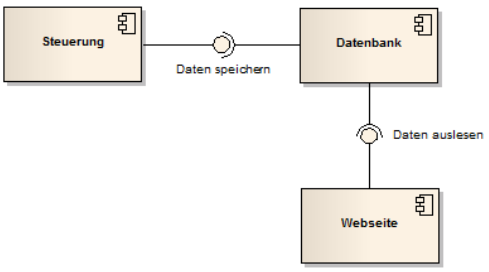

Abbildung 6: Webseite mit PHP mit Zugriff auf die Datenbank

Die Webseite greift auf die Messdaten, welche von der Steuerung in der Datenbank gespeichert wurden, zu. Die Daten werden anschliessend über ein mit PHP entwickeltes Web-UI zur Verfügung gestellt.

7.2.2 Webservice

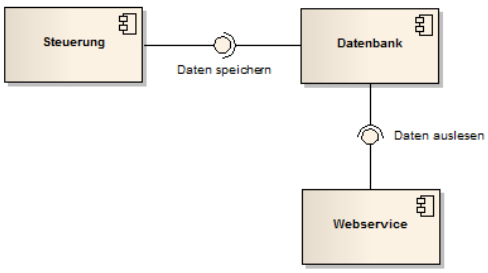

Abbildung 7: Webservice mit PHP mit Zugriff auf die Datenbank

Der Webservice soll als RESTful Service die Daten über eine Schnittstelle öffentlich zur Verfügung stellen. Die Umsetzung ist mit PHP geplant.

7.2.3 Smartphone-App mit Webservice-Zugriff

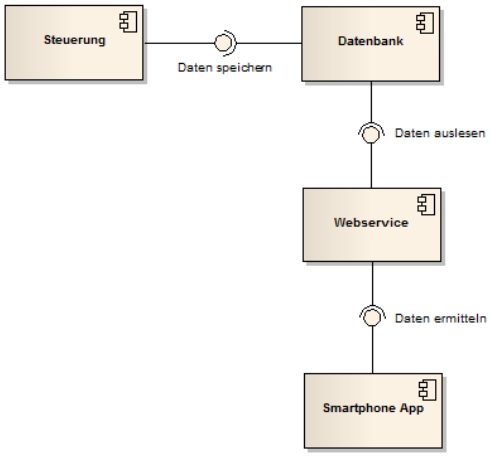

Abbildung 8: Webservice mit PHP mit Zugriff auf die Datenbank mit einer zusätzlichen Smartphone App (Windows Phone 8)

Bei dieser Variante soll ein RESTful Webservice mit PHP erstellt werden, welcher die Messdaten öffentlich zur Verfügung stellt. Zusätzlich zur Variante nur mit einem Webservice soll zusätzlich eine Smartphone-App (Windows Phone 8) erstellt werden, welche die Daten entsprechend konsumiert.

7.3 Lösungsfindung

7.3.1 Hardware

Beide Varianten lassen sich insbesondere durch einen Unterschied voneinander unterscheiden: Bei der direkten Anbindung führt jeder Sensor sowie der LCD-Bildschirm die Verbindung direkt auf das Raspberry Pi und damit auf die Steuerung. Bei der indirekten Anbindung werden die Verbindungen der elektronischen Bauteile zuerst auf einem Master-Baustein zusammengeführt und erst anschliessend auf die Steuerung gebracht.

Dies führt automatisch zu je einem Vor- wie auch Nachteil der beiden Varianten. So funktioniert bei einem Ausfall eines elektronischen Bauteils bei der direkten Anbindung die Wetterstation immer noch, wenn auch nur eingeschränkt. Allerdings ist die Schnittstelle zwischen den Bauteilen und der Steuerung komplexer, da für jedes Bauteil eine eigene Ansteuerung erstellt werden muss. Bei der indirekten Ansteuerung führt andererseits ein Ausfall des Master-Bausteins dazu, dass die Wetterstation nicht mehr funktioniert, da alle Verbindungen über diesen geführt werden. Hingegen ist die Anbindung an die einzelnen Bauteile einfacher, da nur eine Verbindung unterhalten werden muss.

Erfahrungsgemäss führen mehrere Schnittstellen gegenüber nur einer Schnittstelle in einer Steuerung eher zu mehr Problemen (Threads, Asynchron, Synchronisation). Hingegen kann die Möglichkeit eines Ausfalls eines Master-Bausteins bei sachgemässem Einsatz als sehr gering angenommen werden. **Aus diesem Grunde entscheiden wir uns für die indirekte Ansteuerung der elektronischen Bauteile**.

7.3.2 Software

Die Variante mit dem reinen Webservice scheidet aus, da diese Variante nur Benutzern mit Programmierkenntnissen einen Mehrwert bringt (und diese eine eigene Anwendung entwickeln müssten). Die Verbindung zwischen Webservice und Smartphone wäre zwar reizvoll (insbesondere weil diese Variante auf 2 unterschiedlichen Technologien aufsetzen würde), allerdings müsste selbst bei hybriden Apps pro Smartphone-Technologie (Android, iPhone, Windows Phone, Blackberry) eine App erstellt werden, zudem wären die Desktop-Benutzer ausgeschlossen. Aus diesem Grunde **bietet sich die Lösung mit der Webseite mit PHP an**, da diese sowohl von mobilen Benutzern (mit unterschiedlichen Technologien) als auch von Benutzern mit stationären Computern genutzt werden kann.

