Anforderungsspezifikation

Inhaltsverzeichnis

Vision T15-Fernabfrage von Kamerabild und Wetterdaten

Alle 14.08.2020

1. Einführung

Der Zweck dieses Dokuments ist es, die wesentlichen Bedarfe und Funktionalitäten der Wetterstation-Software zu sammeln, zu analysieren und zu definieren. Der Fokus liegt auf den Fähigkeiten, die von Stakeholdern und adressierten Nutzern benötigt werden, und der Begründung dieser Bedarfe. Die Details, wie die Wetterstation-Software diese Bedarfe erfüllt, werden in der Use-Case und Supplementary Specification beschrieben.

1.1. Zweck

Der Zweck dieses Dokuments ist es, die wesentlichen Anforderungen an das System aus Sicht und mit den Begriffen der künftigen Anwender zu beschreiben.

1.2. Gültigkeitsbereich (Scope)

Dieses Visions-Dokument bezieht sich auf das Projekt "Wetterstation", das von Team "Wetter" entwickelt wird. Das System wird es erlauben, eingelesene Wetterdaten zu sammeln, lokal zu speichern und an den Webserver des mfcR zu senden. Dort können die Wetterdaten dann von jedem abgerufen werden.

1.3. Definitionen, Akronyme und Abkürzungen

siehe Glossary

1.4. Referenzen

Architecture Notebook, Systemwide requirements

2. Positionierung

2.1. Fachliche Motivation

Der Modellflugclub Rossendorf e.V. wünscht sich eine vereinseigene Wetterstation mit einer

Webcam, wobei diese Daten übermittelt und auf der Vereinswebseite angezeigt werden sollen. Zweck des Systems ist die Daten für alle Nutzer des Modellflugplatzes bzw. Vereinsmitglieder bereitzustellen. Nutzer können somit nachvollziehen wie die aktuelle Wetterlage ist und wer sich auf dem Flugplatz befindet (ggf. zum Schutz des Eigentums).

2.2. Problem Statement

Das Problem	Dem mfcR stehen keine aktuellen und örtlich genauen Wetterdaten + Bilder zur Verfügung.
betrifft	die Nutzer des Modellflugplatzes des mfcR
die Auswirkung davon ist	Schlechte Einschätzung der Wetterlage auf dem Flugplatz und daraus resultierende Ungewissheit ob sich ein Besuch des Flugplatzes lohnt.
eine erfolgreiche Lösung wäre	Nutzer können schnell aktuelle Wetterdaten des Platzes abrufen, sowie sich ein Bild von der Lage machen.

2.3. Positionierung des Produkts

Für	Nutzer der Webseite/ des Modellflugplatzes
die	die aktuellen Wetterdaten abfragen wollen
Das Produkt / die Lösung ist eine	Webanwendung auf der Vereinswebseite
Die	die aktuellen (Wetter)Verhältnisse auf dem Flugplatz zeigen
Im Gegensatz zur	jetzigen Webseite
Unser Produkt	zeigt aktuelle und örtlich genaue Informationen an.

3. Stakeholder Beschreibungen

3.1. Zusammenfassung der Stakeholder

Name	Beschreibung	Verantwortlichkeiten
Thomas Brenner	Auftraggeber, Vorsitzender des Vereins MODELLFLUGCLUB ROSSENDORF e.V.	Ansprechpartner, überwacht den Projektfortschritt, verwaltet den Raspberry Pi und die Vereinswebsite
Professor Heino Iwe	Auftraggeber	Ansprechpartner, überwacht den Projektfortschritt

Name	Beschreibung	Verantwortlichkeiten
Administratoren	Berechtigte für den internen Bereich	Informationen zum Systemzustand abfragen
User	Besucher der Webseite	aktuelle Informationen und gespeicherte Bilder abrufen
Gesetzgeber	geltende Gesetze und Richtlinien der BRD	Bildrechte
jweiland.net	Hosting-Dienstleister des mfcR	Bereitstellung der physischen webserverseitigen Infrastruktur

3.2. Benutzerumgebung

1. Beschreiben Sie die Arbeitsumgebung des Nutzers:
    - Der Nutzer muss auf einem internetfähigem Gerät in einem Browser online sein. Die Bedienung ist auf unterschiedlichen Endgeräten möglich.
2. Welche anderen Anwendungen sind im Einsatz? Muss ihre Anwendung mit diesen integriert werden?
    - TYPO3-CMS auf dem die Vereinswebsite beruht
    - Anwendung soll in Vereinswebsite integriert werden

4. Produkt-/Lösungsüberblick

4.1. Bedarfe und Hauptfunktionen

Bedarf	Priorität	Features	Geplantes Release
Bilder abrufen	mittel	zeitlich gesteuerte Aktualisierung der Bildergalerie mit Auswahl nach Zeitstempel, Vollbildmöglichkeit	31.07.2020
Diagramme abrufen	hoch	Temperatur, Wind (Stärke+Richtung), nach Zeitstempel abrufbar	31.07.2020
Bilder und gemessene Wetterdaten speichern	hoch	in Datenbank und lokal	31.07.2020
interner Administrationsbereich	hoch	Informationen über Systemzustand	31.07.2020

5. Zusätzliche Produktanforderungen

äußere Faktoren:

- keine Infrastruktur vorhanden
  - somit kein Strom und kabelgebundenes Internet am Modellflugplatz
- System wird nach Fertigstellung im Boden eingegraben → keine leichte Wartung möglich → Remote-Zugriff + hohe Stabilität wichtig

eingesetzte Hardware:

- Raspberry Pi 4
- diversere Sensoren (via I2C/SPI)
- Webcam (via Raspi on-board Camera Connector)
- UMTS-Modul (via USB)
- Akku (LiPo)
- Solarzelle
- Lade-Management
- Gehäuse

Software (Raspi/Webserver)

- Verwendung einer Skriptsprache (Python)
- Zugang zu Webserver wird gestellt (10 GB)
- Zugang zu einer mySQL-DB wird gestellt
- Verlinkung durch Verneinswebseite

