

# Technische Spezifikation

## Baby-Blooz Baby-Überwachung

gute und start verbesserte technische Spezifikation, die jedoch weiterhin nicht ganz vollständig ist.

Weitere Anmerkungen siehe unten!

### Wertung:

Anwendung: 9 Punkte (WLAN noch nicht vorführbar)

TechSpec: 4,5 Punkte

Projektplan: 4,5 Punkte

Qualitätssicherung: 0 Punkte (Dokument liegt in git nicht vor. Bei nachträgliche Abgabe erfolgt 50% Abzug)

### Autor:

Abdallah Tahboub, Mohamed Ben Amar

,Skander Baraket, Aziz Khelladi

Letzte Änderung: 27.06.2019

Dateiname:TechSpech\_Baby-Blooz

<b>Inhaltsverzeichnis</b>	<b>1</b>
1. Einleitung	5
1.1 Überblick	5
1.2 Definitionen und Abkürzungen	5
1.3 Vorhandene Dokumente	6
2. Prozessüberblick	6
2.1 Realisierungsprozess	7
2.2 Technischer Workflow	8
3. System Architektur und Infrastruktur	8
3.1 Systemarchitektur	8
3.2 System Infrastruktur	9
4. Spezifikationen Software	10
4.1 Überblick Komponenten	11
4.2 Schnittstellen zwischen den Komponenten	11
4.3 Beschreibung der Implementierung	11
4.3.1 F1: Log In	11
4.3.2 F2: WiFi Konfiguration	14
4.3.3 F3: Echtzeitdaten Anzeigen	16
4.3.4 F4: History-Page Anzeigen	16
4.3.5 F5: Push-Benachrichtigungen Anzeigen	16
4.3.6 F6: Sensordaten in DB Speichern	17
4.3.7 F7: Daten Verarbeiten	17
4.3.8 F8: Daten Lesen/Senden	17
5. Spezifikation Hardware	19
5.1 Schaltungsplan	19
5.2 Einzelteile	20

## Abbildungsverzeichnis

Abbildung 1: Diagramm des Realisierungsprozesses	5
Abbildung 2: Technischer Workflow	6
Abbildung 3: Systemarchitektur	8
Abbildung 4: Spezialisierung Sensoren	9
Abbildung 5: Komponentendiagramm	10
Abbildung 6: Log In Seite	12
Abbildung 7: Ablauf der Anmeldung	13
Abbildung 8: Konfigurationsseite	14
Abbildung 9: Ablauf der WLAN-Einstellung	15
Abbildung 10: Ablauf Daten Lesen/Senden	18
Abbildung 11: Schaltungsplan von Arduino	19
Abbildung 12: Lilypad Bewegungssensor	20
Abbildung 13: Herzschlagsensor	20
Abbildung 14: Knopfzellen-Batteriehalter	20
Abbildung 15: Temperatursensor	21
Abbildung 16: Lilypad	21
Abbildung 17: WLAN Modul	22

## Tabellenverzeichnis

Tabelle 1: Versionshistorie	4
Tabelle 2: Vorhandene Dokumente	4
Tabelle 3: Zugriffsdaten	9
Tabelle 4: Beschreibung der Softwarekomponenten und deren Funktionen	10
Tabelle 5: Komponenten Log In	11
Tabelle 6: Komponenten WiFi Konfiguration	11
Tabelle 7: Komponenten Echtzeitdaten	12
Tabelle 8: Komponenten History-Page	13
Tabelle 9: Komponenten Push-Benachrichtigungen	15
Tabelle 10: Komponenten Datenbank	17
Tabelle 11: Komponenten Daten Verarbeiten	17
Tabelle 12: Komponenten Daten Lesen/Senden	17

© 2019 **Baby-Blooze**

Die Weitergabe, Vervielfältigung oder anderweitige Nutzung dieses Dokumentes oder Teile davon ist unabhängig vom Zweck oder in welcher Form untersagt, es sei denn, die Rechteinhaber/In hat ihre ausdrückliche schriftliche Genehmigung erteilt.

### Versionshistorie:

Version	Datum	Verantwortlich	Änderung
1.0	23.05.19	Skander,Mohamed,Abdallah, Aziz	Initiale Dokumentenerstellung
1.0	11.05.19	Mohamed,Aziz,Abdallah,Skander	Diagramme und Schaltpläne
1.0	19.05.19	Abdallah,Skander,Aziz,Mohamed	Texte
1.0	19.05.19	Aziz,Mohamed,Abdallah,Skander	Bilder, Diagramme und Texte
1.0	25.05.19	Abdallah,Skander,Aziz,Mohamed	Bilder, Text und Diagramme
1.0	05.06.19	Skander,Mohamed,Abdallah, Aziz	Diagramme und Texte
2.0	10.06.19	Skander,Mohamed,Abdallah, Aziz	Bilder, Text und Diagramme
2.0	12.06.19	Skander,Mohamed,Abdallah, Aziz	Bilder, Text und Diagramme
2.0	14.06.19	Skander,Mohamed,Abdallah, Aziz	Bilder, Text und Diagramme
2.0	19.06.19	Skander,Mohamed,Abdallah, Aziz	Bilder, Text und Diagramme
2.0	23.06.19	Skander,Mohamed,Abdallah, Aziz	Bilder, Text und Diagramme
2.0	24.06.19	Skander,Mohamed,Abdallah, Aziz	Bilder, Text und Diagramme
2.0	27.06.19	Skander,Mohamed,Abdallah, Aziz	Bilder, Text und Diagramme

*Tabelle 1: Versionshistorie*

# 1. Einleitung

## 1.1 Überblick

Damit ein Baby remote überwacht werden kann, wird einen Baby Anzug und eine Mobile App, die miteinander durch API kommunizieren werden, gemacht. Der Hardware Teil (Baby Kleidung) besteht aus einem kleinem passendem Arduino heißt Lilypad, der für unseres Projekt sehr gut geeignet ist, und aus verschiedenen Sensoren, deren Aufgaben es sind, die Messwerte des Babys, aufzunehmen und auf der Oberfläche der App anzuzeigen bzw. zu vergleichen. Die App besteht aus WebAPI, einer Datenbank zum speichern der Daten und einer Benutzeroberfläche(Frontend). Mit der App kann sich der Benutzer mit Hilfe eines QR codes anmelden und Daten abrufen.

Außerdem ist der Benutzer in der Lage die WLAN des Lilypads einzustellen. Notwendige und kritische Infos werden dem Benutzer als Push-Benachrichtigungen angezeigt .

## 1.2 Definitionen und Abkürzungen

API – Application programing interface

QR Code – Quick response code

## 1.3 Vorhandene Dokumente

Dokument	Autor	Datum
Lastenheft_Baby-Blooz	Abdallah,Aziz,Skandar,Mohamed	26.05.19
Pflichtenheft_Baby-Blooz	Skandar,Aziz,Abdallah,Mohamed	17.05.19

*Tabelle 2: Vorhandene Dokumente*

## 2. Prozessüberblick

### 2.1 Realisierungsprozess

Um dieses Projekt zu realisieren ist zuallererst die Strukturierung des gesamten Systems erforderlich. Nachdem das Modell der Baby-blooz strukturiert wurde, müssen alle Bauteile des Projektes besorgt werden und gleichzeitig kann mit der Entwicklung begonnen werden. Diese besteht aus zwei Teilen: Der Mobile App und dem Mikrocontroller.