8 Detailentwurf

8.1 Hardware / Schaltung

Nachfolgend wird der Hardware-Entwurf der Wetterstation inkl. Ansteuerung des Raspberry Pi dargestellt. Folgende Hardware-Komponenten werden verwendet:

- 3 Sensoren (Temperatur und Druck, Feuchtigkeit, Lichtstärke)
- LCD-Anzeige mit 4 Zeilen
- Master (Zusammenführung der 3 Sensoren und der LCD-Anzeige) zur Verbindung mit dem Raspberry Pi
- Stromversorgung (Wandler) mit 6 bis 27 V DC Eingangsspannung und 5 V DC Ausgangsspannung für die Hardware-Bausteine und das Raspberry Pi
- Raspberry Pi

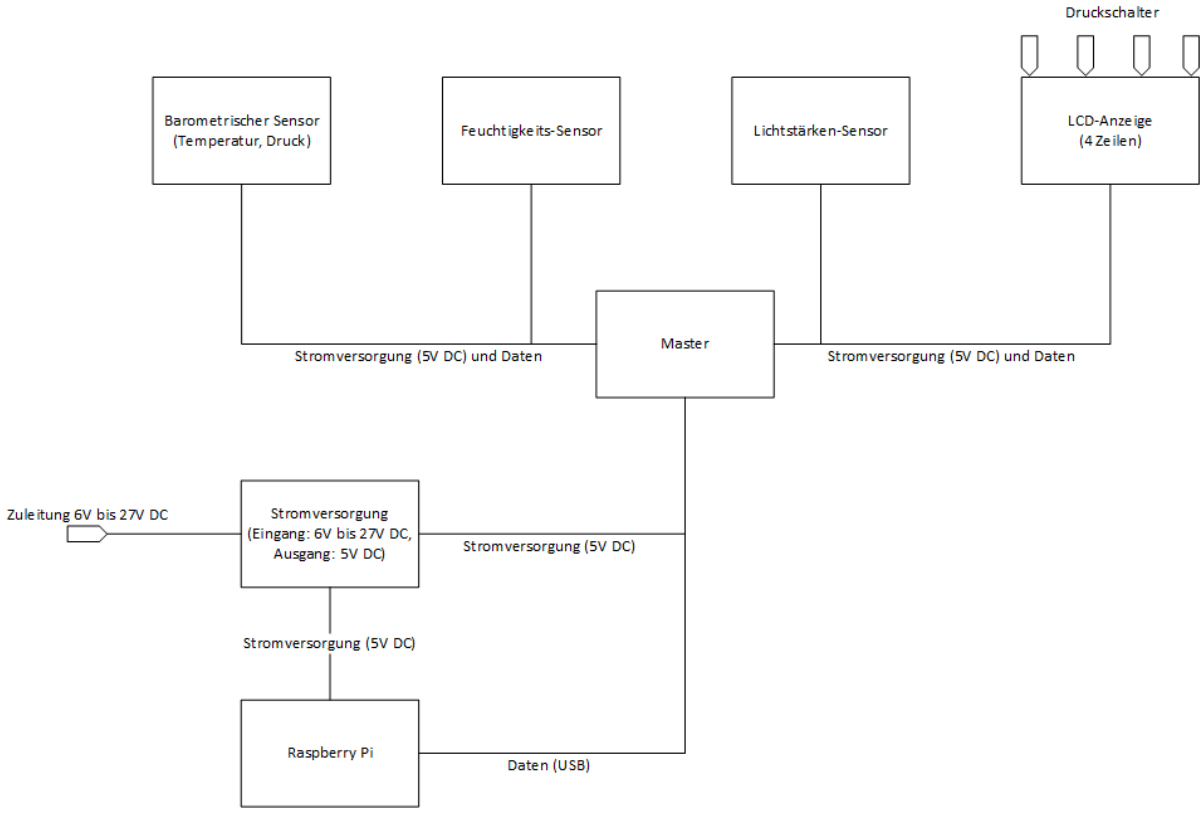

Abbildung 9: Entwurf der Schaltung der Sensoren, der LCD-Anzeige, sowie des Raspberry Pi

8.2 Steuerung

Da die Anwendung nach dem Einschalten des Gerätes nicht mehr gestoppt wird, bis das Gerät ausgeschaltet wird, handelt es sich um einen Prozess, welcher im Hintergrund läuft. Dieser Prozess ermittelt die Messwerte von den Sensoren, speichert diese in der Datenbank und stellt die Werte auf dem LCD dar. Somit benötigt der Prozess während der gesamten Laufzeit Zugriff auf die genannten Komponenten.

8.2.1 Sensoren

Die Sensoren werden periodisch von der Anwendung abgefragt (Intervall 5 Minuten). Zudem soll das Display die Anzeige aktualisieren, sobald sich ein Wert ändert. Es wird daher eine Möglichkeit benötigt, auf Ereignisse bezüglich der gemessenen Werte eines Sensors reagieren zu können. Ein Sensor benötigt daher eine Beobachter-Schnittstelle und die Möglichkeit, die aktuellen Messwerte zu liefern.