Anforderung	**Priorität**	**Geplantes Release**
Verwendung einer Skriptsprache	mittel	16.03.2020
Anwenderdokumentation	hoch	31.07.2020
Betriebsdokumentation	hoch	31.07.2020
Projektbericht	hoch	12.08.2020
Entwicklerdokumentation	hoch	12.08.2020
Testdokumentation	hoch	12.08.2020
Anforderungsspezifikation	hoch	12.08.2020
Glossar	hoch	12.08.2020

Wetterstation System-Wide Requirements Specification

Hannes Fogut <s76919@htw-dresden.de>; Josefin Hähne <s78927@htw-dresden.de>; Philipp Barth <s78830@htw-dresden.de>; Justin Schirdewahn <s78878@htw-dresden.de>; Alexander Schoch <s78797@htw-dresden.de>; Agustin Calvimontes <s78927@htw-dresden.de>; Clemens Kujus <s78802@htw-dresden.de>;

1. Einführung

In diesem Dokument werden die systemweiten Anforderungen für das Projekt **Wetterstation** spezifiziert. Die Gliederung erfolgt nach der FURPS+ Anforderungsklassifikation:

- Systemweite funktionale Anforderungen (F),
- Qualitätsanforderungen für Benutzbarkeit, Zuverlässigkeit, Effizienz und Wartbarkeit (URPS) sowie
- zusätzliche Anforderungen (+) für technische, rechtliche, organisatorische Randbedingungen

Die funktionalen Anforderungen, die sich aus der Interaktion von Nutzern mit dem System ergeben, sind als Use Cases in separaten Dokumenten festgehalten.

2. Systemweite funktionale Anforderungen

- F01: Es werden jede Minute neue Wetterdaten gemessen
- F02: Die Übertragung der Daten geschieht zyklisch
- F03: Admins haben eine Berechtigung um sich im internen Bereich anzumelden
- F04: Änderungen von Daten können eingesehen werden
- F05: Die Übertragung der Daten geschieht über https

3. Qualitätsanforderungen für das Gesamtsystem

3.1. Benutzbarkeit (Usability)

- U01: Der Nutzer kann sich in wenigen Schritten (max. 5 Klicks) ein Diagramm anzeigen lassen und gelangt mit maximal 2 Klicks zur Bildergalerie
- U02: Der Nutzer benötigt keine weiteren Vorkenntnisse um das System zu benutzen (auch für Gelegenheitsnutzer zu verstehen)
- U03: Fehlertexte werden verständlich angezeigt

- U04: Die Sprache für die Benutzeroberfläche ist deutsch

3.2. Zuverlässigkeit (Reliability)

- R01: Die Wetterdaten werden genau angezeigt
- R02: Wenn ein Fehler auftritt werden die Daten nach der Fehlerbehebung wieder aus der Datenbank gezogen

3.3. Effizienz (Performance)

- P01: Die Bilder bzw. Diagramme werden innerhalb von 10 Sekunden angezeigt
- P02: Es können mindestens 2 Nutzer parallel auf das System zugreifen
- P03: Es werden mindestens zweimal pro Stunde Daten an den Server geschickt

3.4. Wartbarkeit (Supportability)

- S01: Das System muss in die Vereinswebsite eingebunden werden können
- S02: Das System muss auch auf mobilen Oberflächen laufen (z.B. Smartphones)
- S03: Das System soll auf Linux installierbar sein

4. Zusätzliche Anforderungen

4.1. Einschränkungen

- C01: zu nutzende Komponenten: mySQL-Datenbank, Webapplikation
- C02: Vorgaben für die Programmiersprache auf Raspi: Python
- C03: Hardwareeinschränkungen durch Raspi

4.2. Organisatorische Randbedingungen

- I01: Logo des mfcR soll auf der Oberfläche sichtbar sein

4.3. Rechtliche Anforderungen

- BR01: Datenschutz (personenbezogene Daten auf Bildern)

Use-Case Model Projekt Wetterstation

Hannes Fogut <s76919@htw-dresden.de>; Josefin Hähne <s78927@htw-dresden.de>; Philipp Barth

<s78830@htw-dresden.de>; Justin Schirdewahn <s78878@htw-dresden.de>; Alexander Schoch <s78797@htw-dresden.de>; Agustin Calvimontes <s78927@htw-dresden.de>; Clemens Kujus <s78802@htw-dresden.de>; 14.08.2020

Use-Cases

- Bildergalerie abrufen → User, Administratoren
- Daten aufrufen → User, Administratoren
- Diagramm abrufen → User, Admininstratoren
- Daten ändern → Administratoren
- logfile lesen → Administratoren

Priorisierung:

1. Diagramm abrufen
2. Bildergalerie abrufen
3. Daten aufrufen
4. Daten ändern
5. logfile lesen

Use Case Diagramm

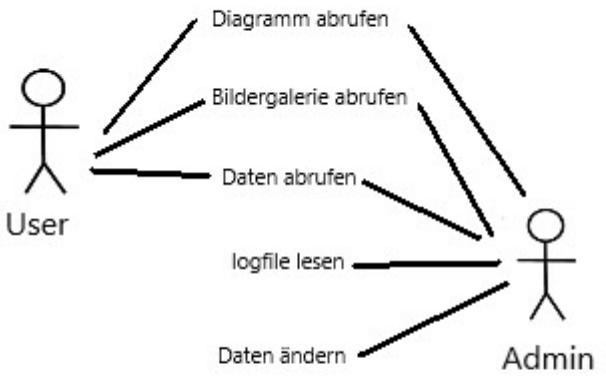

Figure 1. Use-Case Diagramm

1. Use Case: Bildergalerie abrufen

1.1. Kurzbeschreibung

In diesem Use-Case wird die Anzeige und das Bearbeiten der aufgenommenen Bilder behandelt.