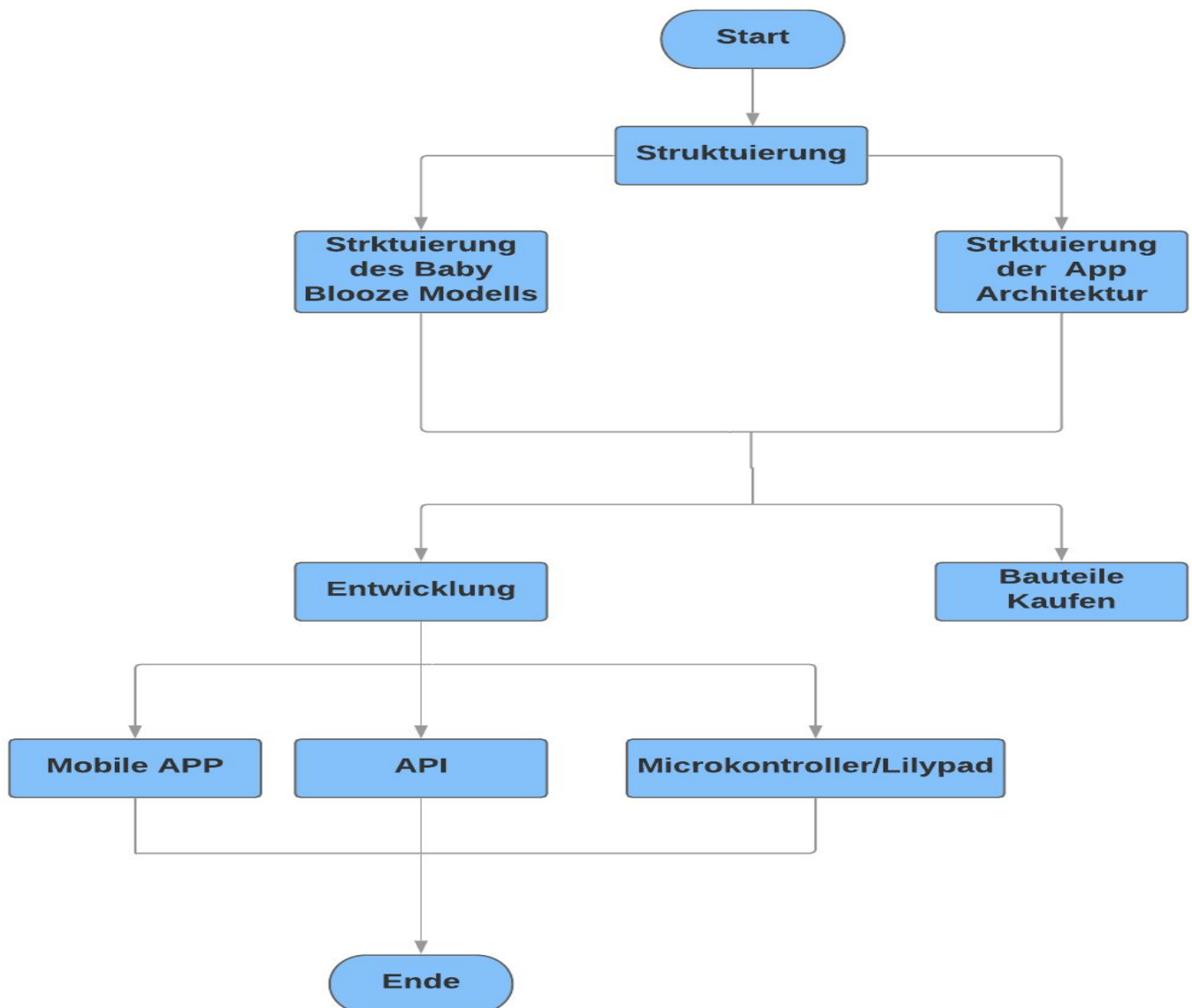


Abbildung 1: Diagramm des Realisierungsprozesses

## 2.2 Technischer Workflow

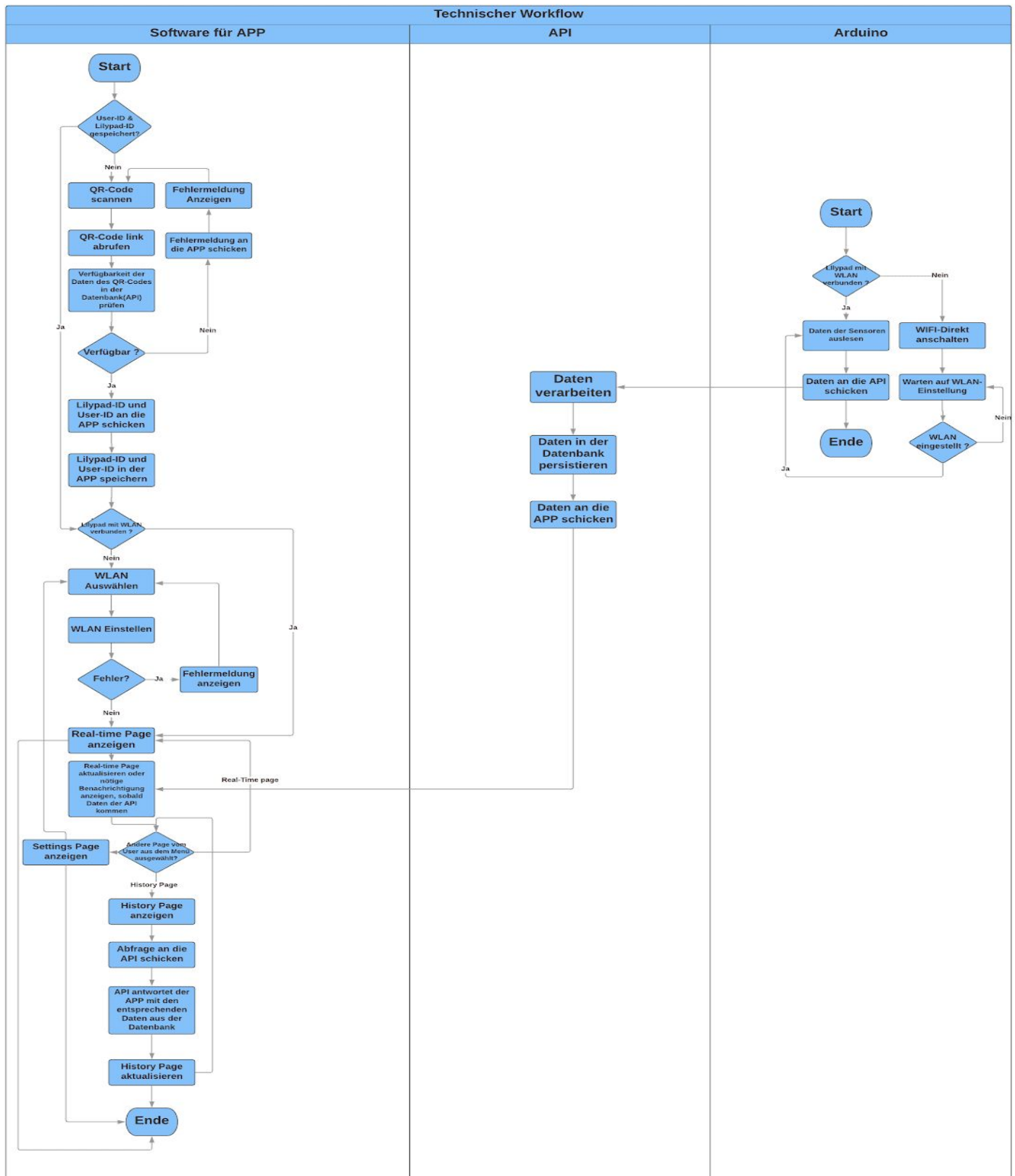


Abbildung 2: Technischer Workflow

### 3. System Architektur und Infrastruktur

#### 3.1 Systemarchitektur

Die folgende Abbildung zeigt unsere Systemarchitektur in Verbindung mit dem Anwender.