8.2.2 LCD-Display

Das LCD-Display wird zur Darstellung von verschiedenen Werten verwendet, darin werden sowohl Messwerte mit unterschiedlichen Einheiten (Grad, Pascal oder hPa, Prozent etc.) sowie die IP Adresse und evtl. auch die Zeit dargestellt. Aufgrund des beschränkten Platzes auf dem Display wäre es schwierig, dem Anwender des Displays freizustellen, wie der anzuzeigende Inhalt auf dem Display dargestellt werden soll. Das Gerät muss lediglich die gewünschten Werte auf dem Display darstellen, es reicht daher, wenn die Abstraktion des Displays dies entsprechend ermöglicht. Somit muss das Display eine Möglichkeit bieten, einen Text wie die IP Adresse, eine Temperatur etc. oder alle aktuellen Messwerte zugleich anzuzeigen.

Durch die Notwendigkeit, dass das Display die Anzeige aktualisiert, was das Beobachten der Sensoren bedingt, bietet es sich an, einen Model-View Ansatz zu verfolgen. Die Sensoren bilden so das Model, während dem das Display als View auf Änderungen am Model lauscht.

8.2.3 Schalter am LCD-Display

Der oder die Schalter fungieren als Botschafter, welche die Anwendung benachrichtigen, sobald ein Schalter gedrückt wurde. Hierzu bietet sich daher ebenfalls ein klassisches Beobachter-Modell an.

8.2.4 Ablauf

Die Anwendung ermittelt in periodischen Zeitabständen (Intervall 5 Minuten) die aktuellen Messwerte und speichert diese in der Datenbank. Auf Anfrage über einen Schalter wechselt die Anwendung die Anzeige und stellt den gewünschten Wert dar. Das periodische Speichern der Werte beeinflusst jedoch nicht die Anzeige, für die Anzeige werden stets aktuelle Werte ermittelt. Sobald die Anzeige wechselt, wird der entsprechende Wert ermittelt und angezeigt. Anschliessend werden Änderungen des aktuell dar- gestellten Wertes registriert und bei jeder Änderung wird die Anzeige aktualisiert. Der Ablauf ist in der Abbildung 10 (Ermittlung und Darstellung der Messwerte) ersichtlich. In Abbildung 11 ist der Programmfluss der periodischen Speicherung der Messwerte visualisiert. Dieser Vorgang läuft unter einem separaten Thread kontinuierlich ab.

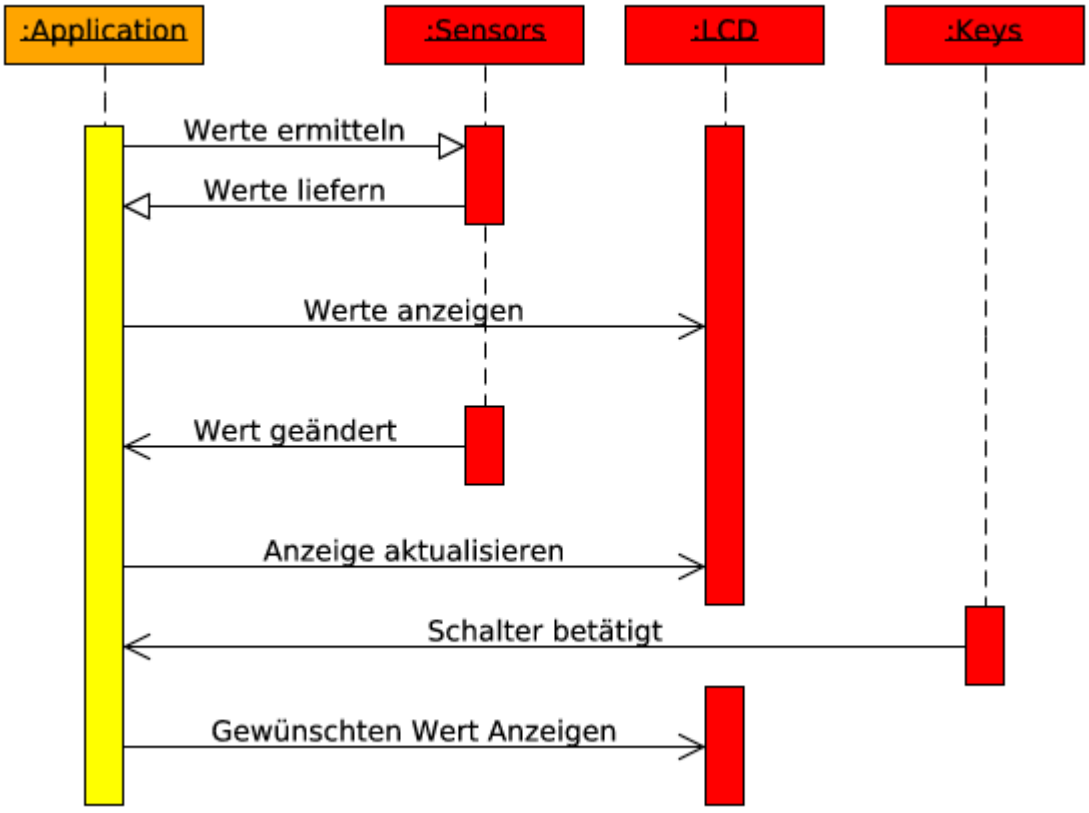

Abbildung 10: Die Anwendung reagiert auf Änderungen der Messwerte und auf Tastendruck