1.2. Kurzbeschreibung der Akteure

1.2.1. User

User können sich die Bilder anschauen, welche in einer Galerie auf der Website des MFCR zur Verfügung gestellt werden.

1.2.2. Admins

Admins können sich diese Bilder ebenfalls anzeigen lassen, haben aber erweiterte Rechte, um Bilder zu bearbeiten, zu kopieren, zu verschieben oder zu löschen.

1.3. Vorbedingungen

1. Die Kamera hat bereits Bilder aufgenommen
2. Die Bilder werden korrekt auf den Server gepusht und dazugehörige Meta-Daten in der DB gespeichert
3. Die Bilder werden vom Server richtig zum Client übertragen

1.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der User/Admin die Website mit der Bildergalerie aufruft
2. Die Bilder werden in aufgenommener Reihenfolge als Galerie angezeigt, das aktuellste Bild wird größer dargestellt
3. Beim Klicken auf ein anderes Bild wird die Auswahl an die Bildschirmgröße des Nutzers angepasst dargestellt
4. Die Bilder laden nach unten weiter, wenn der User weiter nach unten scrollt
5. Der User/Admin kann einen Zeitraum angeben, in dem die aufgezeichneten Bilder angezeigt werden sollen
6. Der Use Case ist abgeschlossen.

1.5. Alternative Abläufe

1.5.1. Bildfehler

Wenn der User/Admin im Schritt 1 des Standardablauf die Website aufruft, dann kann es sein, dass

ein oder mehrere Bilder nicht richtig verarbeitet wurden

1. Der Admin loggt sich ein und kann die entsprechenden Bilder bearbeiten oder löschen.
2. Der Use Case wird im Schritt 1 des Standartablaufes fortgesetzt.

1.5.2. Zeitraumfehler

Wenn der User/Admin im Schritt 4 des Standardablauf einen ungültigen Zeitraum angibt, dann wird er gebeten den Zeitraum zu ändern.

1. Der User/Admin gibt einen Zeitraum an, in dem keine Bilder aufgezeichnet wurden.
2. Der Server greift auf die Datenbank zu und stellt fest, dass es keine Bilder im angegebenen Zeitraum gibt.
3. Dem User/Admin wird die Meldung angezeigt, dass er den Zeitraum ändern soll und wann das erste und das letzte Bild aufgezeichnet wurden.
4. Der Use Case wird im Schritt 4 Standardablaufes fortgesetzt.

1.6. Unterabläufe (subflows)

1.6.1. Zeiträume

1. Der User/Admin kann einen Zeitraum angeben, um nach Bildern in diesem zu filtern
2. Der Server greift auf die Datenbank zu
3. Die Website generiert die Galerie dem entsprechenden Zeitraum nach neu
4. Der Subflow ist abgeschlossen

1.7. Wesentliche Szenarios

1.7.1. Bilder bearbeiten/löschen/kopieren

Wenn der Admin sich eingeloggt hat, hat er die Möglichkeit in Schritt 1 die Bilder von der Website zu löschen oder diese herunterzuladen, um sie lokal zu bearbeiten oder zu sichern.

1.8. Nachbedingungen

1.8.1. Galerieanzeige

Die Website hat die Galerie entsprechend der User/Admin-Anforderungen geladen und wartet auf neue Änderungen (weitere Bilder laden, Zeitraumänderung, Bilder löschen), um die Galerie dementsprechend neu zu generieren

1.9. Besondere Anforderungen

1.9.1. Qualitätsanforderung

Sollte ein Admin ein Bild löschen, muss die Website die Galerie ohne dieses Bild weiterhin ordnungsgemäß und lückenlos anzeigen können.