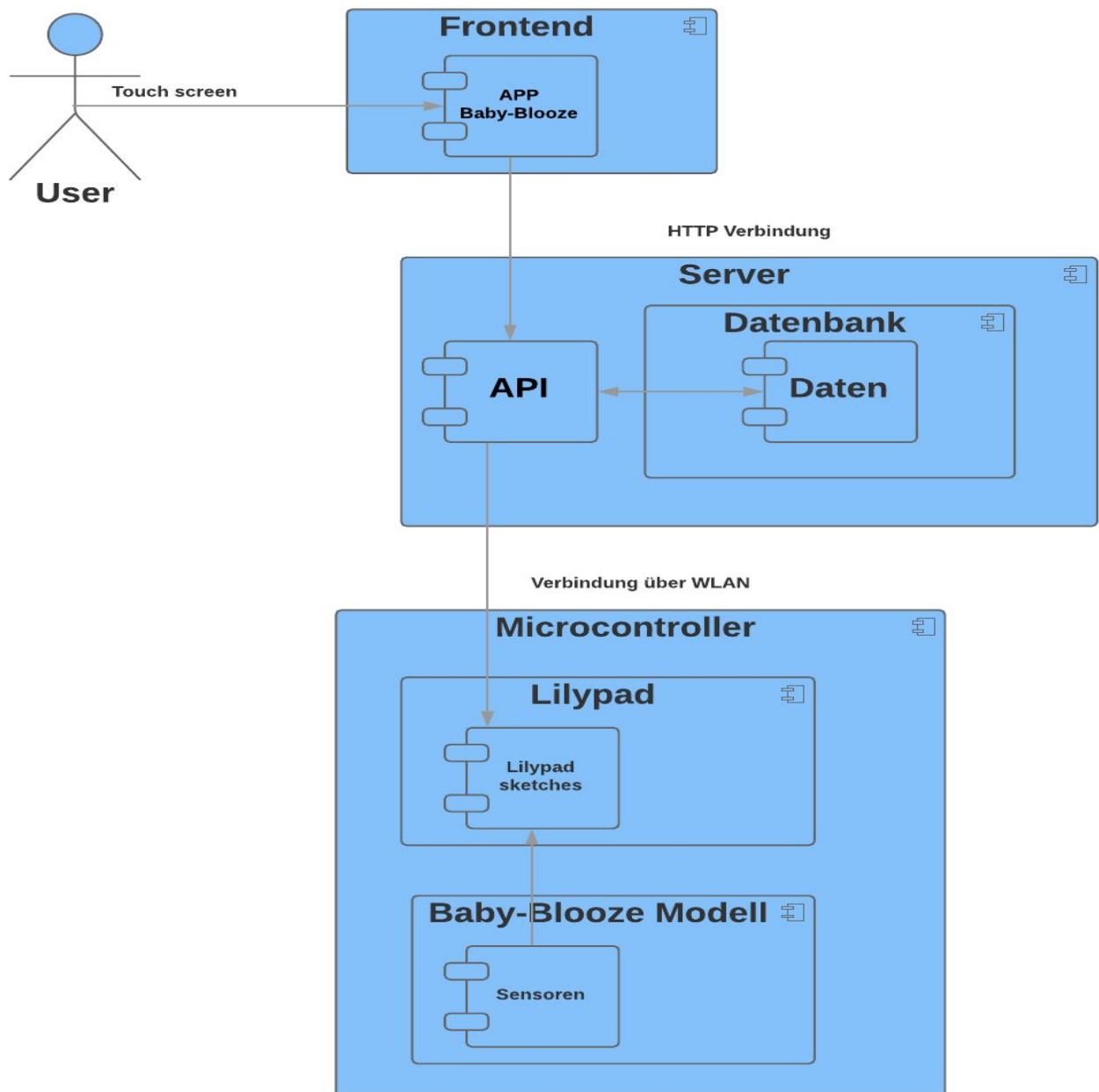


Abbildung 3: Systemarchitektur



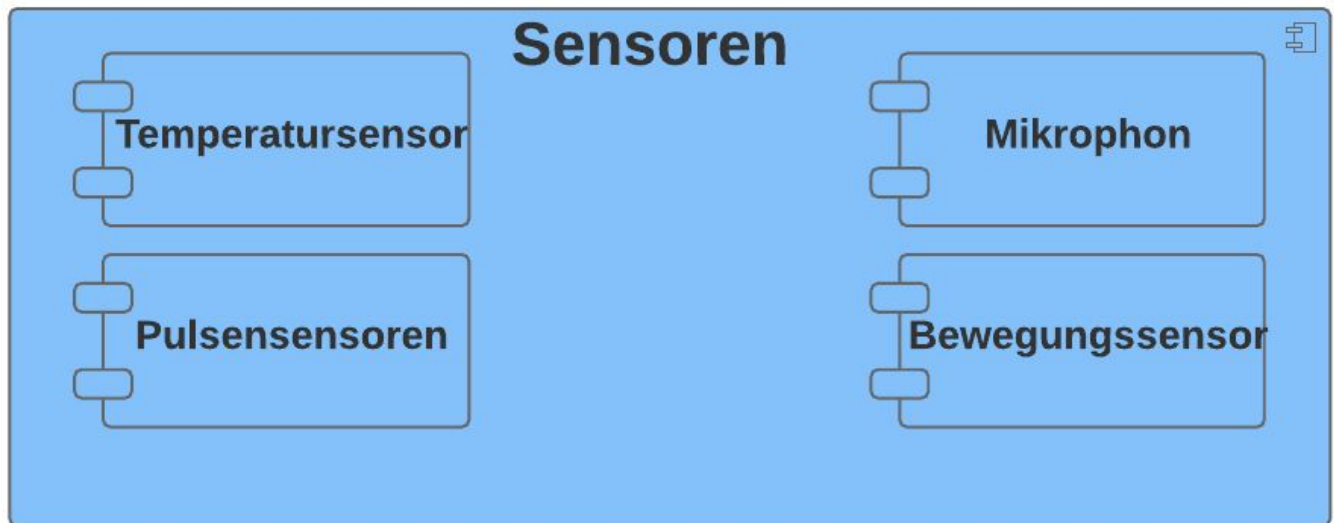


Abbildung 4: Spezialisierung Sensoren

### 3.2 System Infrastruktur

Daten die benötigt werden um Zugriff auf den Server zu erhalten, worauf die Daten zu dem Webserver, sowie der Datenbank zu finden sind.