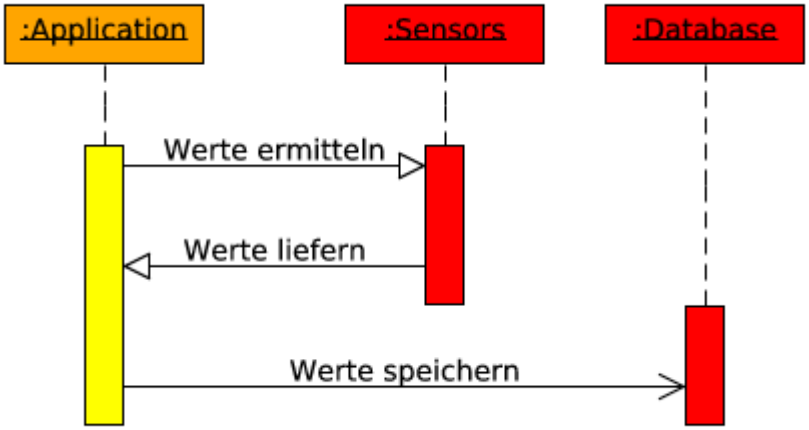

Abbildung 11: Die Messwerte werden periodisch mittels Polling gespeichert

8.2.5 Struktur

Statt einen monolithischen Ansatz zu verfolgen und dem LCD die Rolle der View zu zuordnen, sollte eher angestrebt werden, einen modularen Aufbau zu erzielen. Da andernfalls das Hinzufügen von weiteren Sensoren und die Implementierung der View schwierig sind. Eine einfache Indirektion löst dieses Problem: Statt dass die View auf alle Sensoren lauscht, existiert zu jedem Sensor eine kleine View, welche nur auf einen einzelnen Sensor lauscht und das LCD als Zeichenoberfläche erhält. Auf diese Weise können die einzelnen Sensoren leichter ausgetauscht werden und die Anwendung weist eine schwächere Kopplung auf. Zusammenfassend ergibt sich die Klassenstruktur wie sie in Abbildung 3 dargestellt ist. Der Grund, weswegen keine Schnittstelle Observer definiert wurde, liegt darin, dass mit C++ gearbeitet wird und statt einer Schnittstelle für den Observer lediglich ein Funktionsobjekt verwendet werden kann. Da der Funktor vorgängig mit Argumenten gebunden werden kann, muss dem Observer beim Aufruf kein spezifisches Argument mitgegeben werden. Generell interessiert den Beobachter lediglich, dass sich der Wert des beobachteten Sensors geändert hat. Eine Schnittstelle des Beobachters, welche beispielsweise den Sensor als Argument erhält, wäre nutzlos, da es sich bei dem Sensor, welcher der Observer erhalten würde, um eine Schnittstelle handeln würde, um den konkreten Sensor zu erhalten, wäre daher ein casting notwendig. Dies entfällt durch das vorgängige Binden des Funktors, beispielsweise kann eine Instanz einer `std::function<void()>` mittels einer Closure oder einer gebundenen Member Funktion erzeugt werden.

Aus Platzgründen wurden nicht alle Attribute und Methoden der Klasse Application notiert, generell handelt es sich um eine grobe Sicht auf die Architektur. Eine zentrale Rolle spielt der Konstruktor der Klasse, darin werden die Views beiden Sensoren als Beobachter

registriert und die Anzeige mit den aktuellen Werten versehen. Bei einem Wechsel der Anzeige können schliesslich die registrierten Beobachter deregistriert werden um die IP Adresse oder im Fehlerfall die Fehlermeldung anzuzeigen. Eine Alternative zu dem erwähnten Modell, in welchem die Views auf Änderungen am Modell lauschen, besteht darin, dass nicht die Views direkt auf Änderungen am Display lauschen sondern Application. Dies kann realisiert werden, indem wie angedeutet, eine Methode `onValueChanged` als Grundlage dient, einen Observer zu erstellen. Die Methode kann mit `this` und der zugehörigen View gebunden werden, um einen Funktor zu erzeugen, welcher als Beobachter eines Sensors dient. Ana- log dazu muss schliesslich noch dasselbe für den Thread `dbWriter` getan werden, indem statt bloss den Observer mit dem Sensor zu speichern, zusätzlich noch ein Funktor gespeichert wird, welcher den zugehörigen Wert in Values setzt. Schliesslich kann der Thread die Map durchwandern, jeden zugehörigen Funktor mit einem Value aufrufen, den Zeit Stempel setzen und die so assemblierten Werte mittels der Datenbank speichern. Im Falle eines Wechsels der Anzeige werden jeweils die Beobachter bei den Sensoren an- resp. abgemeldet.

In der Abbildung 12 auf der nächsten Seite wird das Klassendiagramm der Steuerung dargestellt.