1.10. Wireframes

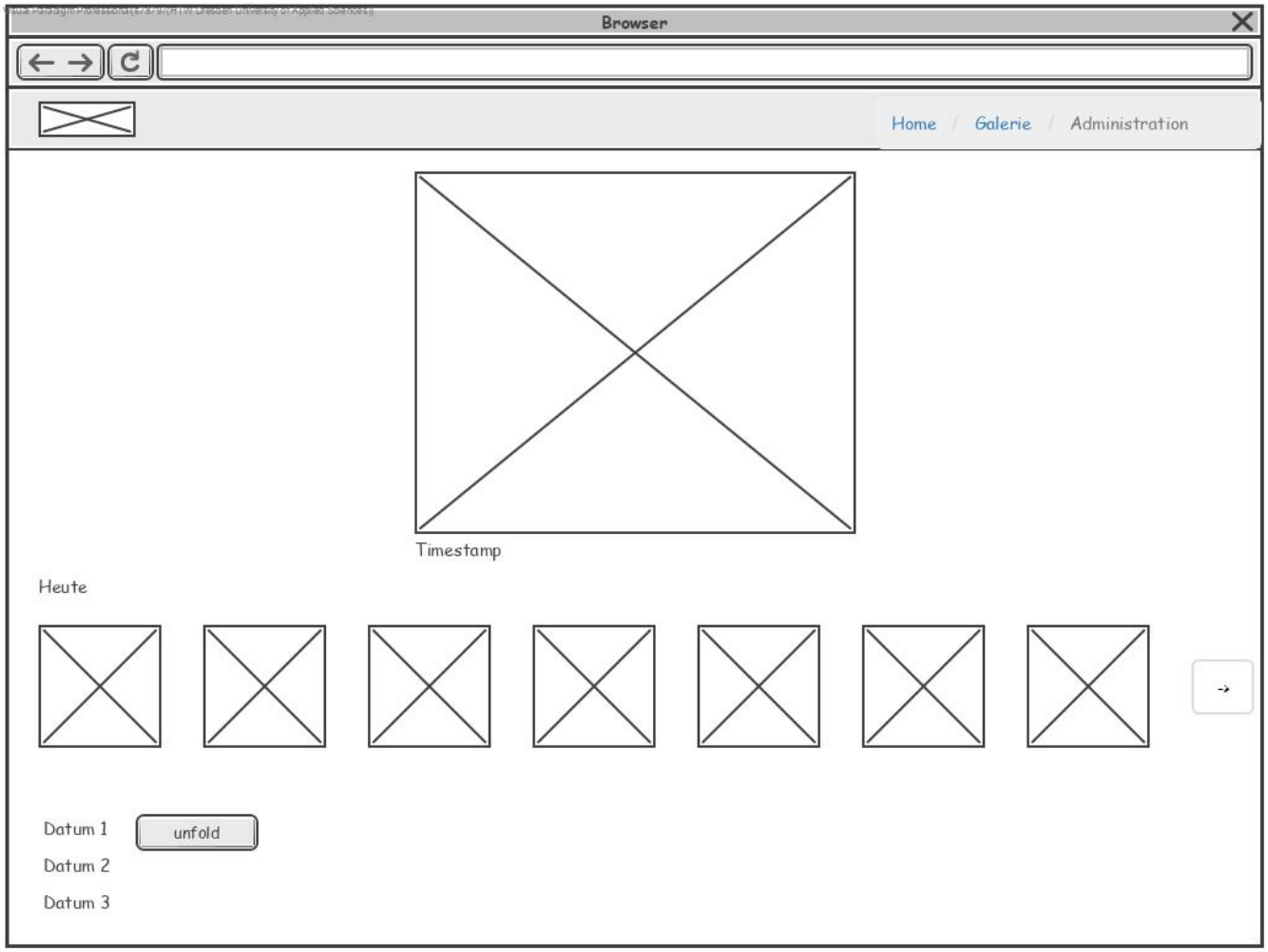

Figure 2. Wireframe Galerie 1

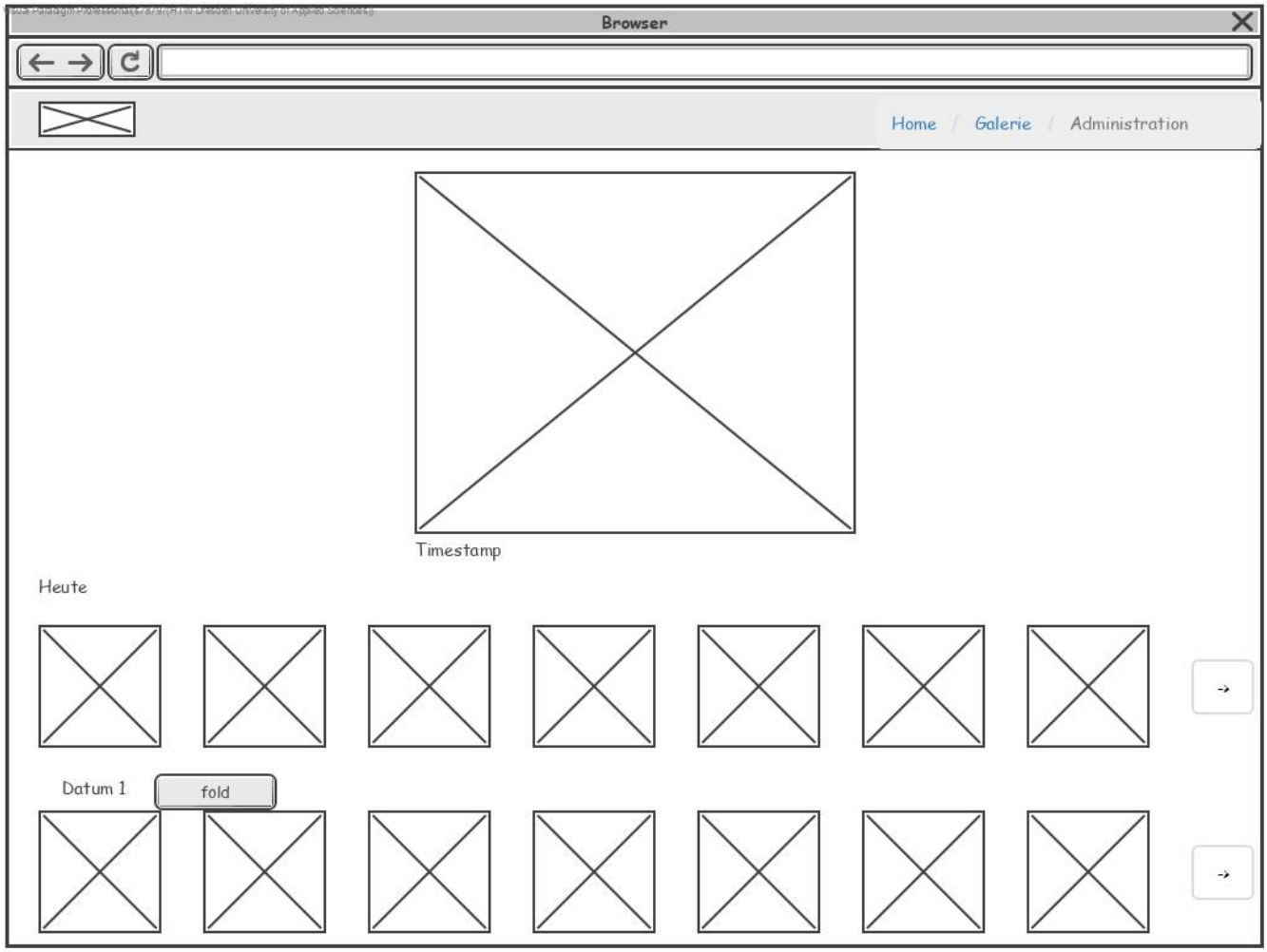

Figure 3. Wireframe Galerie 2

2. Use Case: Daten aufrufen

2.1. Kurzbeschreibung

Es geht um die Anzeige der gesammelten Daten, die Funktionen, die durch die Anzeige gewährleistet werden sollen und mögliche Schwierigkeiten.