IP	149.233.37.64
Benutzer	User ID
QR Code	Generierter QR Code

Tabelle 3: Zugriffsdaten

149.233.37.64

## 4. Spezifikationen Software

### 4.1 Überblick Komponenten

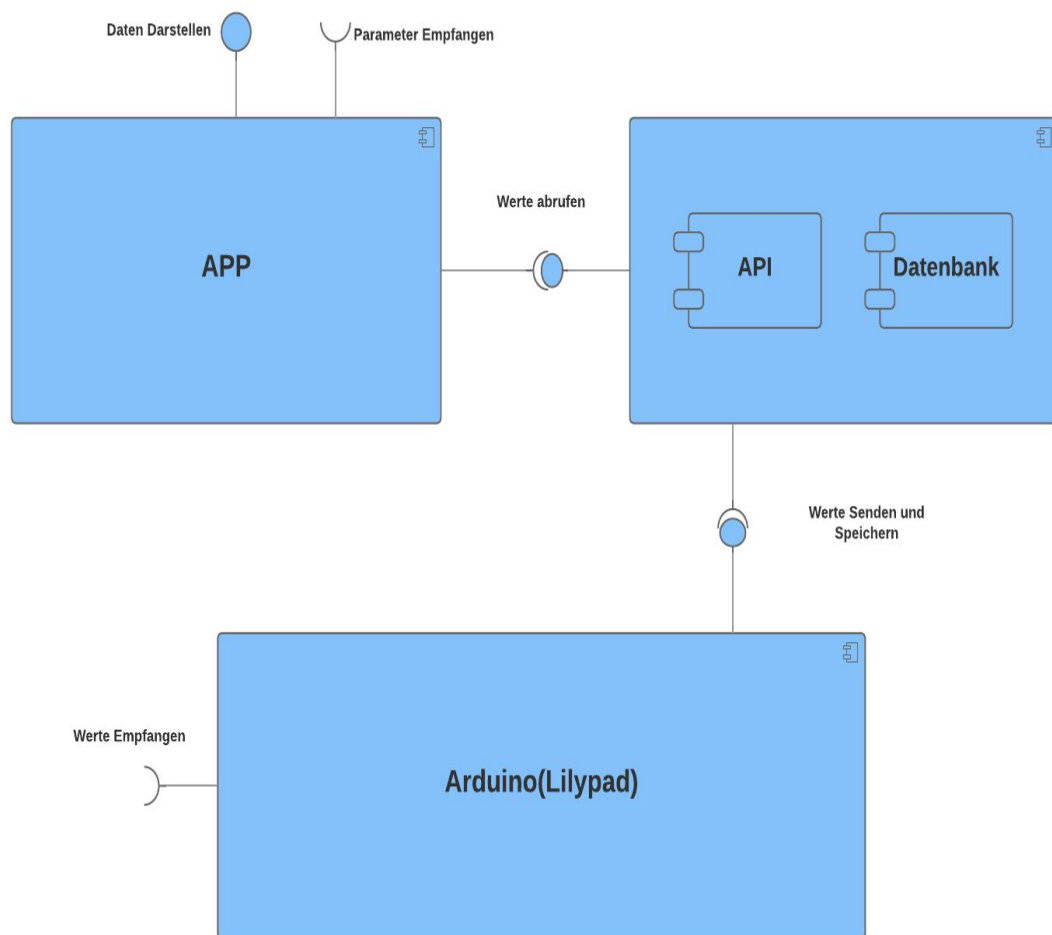


Abbildung 5: Komponentendiagramm

SW Komponente	Funktionen	Sprache / Typ	Ort
App	F1: Login F2: WiFi Konfigurieren F3: Real-Time Daten anzeigen F4: History Page anzeigen F5: Push-Benachrichtigungen anzeigen	HTML, CSS, js, xml typescript	App
API Datenbank	F6: Daten in DB speichern F7: Daten verarbeiten	C#,MySQL,Ty pescrypt	Server

Arduino (Lilypad)	F8:Sensordaten lesen	Arduino	Lilypad
-------------------	----------------------	---------	---------

Tabelle 4: Beschreibung der Softwarekomponenten und deren Funktionen

## 4.1 Schnittstellen zwischen den Komponenten

Die Übertragung der Sensordaten zwischen dem Arduino und dem Server funktioniert über das WLAN mit Hilfe des WLAN Moduls(ESP8266), welches an dem Arduino angeschlossen ist. Die Datenbank befindet sich auf demselben Server weshalb diesbezüglich keine weiteren Schnittstellen erforderlich sind. Der Server bekommt die Sensordaten vom Arduino. Der Server schickt in der echten Zeit die Sensordaten durch WLAN zur APP. Die APP selbst ist eine grafische Schnittstelle, die es dem Nutzer ermöglicht die Daten aus dem System auszulesen und die Anlage fernzusteuern. Die App kann in jeder Zeit die Sensordaten von der Datenbank, die im Server befindet, erhalten.

## 4.2 Beschreibung der Implementierung

### 4.2.1 F1: Log In

Um den Zugriff auf alle Daten und Funktionen zu erhalten, muss man sich als erstes im Log In Fenster anmelden. Dazu gibt es ein Feld für den QR-Code zu scannen. Mit dem "Scan QR-Code"-Button wird der Login Versuch gestartet. Bei erfolgreicher Anmeldung wird der Nutzer auf die Startseite, das *Dashboard* weitergeleitet. Bei erfolgloser Anmeldung wird der Nutzer zu einer Neuen QR-code zum scannen gebetet. Ein Account muss dafür in der Datenbank erstellt worden sein. Wenn eine Bluse schon erstellt wurde, wird der angepasste QR-code in der Bank gespeichert.

#	Komponente	Erforderliche Arbeiten
	APP	Vergleichen der QR-code mit den QR-code, die aus der Datenbank gespeicherten Blusen.
	Datenbank	Bereitstellen des Benutzer mithilfe des QR-codes.