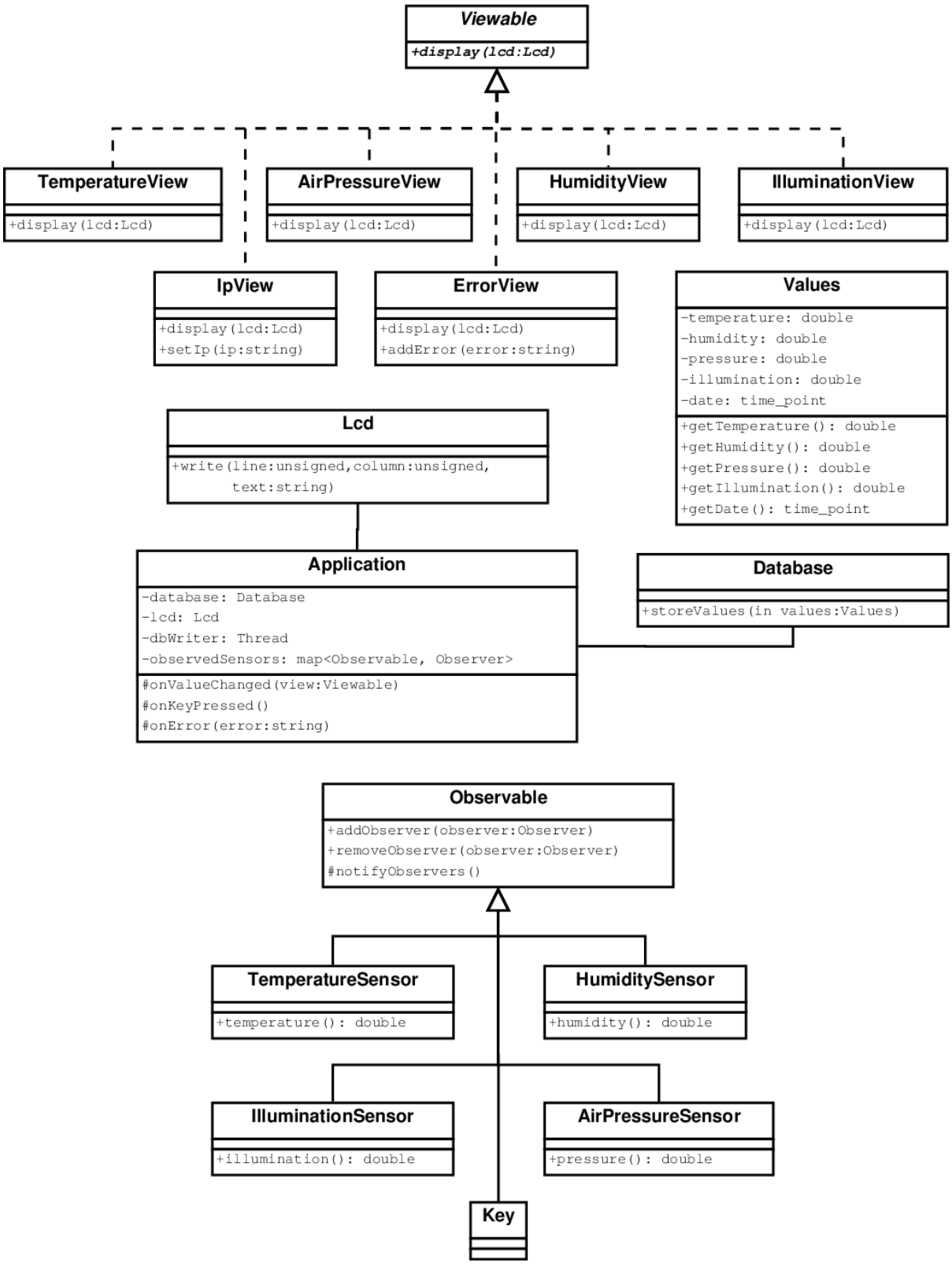

Abbildung 12: Die Messwerte werden periodisch mittels Polling gespeichert

8.3 Datenbank

Eine Sammlung von Messwerten wird in der Datenbank abgelegt, zu welcher folglich ein entsprechendes Schema folgende Felder umfasst (ohne Primärschlüssel):

Feldbezeichnung (Tabelle Messwerte)	Datentyp
Id	Integer
Created_At (ISO Darstellung YYYY-MM-DD HH:MM:SSS.Z)	Text
Temperature	Real
Humidity	Real
Pressure	Real
Illumination	Real

Das Auslesen der Messwerte aus der Datenbank wird durch eine andere Anwendung durchgeführt (Web-Anwendung), daher muss die Abstraktion, welche hier eingesetzt wird, dies vorerst nicht unterstützen. Die Daten werden in einer SQLite Datenbank abgelegt, auf diese Weise muss kein separater Datenbankserver betrieben werden und eine externe Anwendung (Web Anwendung) kann ebenfalls mittels einem standardisierten Verfahren (SQL) auf die Daten zugreifen.