2.2. Kurzbeschreibung der Akteure

2.2.1. User

User können sich die Daten aufrufen, welche auf der Website des MFCR zur Verfügung gestellt werden.

2.2.2. Admins

Admins können sich diese Daten ebenfalls anzeigen lassen.

2.3. Vorbedingungen

- Daten werden gesammelt
- Daten werden in Datenbank ordnungsgemäß gespeichert
- Daten werden vom Server richtig verarbeitet

2.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn ein User oder ein Admin die gesammelten Daten aufrufen und auswerten möchte
2. Der User/Admin ruft die Seite auf, in der die Daten angezeigt werden sollen
3. Die gesammelten Daten werden als Diagramm angezeigt
4. Der User/Admin wählt die Datenart aus, die er sehen möchte
5. Der User/Admin wählt den Zeitraum aus in dem die gesammelten Daten angezeigt werden sollen
6. Der Server greift auf die Datenbank, in der die Daten gespeichert werden, zu und nutzt die Daten, die angegeben wurden
7. Dem User/Admin wird ein Diagramm aus den gewählten Daten angezeigt
8. Der Use Case ist abgeschlossen.

2.5. Alternative Abläufe

2.5.1. Zeitraum ohne Daten

Wenn der User/Admin im Schritt 7 des Standardablauf die Datenart oder den Zeitraum ändert, dann wird das Diagramm neu generiert . Der User/Admin wählt einen Zeitraum, in dem keine Daten aufgezeichnet wurden . Der Server zeigt dem User/Admin, dass es zu diesem Zeitraum keine Daten gibt . Der Use Case wird im Schritt 5 fortgesetzt.

2.6. Unterabläufe (subflows)

2.6.1. Datenartänderung

1. Der User/Admin wählt eine andere Datenart
2. Der Server greift erneut auf die Datenbank zu, um die erforderlichen Daten zu erhalten
3. Der Subflow wird in Schritt 5 des Use Cases fortgesetzt

2.6.2. Zeitraumänderung

1. Der User/Admin wählt einen anderen Zeitraum für die momentan ausgewählte Datenart
2. Der Server greift erneut auf die Datenbank zu, um die erforderlichen Daten zu erhalten

3. Der Use Case wird in Schritt 6 des Use Cases fortgesetzt

2.7. Nachbedingungen

2.7.1. Diagramm

Der Server zeigt dem User/Admin ein Diagramm, was auf den von ihm/ihr ausgewählten Daten und dem ausgewählten Zeitraum generiert wurde, um eine schnelle und einfache Auswertung der aufgezeichneten Daten zu ermöglichen.

2.8. Wireframes

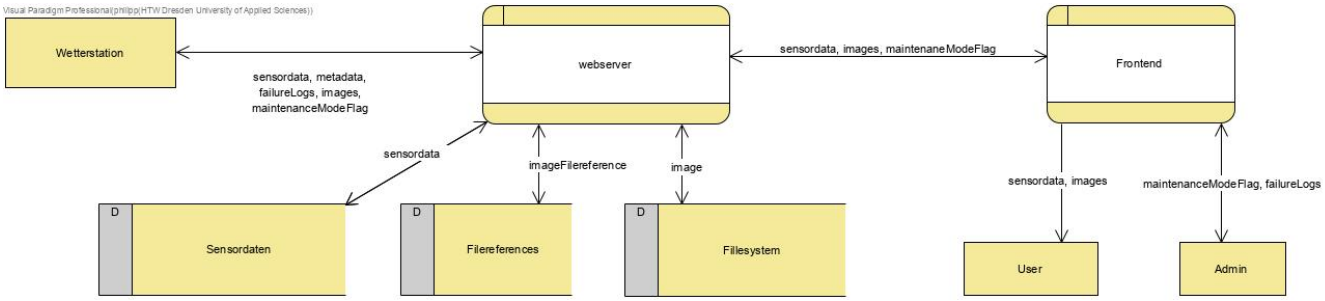

Figure 4. Data Flow Diagramm

3. Use Case: Diagramm abrufen

3.1. Kurzbeschreibung

Dieser Use Case behandelt das Diagramm, welches auf der Website aus den gesammelten Daten erstellt werden soll und von den Nutzern der Website abgerufen werden soll.

3.2. Kurzbeschreibung der Akteure

3.2.1. User

Die normalen User, die sich die Daten der Wetterstation als Diagramm anzeigen lassen wollen.

3.2.2. Admins

Die Admins, die sich die Daten der Wetterstation anzeigen lassen wollen.

3.3. Vorbedingungen

1. Raspberry Pi hat valide Sensordaten gesammelt
2. Raspberry Pi übermittelt Daten an Webserver
3. Daten werden richtig in Datenbank gespeichert
4. Daten werden vom Webserver richtig aufgerufen

5. Daten werden ordnungsgemäß verarbeitet

3.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn User/Admins die Website aufrufen
2. Der User/Admin wählt eine Datenart aus
3. Der User/Admin wählt einen Zeitraum
4. Die Website greift auf die Datenbank zu und zieht sich die entsprechenden Daten aus der Datenbank
5. Die Website generiert das Diagramm den Daten entsprechend
6. Der Use Case ist abgeschlossen.

3.5. Alternative Abläufe

3.5.1. Zeitraumfehler

Wenn der User/Admin im Schritt 3 des Standardablauf einen Zeitraum wählt in dem keine Daten aufgezeichnet wurden, dann wird der User/Admin darum gebeten einen anderen Zeitraum auszuwählen

1. Der User/Admin wählt einen Zeitraum
2. Der Server versucht auf die Datenbank zuzugreifen und stellt fest, dass im angegebenen Zeitraum keine Daten aufgezeichnet wurden
3. Dem User/Admin wird angezeigt, dass er bitte einen anderen Zeitraum auszuwählen soll
4. Der Use Case wird im Schritt 3 fortgesetzt.