Tabelle 5: Komponenten Log In

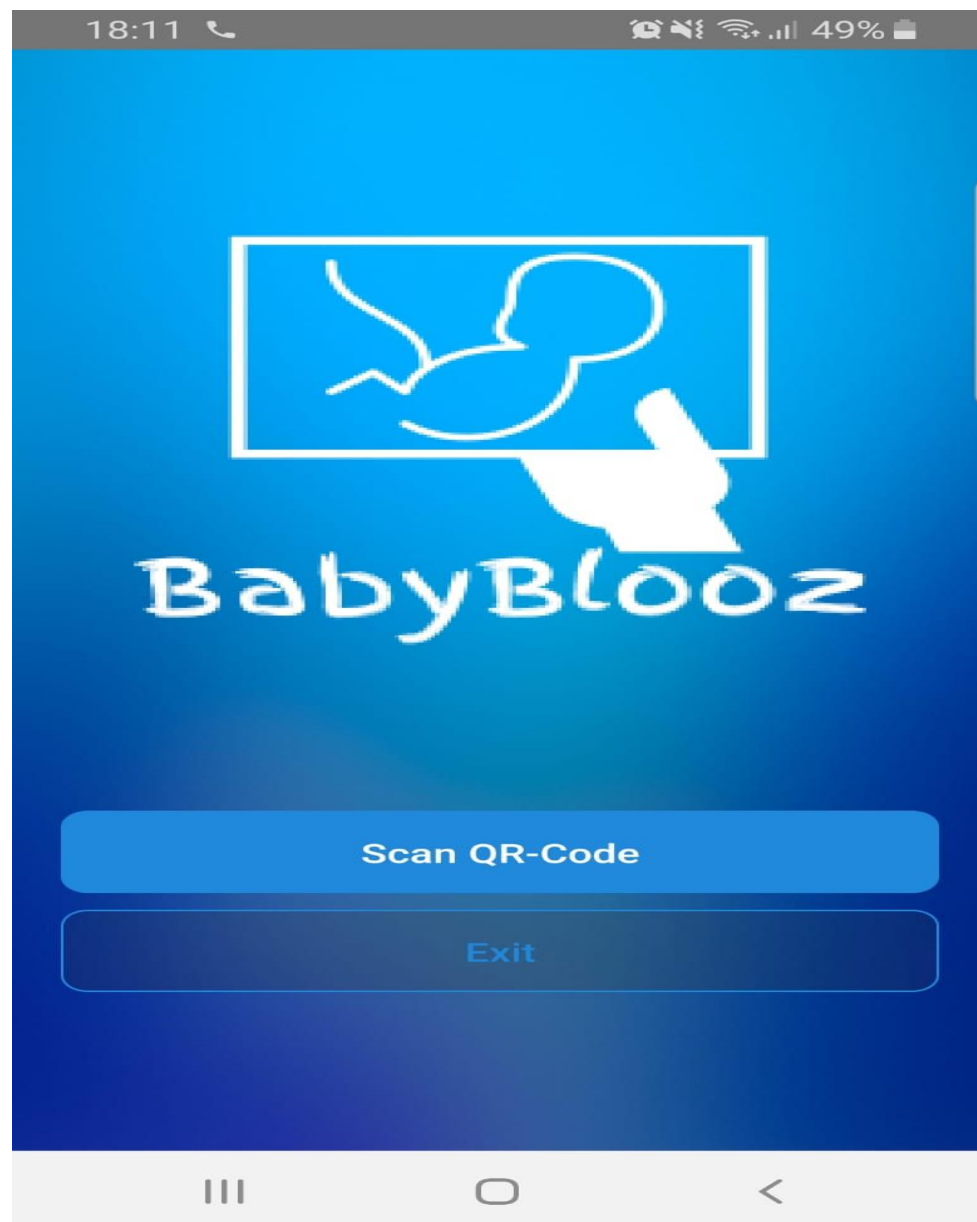


Abbildung 6: Log In Seite

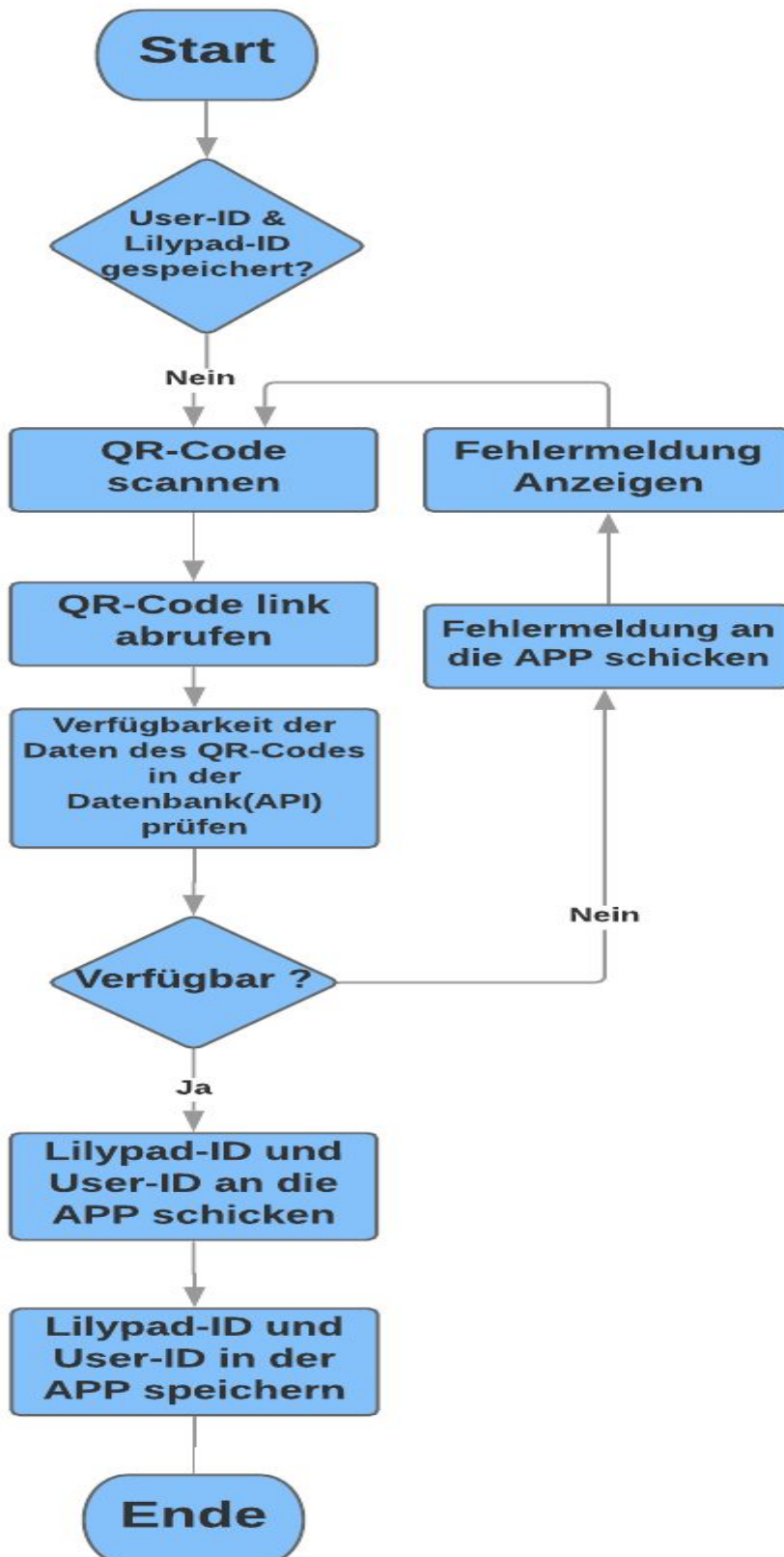


Abbildung 7: Ablauf der Anmeldung

#### 4.2.2 F2: WiFi Konfiguration

Nachdem sich der Benutzer mit dem QR-Code angemeldet hat, landet er auf die WiFi Konfigurationsseite, wo man mit dem Internet verbinden kann. Dort darf man sich aus Sicherheitsgründen ausschließlich mit geschützten Netzen anmelden. Der Benutzer klickt auf dem gewünschten Netz und gibt das Passwort ein. Schließlich bestätigt der User seine Eingabe mit einem Fingerklick auf "Confirm". Sollte das eingegebene Passwort falsch sein, erscheint eine Fehlermeldung, dass es keine Verbindung mit dem Netz erstellt werden darf.

#	Komponente	Erforderliche Arbeiten
	APP	<ul style="list-style-type: none"><li>Mit dem Finger kann man das Netz auswählen und das Passwort eingeben.</li></ul>