8.4 Webseite

8.4.1 Benutzeroberfläche

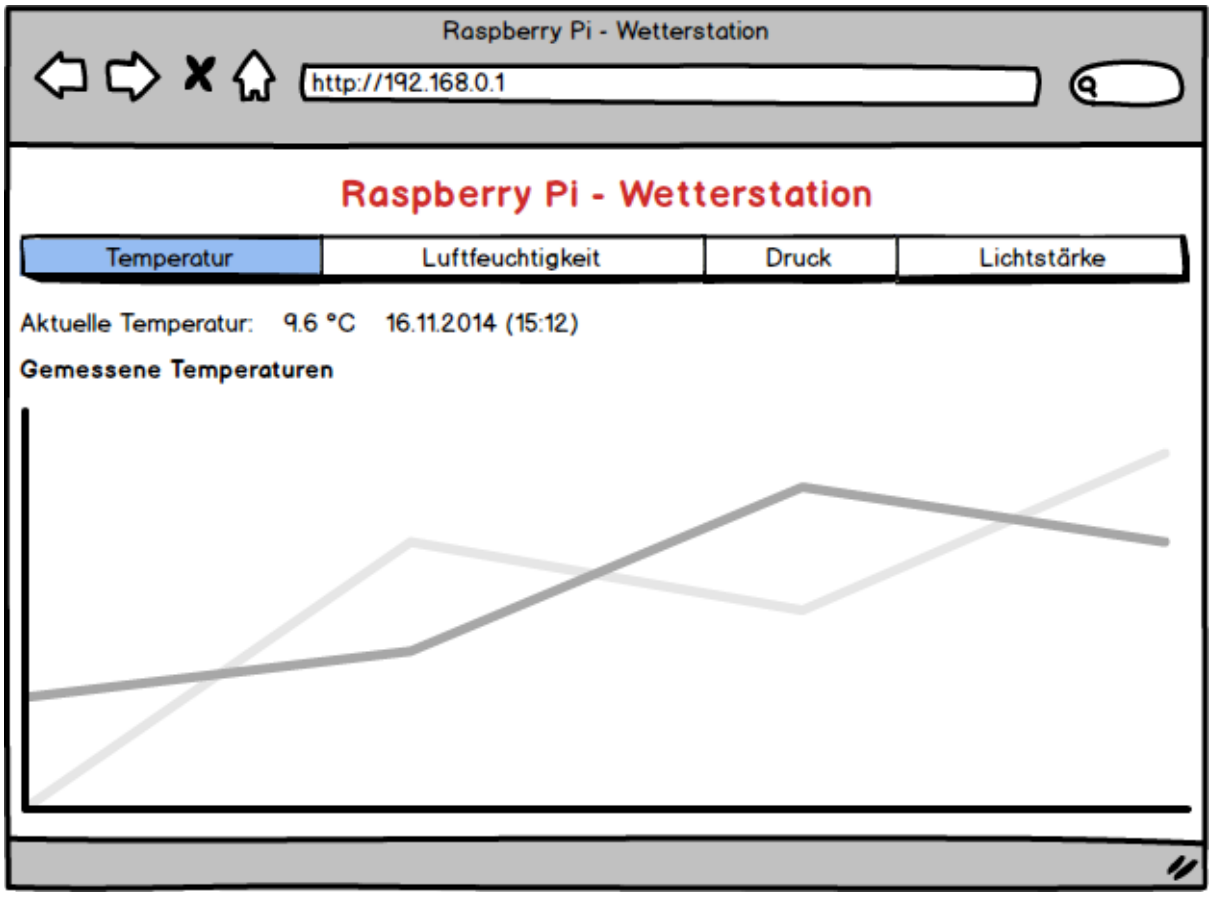

Abbildung 13: Benutzeroberfläche der Web-Anwendung

Auf der Benutzeroberfläche kann zwischen den unterschiedlichen Sensordaten navigiert werden. Auf der entsprechenden Seite wird der aktuelle (zuletzt gemessene) Datensatz dargestellt, sowie ein Verlauf der gemessenen Werte der vergangenen 7 Tage.

8.4.2 Software Entwurf

Im der nachfolgenden Abbildung wird das Design der Web-Anwendung schematisch dargestellt.

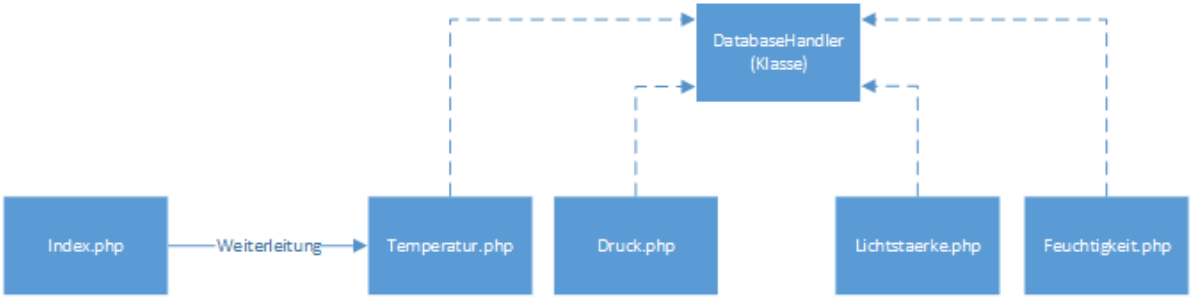

Abbildung 13: Benutzeroberfläche der Web-Anwendung

Es werden 5 PHP Seiten erstellt (Index, Temperatur, Druck, Lichtstaerke und Feuchtigkeit) wobei Index nur zur Weiterleitung auf die Seite der Temperatur verwendet wird. Für den Zugriff auf die SQLite Datenbank (zur Abfrage der Messwerte) wird eine separate Klasse erstellt (DatabaseHandler) welche durch alle Seiten mittels einem Include verwendet werden kann.