3.6. Unterabläufe (subflows)

3.6.1. Andere Datenart

1. Der User/Admin möchte die Datenart ändern
2. Der Server greift erneut auf die Datenbank zu
3. Die Website generiert ein neues Diagramm, den angegebenen Daten entsprechend

3.6.2. Anderer Zeitraum

1. Der User/Admin möchte den Zeitraum ändern
2. Der Server greift erneut auf die Datenbank zu
3. Die Website generiert ein neues Diagramm, dem angegebenen Zeitraum entsprechend

3.7. Wesentliche Szenarios

3.8. Nachbedingungen

3.8.1. Diagrammanzeige

1. Dem User/Admin wird ein Diagramm entsprechend den ausgewählten Daten und dem Zeitraum angezeigt und wartet auf Änderungen von User-/Adminseite

3.9. Besondere Anforderungen

3.9.1. Gleichzeitige Änderung der Daten und des Zeitraumes

1. Der User/Admin hat die Möglichkeit die Datenart und den Zeitraum gleichzeitig zu ändern, sodass das Diagramm dementsprechend generiert wird

3.10. Wireframes

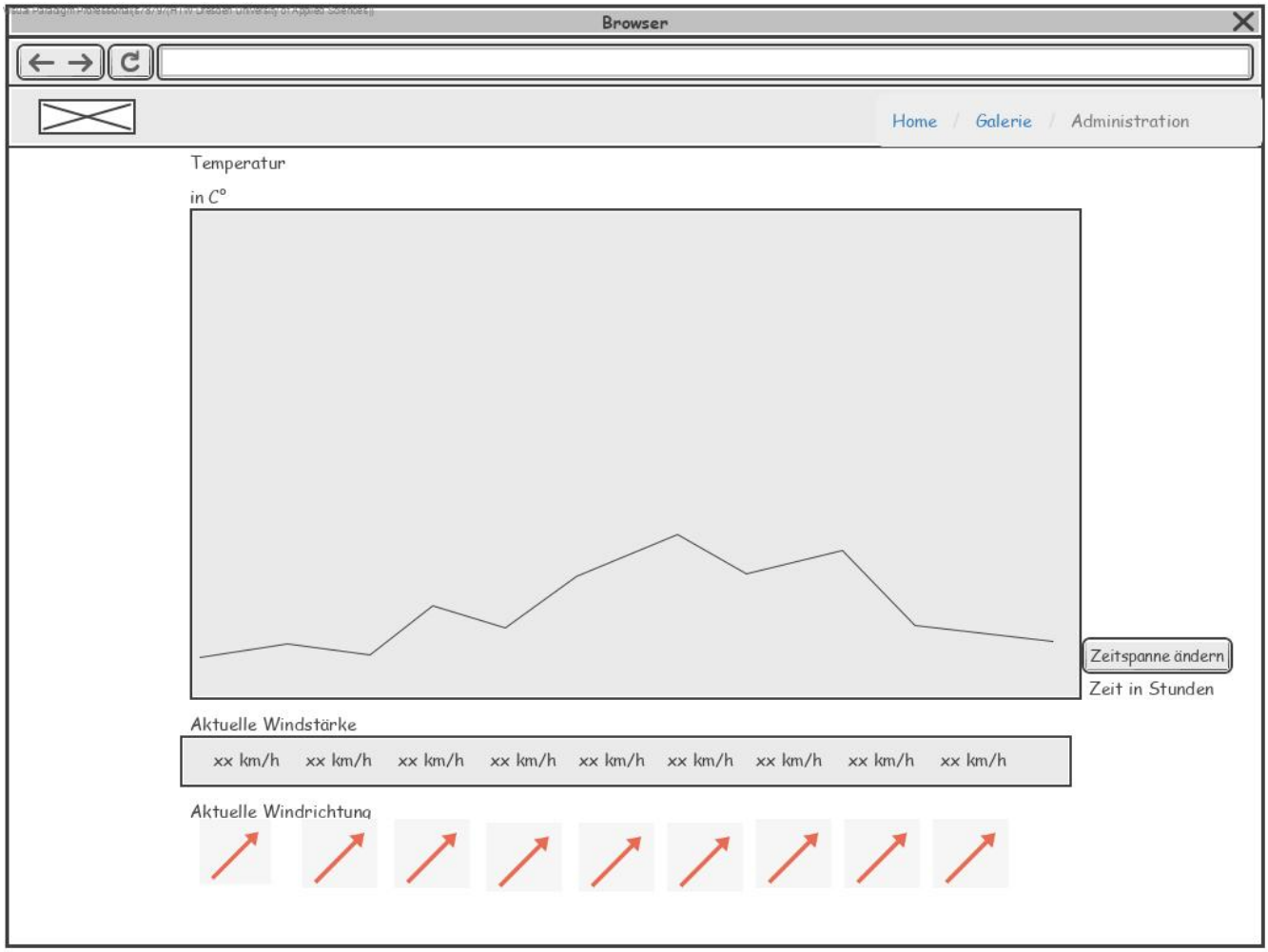

Figure 5. Wireframe Diagramme

4. Use Case: Daten ändern

4.1. Kurzbeschreibung

Administratoren wird es ermöglicht bestimmte Werte, welche in einer Tabelle der Datenbank gespeichert sind, zu ändern. Diese Tabelle wird nach jedem Neustart des Raspis ausgelesen und die Werte werden übernommen.

4.2. Kurzbeschreibung der Akteure

4.2.1. Administratoren

Nur Administratoren, sprich User mit einem gültigen Login und entsprechenden Zugriffsrechten auf erweiterte Funktionen, haben Zugriff auf die Funktionalitäten, die in diesem Use Case definiert sind

4.3. Vorbedingungen

1. Datenbank ist aktiv und erfolgreich mit Website verbunden
2. Verbindung zwischen Raspi und Datenbankserver wird erfolgreich hergestellt

4.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Administrator auf den Adminbereich zugreift
2. Der Administrator sieht den aktuellen Inhalt der Config-Tabelle
3. Der Administrator kann die Werte über die Website ändern
4. Die Website übermittelt die neuen Werte an den Datenbankserver
5. Der Datenbankserver speichert die neuen Werte in der Config-Tabelle
6. Der Raspi baut nach jedem Neustart eine Verbindung zum Datenbankserver her
7. Der Raspi fragt den Inhalt der Config-Tabelle nach jedem Neustart ab
8. Der Raspi übernimmt die Werte aus der Config-Tabelle
9. Der Use Case ist abgeschlossen.