Tabelle 6: Komponenten WiFi Konfiguration

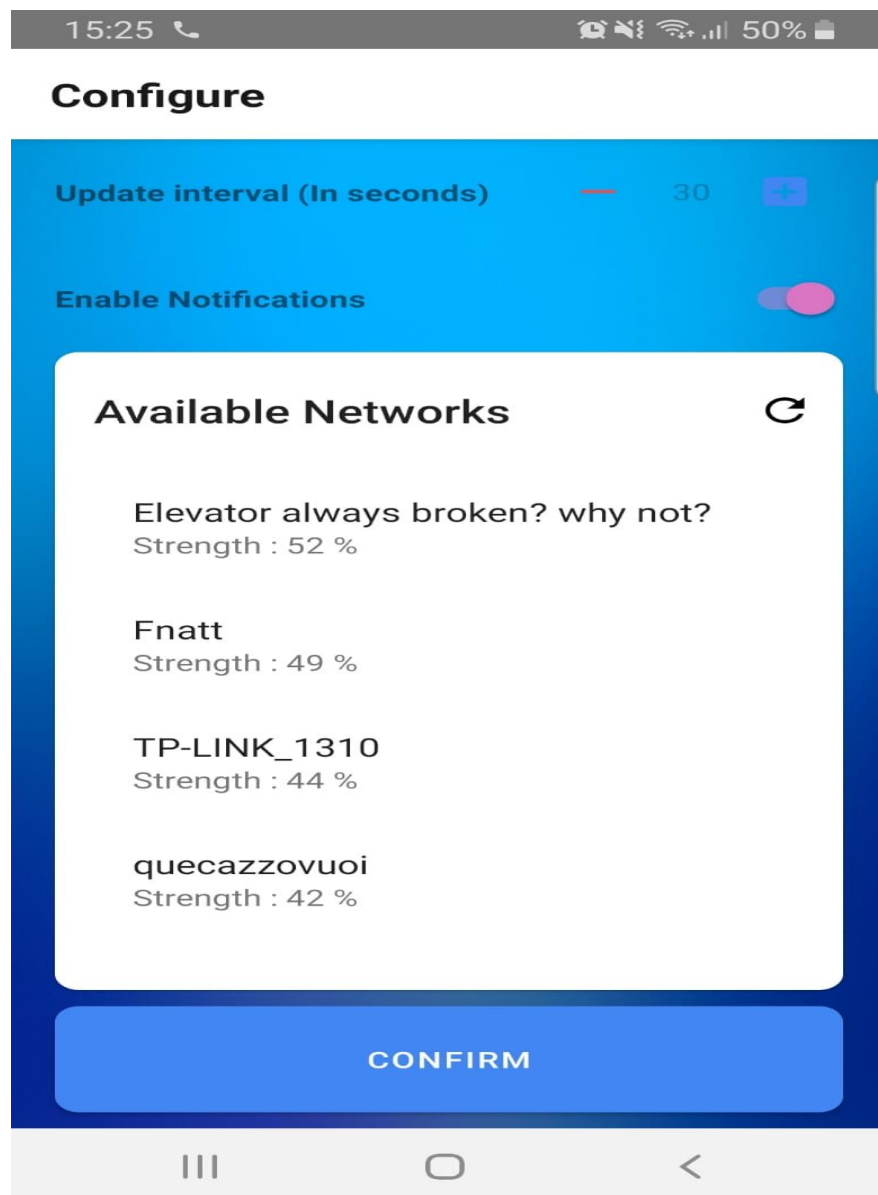


Abbildung 8: Konfigurationsseite

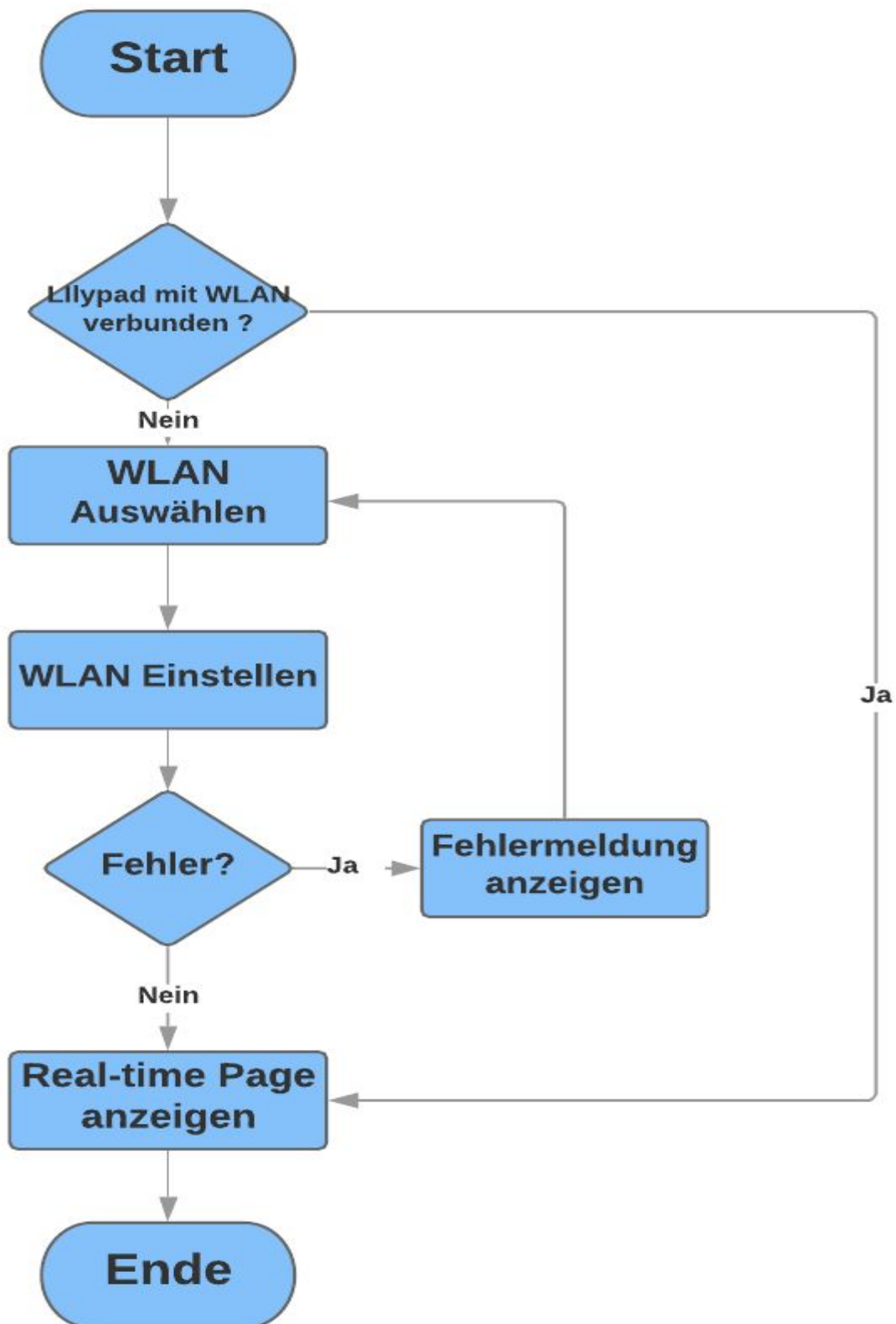


Abbildung 9: Ablauf der WLAN-Einstellung

#### 4.2.3 F3:Real-Time Daten Anzeigen

Sobald die App mit dem Internet erfolgreich verbunden ist wurde,lassen sich Real-Time Daten der Sensoren darstellen. Dort hat man ein Einblick in die Sensordaten, die in der Datenbank gespeichert sind .

#	Komponente	Erforderliche Arbeiten
	APP	<ul style="list-style-type: none"> <li>• Es muss eine fehlerfreie Anmeldung des Benutzers bestehen.</li> <li>• Die App muss mit dem Internet verbunden sein.</li> <li>• Der Benutzer kann mit dem Finger auf der Echtzeit-Seite. hoch und runterscrollen oder rein und raus zoomen.</li> </ul>
	API	<ul style="list-style-type: none"> <li>• Verarbeitung der Sensordaten.</li> <li>• Die Daten werden durch die API geschickt.</li> </ul>
	Datenbank	<ul style="list-style-type: none"> <li>• Datenbank stellt die Sensordaten bereit.</li> </ul>

Tabelle 7: Komponenten Echtzeitdaten

#### 4.2.4 F4:History-Page

Der Benutzer wird in der Lage sein, die History in der Vergangenheit von 90 Tagen anzusehen. Außerdem hat der Benutzer die Möglichkeit, die 90 Tagen der History Seite in Teilsegmenten anzeigen zu lassen.Eine Woche,einen Monat und drei Monate.

Die History Seite beinhaltet Daten von Herzschlägen, Schlafrhythmus, und die Temperatur.

#	Komponente	Erforderliche Arbeiten
	APP	<ul style="list-style-type: none"> <li>• Es muss eine fehlerfreie Anmeldung des Benutzers bestehen (User meldet sich an).</li> <li>• Die App muss mit dem Internet verbunden sein(User stellt die W LAN ein)</li> <li>• Der Benutzer kann mit dem Finger auf der Echtzeit-Seite. hoch und runterscrollen oder rein und raus zoomen.</li> </ul>
	API	<ul style="list-style-type: none"> <li>• Verarbeitung der Sensordaten.</li> <li>• Die Daten werden durch die API geschickt.</li> </ul>