Folgende Komponenten werden für den Betrieb der Webseite benötigt:

- Apache Webserver mit PHP 5 Bibliothek
- JPGraph Bibliothek (Diagrammerstellung mit PHP)
- SQLite 3

9 Qualitätssicherung

Um den notwendigen Qualität Standard zu ermitteln, werden die funktionalen Anforderungen herbeigezogen um daraus entsprechenden Test Szenarien abzuleiten.

- Luftdruck, Temperatur, Feuchtigkeit und Lichtstärke müssen korrekt ermittelt, dargestellt und gespeichert werden.
    - Mittels manuellen Tests und einem Referenz Gerät kann verifiziert wer- den, ob die Werte in einem gültigen Toleranzbereich liegen. Falls dies über das LCD verifiziert werden kann, ist auch sichergestellt, dass die Werte entsprechend ermittelt werden.
    - Mittels manueller Überprüfung kann verifiziert werden, ob der entsprechende Inhalt in der Datenbank abgelegt wurde. Beispielsweise mittels einer SQLite Shell.
- Sämtliche Informationen müssen über eine Web Applikation eingesehen wer- den können.
    - Auch dies kann mittels manueller Überprüfung auf dieselbe Weise verifiziert werden: Sämtliche online dargestellten Werte müssen im Vergleich mit den Werten eines Referenz Gerätes in einem gültigen Toleranzbereich liegen. Zudem kann überprüft werden, ob zuvor gespeicherte Daten abrufbar sind.

Zudem müssen weitere Risiken berücksichtigt werden, wobei externe Fehlerquellen wie ein Ausfall der Stromzufuhr ignoriert werden, solche Fehler können nicht adressiert werden. Ein Ausfall des LCD oder ein festsitzender Schalter können ebenfalls nicht behandelt werden. Es sollte jedoch sichergestellt werden, dass keine falschen Werte gespeichert werden, falls ein Sensor ausfällt, sollte dies von der Anwendung erkannt werden.

Zudem ist es möglich, dass irgendwann der Speicher ausgeht. Es wäre auch denk- bar, dass eine sehr grosse Datenbank Probleme verursacht. Es bleiben zusätzlich die folgenden möglichen Fehler, welche behandelt werden müssen:

- Ausfall von Sensoren
    - Im Falle von ausgefallenen Sensoren sollen die entsprechenden Werte nicht gespeichert werden. Auf dem LCD soll in diesem Fall eine Fehlermeldung angezeigt werden.
- Kein Speicherplatz mehr verfügbar
    - Falls das Speichern der Messwerte fehlschlägt, weil kein Speicherplatz mehr verfügbar ist, soll auf dem LCD ebenfalls eine Fehlermeldung angezeigt werden.
- Kein Netzwerk verfügbar
    - In diesem Fall soll bei einem Wechsel der Anzeige zur IP Adresse eine Fehlermeldung angezeigt werden.

10 Testing

Um den korrekten Betrieb des Systems zu verifizieren, werden Testfälle definiert, welche im Anschluss an die Implementierung und auch währenddessen durchlaufen werden. Die benötigten Testfälle können anhand der zuvor ermittelten Kriterien und Risiken erstellt werden. Folgende Testfälle wurden definiert:

- Es wird geprüft, ob die korrekten Werte ermittelt werden.
- Es wird geprüft, ob die Speicherung der gemessenen Daten funktioniert
- Es wird geprüft, ob die aktuellen Messwerte über die Web Anwendung abruf- bar sind.
- Es wird geprüft, ob die zuvor gespeicherten Messdaten über die Web Anwendung abrufbar sind.
- Es wird geprüft, ob sich das System gemäss den Anforderungen verhält, wenn alle Sensoren ausfallen.
- Es wird geprüft, ob sich das System gemäss den Anforderungen verhält, wenn ein Sensor ausfällt.
- Es wird geprüft, ob sich das System gemäss den Anforderungen verhält, wenn kein weiterer Speicherplatz mehr verfügbar ist.

Neben den hier aufgeführten, eher technischen, Testfällen müssen beim Testen zusätzlich die in Kapitel 6 definierten Use Cases nach Abschluss der Entwicklung (als Abnahmetests) getestet werden. Sollten ein oder mehrere Use Cases durch die hier definierten Testfälle abgedeckt werden entfällt der entsprechende Use Case Test.

10.1 Testfälle

TC 1 - Ermitteln der Werte	
Beschreibung	Das korrekte Ermitteln der Werte der Sensoren wird manuell überprüft. Ein Referenz Gerät liefert die Werte, mit welchen die ermittelten Werte verglichen werden. Die Differenz muss anschliessend im gültigen Toleranzbereich liegen.