4.5. Alternative Abläufe

4.5.1. Verbindungsfehler

Wenn die Website im Schritt 4 des Standardablauf keine Verbindung zum Datenbankserver aufbauen kann, dann wird dem Admin eine Fehlermeldung ausgegeben

1. Bei Verbindungsfehler wird Fehlermeldung im Adminbereich angezeigt
2. Der Use Case wird im Schritt 9 fortgesetzt.

4.5.2. Unzulässige Werte

Wenn der Admin im Schritt 3 des Standardablaufes unzulässige Werte in die Config-Tabelle eintragen möchte wird eine Warnung ausgegeben

1. Bei Änderungen der Werte in unzulässige Werte wird eine Warnmeldung ausgegeben
2. Der Use Case wird im Schritt 3 fortgesetzt.

4.6. Wesentliche Szenarios

4.7. Nachbedingungen

4.7.1. Änderungen des Tabelleninhaltes

1. Der Raspi übernimmt die Änderungen und reagiert dementsprechend auf diese

4.8. Besondere Anforderungen

4.8.1. Einhalten der Datentypen

1. Jede Zeile der Tabelle Config kann nur bestimmte Werte speichern
2. Der Admin darf die Werte nur so ändern, dass diese Ordnungsgemäß gespeichert und verarbeitet werden können