	Datenbank	<ul style="list-style-type: none"> <li>• Datenbank stellt die Sensordaten bereit.</li> </ul>

Tabelle 8: Komponenten History-Page

#### 4.2.5 F5: Push-Benachrichtigungen

Push-Benachrichtigungen werden auf dem Bildschirm angezeigt, sobald notwendige oder kretische medizinische Feststellungen zur Baby in Erscheinung treten.

#	Komponente	Erforderliche Arbeiten
	APP	<ul style="list-style-type: none"> <li>• Anzeigen der Push-Benachrichtigungen auf dem Bildschirm</li> </ul>
	API	<ul style="list-style-type: none"> <li>• Verarbeitung der Sensordaten.</li> </ul>
	Datenbank	<ul style="list-style-type: none"> <li>• Datenbank stellt die Sensordaten bereit.</li> </ul>

Tabelle 9: Komponenten Push-Benachrichtigungen



#### 4.2.6 F6: Daten in der Datenbank speichern

Mit dieser Funktion werden alle von Sensor gelesenen Daten durch die API in einer Datenbank gespeichert. Mit Hilfe der WLAN werden Daten zur API transportiert.

#	Komponente	Erforderliche Arbeiten
	Datenbank	<ul style="list-style-type: none"> <li>• Empfang der Sensordaten</li> <li>• Speichern der empfangenen Sensordaten</li> </ul>

Tabelle 10: Komponenten Datenbank



#### 4.2.7 F7: Daten verarbeiten

Die Daten werden nach der Sendung, von API empfangen, und dort medizinisch verarbeitet. Medizinisch bedeutet, wenn z.B. die Temperatur zu hoch oder zu niedrig ist, wird diese Daten in der API verarbeitet.

#	Komponente	Erforderliche Arbeiten
	API	<ul style="list-style-type: none"> <li>• Sensordaten Empfangen</li> <li>• Sensordaten verarbeiten</li> <li>• Sensordaten an die App und an die Datenbank senden</li> </ul>

Tabelle 11: Komponenten Daten Verarbeiten

#### 4.2.8 F8: Sensordaten lesen/senden

Jede Sekunde wird der Wert der Temperatursensors, Mikrophons, Bewegungssensors, und Herzschlagsensor ausgelesen. Alle Sensordaten werden an die API geschickt.