Vorbedingungen	• Ein beliebiges, handelsübliches Gerät als Referenz Gerät
	• Definition der Toleranzbereiche
	  ○ Feuchtigkeit: 2%
	  ○ Temperatur: 0.7 °C
	  ○ Lichtstärke: 0.4 lx
	  ○ Luftdruck: 1.4 hPa

Ablauf	Die Messwerte werden vom Display des Gerätes abgelesen und mit den Werten des Referenz Gerätes verglichen. Die Differenz der beiden Messdaten wird für jeden gemessenen Werte berechnet.
Auswertung	Die berechnete Differenz eines jeden Messwertes muss im Toleranzbereich liegen.

TC 2 - Speicherung	
Beschreibung	Mittels einer SQLite Shell wird geprüft, ob der periodisch gespeicherte Inhalt in der Datenbank vorhanden ist.
Vorbedingungen	Die Anwendung befindet sich in laufendem Betrieb und es wurden gemäss der bereits verstrichenen Zeit Datensätze angelegt.
Ablauf	Nachdem die Anwendung bereits so lange läuft, dass mind. 3 Datensätze angelegt wurden, wird mittels Shell eine Verbindung zu der SQLite Datenbank hergestellt. Es wird geprüft, dass die erwarteten Datensätze vorhanden sind.
Auswertung	Die erwarteten Datensätze müssen mit den korrekten Zeitstempeln existieren.

TC 3 - Aktuelle Daten über die Webapplikation	
Beschreibung	Mittels manuellem Testing wird verifiziert, dass die von der Web Applikation dargestellten Werte im gültigen Toleranzbereich liegen.
Vorbedingungen	Die Web Applikation ist über den Browser erreichbar.
Ablauf	Die Web Anwendung wird mit dem Browser geöffnet. Es werden die aktuellen Werte über die Web-Oberfläche mit den Werten des Referenz Gerätes verglichen. Zu den Werten des Referenz Gerätes wird zu jeder gemessenen Grösse die Differenz berechnet.
Auswertung	Die angezeigten Werte müssen im Toleranzbereich von TC1 liegen.

TC 4 - Gespeicherte Daten über die Webapplikation	
Beschreibung	Mittels manuellem Testing wird verifiziert, dass die zuvor gespeicherten Werte auch über die Web-Oberfläche verfügbar sind.
Vorbedingungen	Die Web Applikation ist über den Browser erreichbar.
Ablauf	Die Web Anwendung wird mit dem Browser geöffnet. Es werden die gespeicherten Werte über die Web-Oberfläche abgefragt. Parallel dazu werden die gespeicherten Werte über eine SQLite Shell ermittelt.
Auswertung	Die angezeigten Werte müssen mit denjenigen in der SQLite Datenbank übereinstimmen.

TC 5 - Ausfall aller Sensoren	
Beschreibung	Während des Betriebs wird das Verbindungskabel zwischen dem Raspberry Pi und der Wetterstation getrennt. Auf der Anzeige der Werte der Sensoren muss eine entsprechende Fehlermeldung angezeigt werden.
Vorbedingungen	Der Dämon Prozess wurde gestartet und das LCD zeigt die gemessenen Werte an.
Ablauf	Das Verbindungskabel zum Bricklet wird entfernt. Es wird überprüft, ob auf dem Display eine entsprechende Fehlermeldung angezeigt wird.
Auswertung	Es wird geprüft, ob eine entsprechende Fehlermeldung angezeigt wird.

TC 6 - Ausfall eines Sensors	
Beschreibung	Während des Betriebs wird ein Verbindungskabel zwischen dem Master und einem Sensor getrennt. Auf der Anzeige der Werte der Sensoren muss eine entsprechende Fehlermeldung angezeigt werden.
Vorbedingungen	Der Dämon Prozess wurde gestartet und das LCD zeigt die gemessenen Werte an.
Ablauf	Das Verbindungskabel zum Sensor wird entfernt. Es wird überprüft, ob auf dem Display eine entsprechende Fehlermeldung angezeigt wird.
Auswertung	Es wird geprüft, ob eine entsprechende Fehlermeldung angezeigt wird.

TC 7 - Speicherplatz	
Beschreibung	Es wird eine ausreichend grosse Datei auf eine separate, für die-sen Test präparierte SD Karte gelegt. Die Datei muss so gross sein, dass bloss noch wenige Bytes an Speicher auf der Karte vorhanden sind. Eine solche Datei kann mittels dd erstellt wer- den. Anschliessend wird verifiziert, dass die Anwendung versucht, die Datensätze abzulegen. Sobald dies fehlschlägt, muss eine entsprechende Fehlermeldung angezeigt werden.
Vorbedingungen	Für den Test wird eine übergrosse Datei auf dem Dateisystem abgelegt. Die Datei weist eine Grösse auf, welche bis auf wenige Bytes dem noch verfügbaren Speicher der Karte entspricht. So- mit weist die verwendete SD Karte bloss noch wenig Speicherplatz auf.
Ablauf	Die Anwendung wird gestartet und es wird beobachtet, wie der verfügbare Speicher auf der Karte abnimmt. Sobald kein verfügbarer Speicher mehr vorhanden ist, wird geprüft, ob auf dem LCD eine entsprechende Fehlermeldung angezeigt wird.
Auswertung	Es wird geprüft, ob eine entsprechende Fehlermeldung angezeigt wird.