4.9. Wireframes

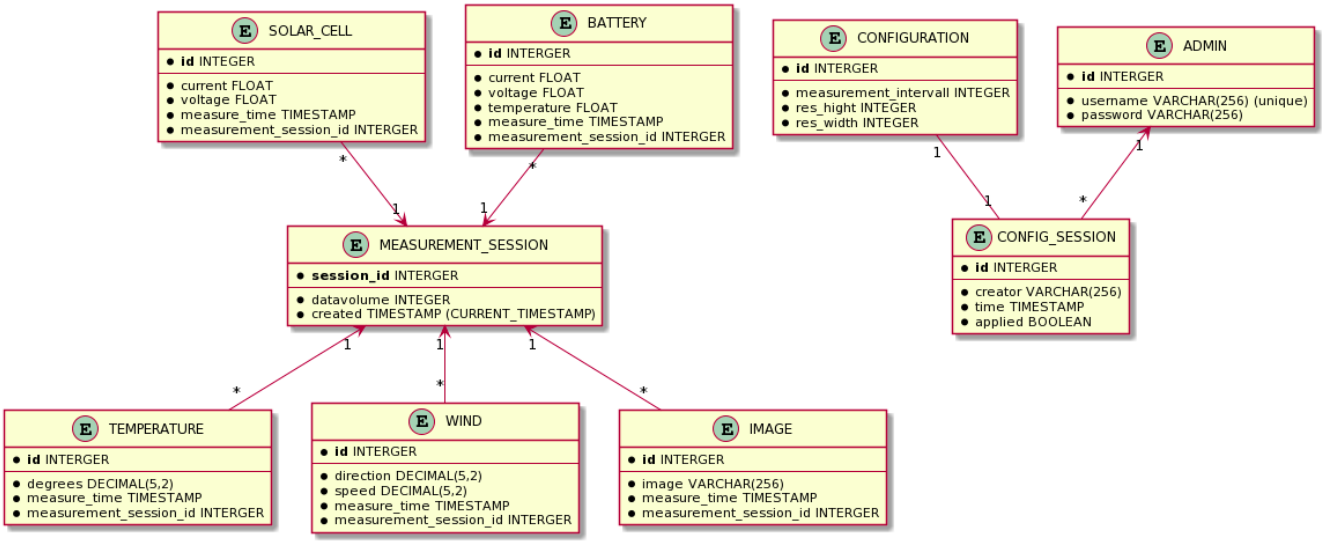

Figure 6. ERM

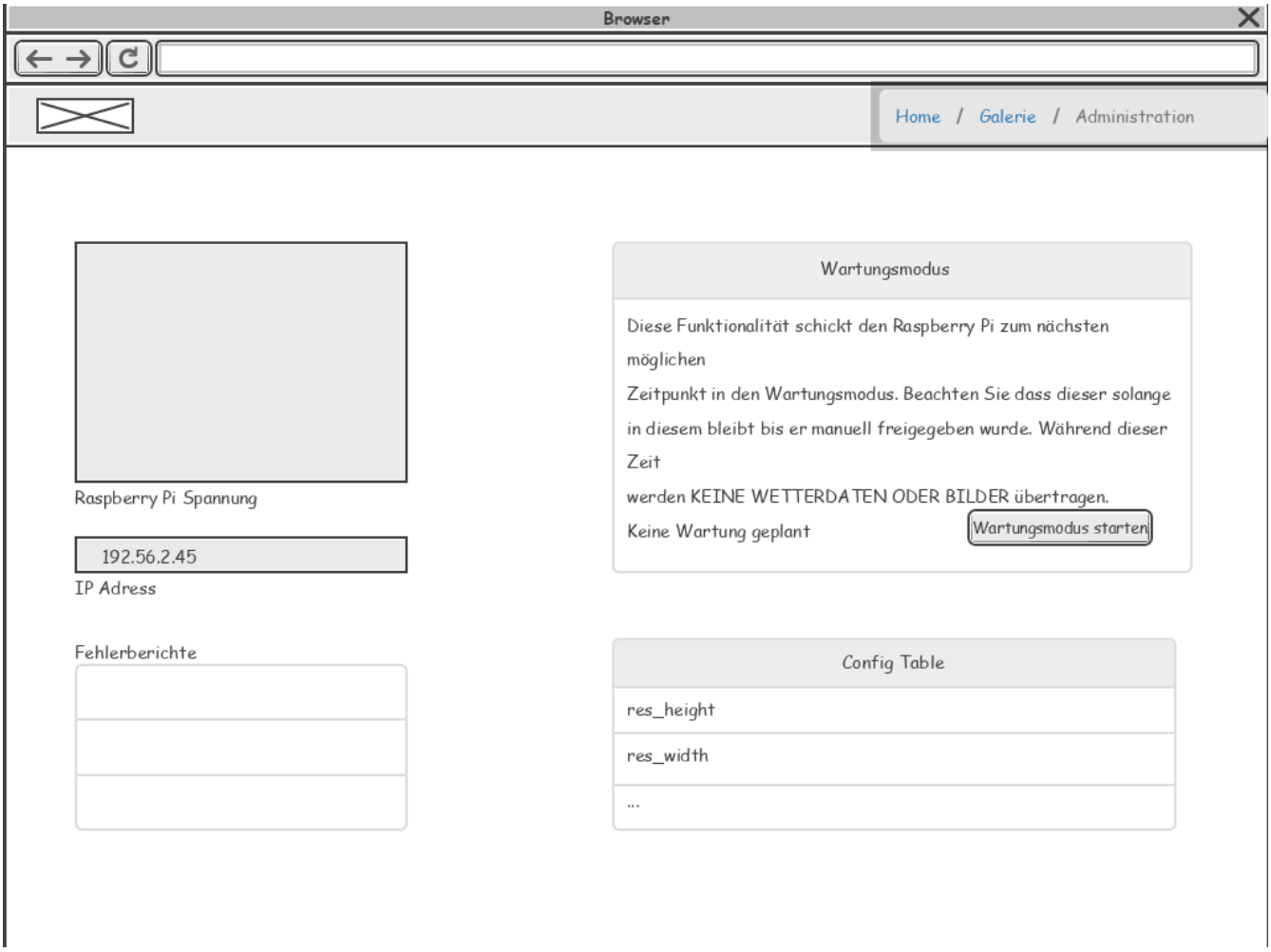

Figure 7. Wireframe

5. Use Case: Logfile lesen

5.1. Kurzbeschreibung

In diesem Use Case wird die Auswertung der Daten, welche in den Logfiles gespeichert werden, beschrieben.

5.2. Kurzbeschreibung der Akteure

5.2.1. Administratoren

Nur Administratoren, sprich User mit einem gültigen Login und entsprechenden Zugriffsrechten auf erweiterte Funktionen, haben Zugriff auf die Funktionalitäten, die in diesem Use Case definiert sind

5.3. Vorbedingungen

1. Speichern bestimmter Daten in einer Logfile
2. Speichern der Logfile in der Datenbank

3. Verbindung zwischen Website und Datenbank wird ordnungsgemäß hergestellt

5.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Admin auf den Adminbereich zugreift
2. Der Inhalt der Logfile wird dem Admin im Adminbereich angezeigt
3. Der Use Case ist abgeschlossen.

5.5. Alternative Abläufe

5.6. Unterabläufe (subflows)

5.6.1. Überschreiben der Logfile

1. Die Logfile wird in der Datenbank gespeichert
2. Beim erneuten Start des Raspi wird eine neue Logfile an die Datenbank übermittelt
3. Die neue Logfile überschreibt die schon gespeicherte Logfile in der Datenbank

5.7. Wireframes

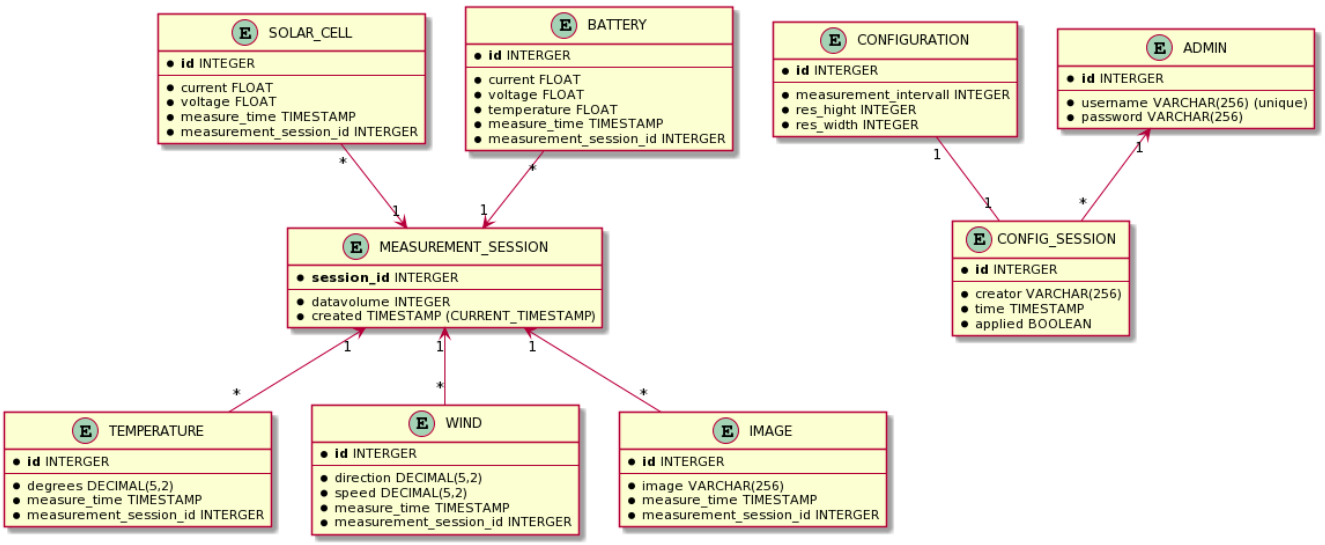

Figure 8. ERM