#	Komponente	Erforderliche Arbeiten
	Arduino lilypad	<ul style="list-style-type: none"> <li>• Empfang der Sensordaten</li> <li>• Ausgelesene Sensordaten an die API schicken</li> </ul>
	API	<ul style="list-style-type: none"> <li>• Empfang der von Lilypad gesendeten Sensordaten</li> </ul>

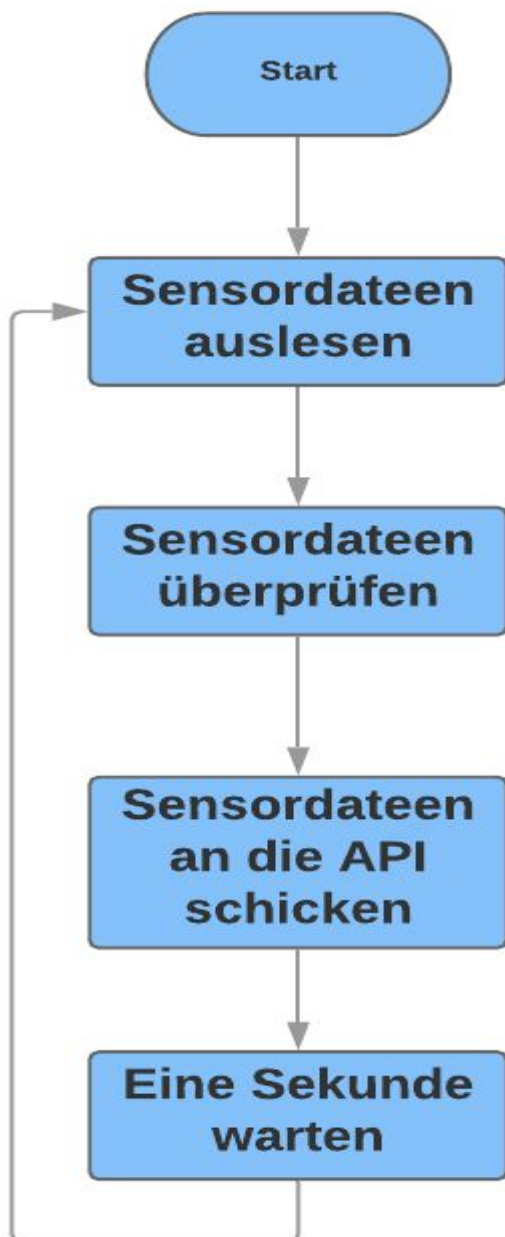
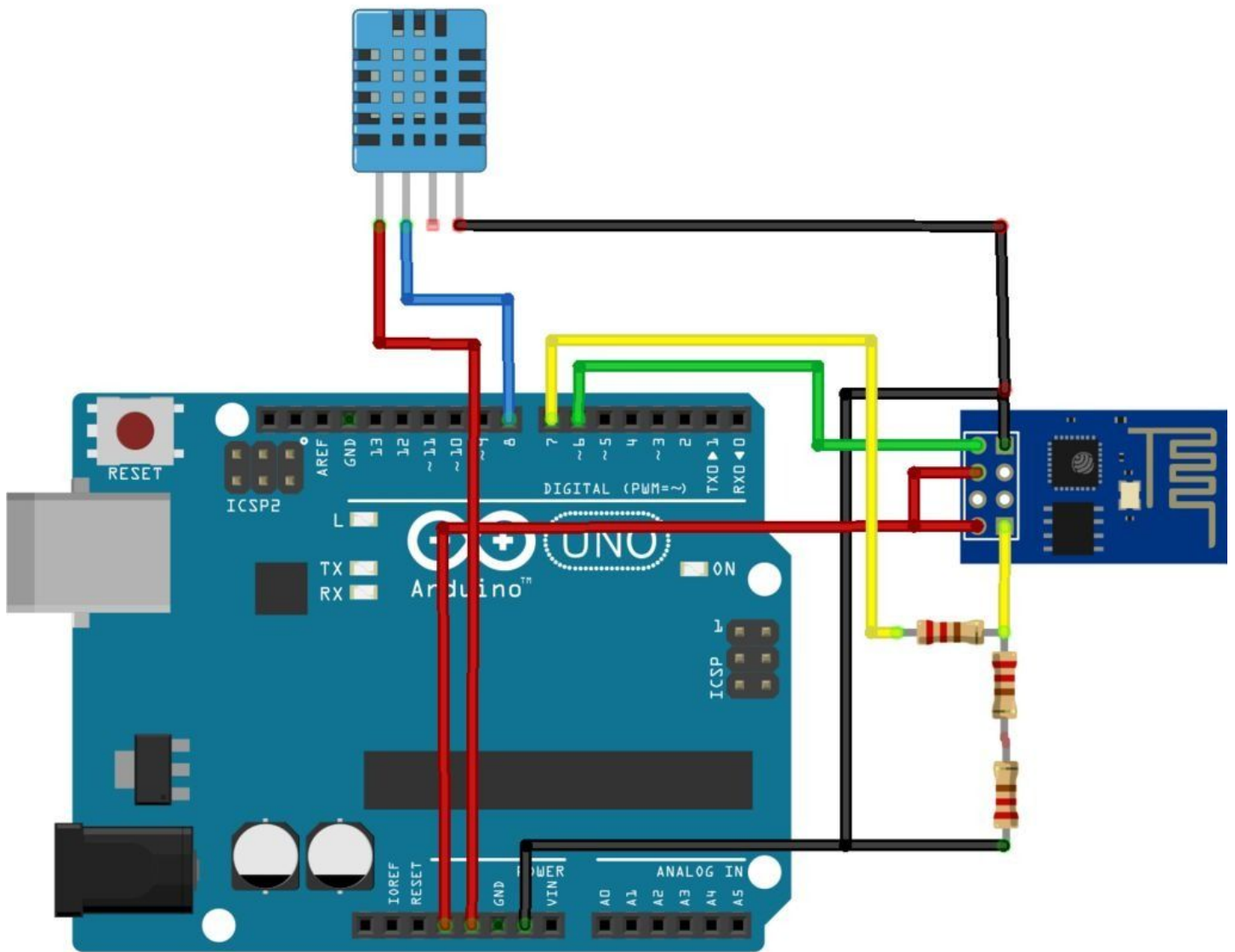


Abbildung 10: Ablauf Daten Senden/Lesen

## 5. Spezifikationen Hardware:

### 5.1. Schaltungsplan :



fritzing

Abbildung 11 :arduino mit einem Temperatursensor und WLAN Modul ESP8266 verbunden



### 5.2 Einzelteile:

#### \*\*Lilypad Bewegungssensor ADXL335:

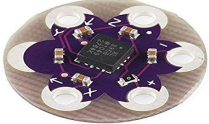


Abbildung 12 :Lilypad Bewegungssensor

DEV 09267, Lilypad-Beschleunigungssensor ADXL335 Beschreibung Dies ist ein dreiachsiger Beschleunigungssensor für das Lilypad-System. Basierend auf dem MEMS-Beschleunigungssensor ADXL335 von Analog Devices kann der Lilypad-Beschleunigungsmesser als Neigungs- und Vibrationsmesser verwendet werden. Der ADXL335 gibt an jeder der X-, Y- und Z-Achsen ein analoges Signal von 0 V bis 3 V aus. Jedes Lilypad wurde kreativ gestaltet, damit Sie eine Verbindung herstellen können. Verschiedene Eingangs-, Ausgangs-, Stromversorgungs- und Sensorplatinen sind verfügbar.

#### \*\*Herzschlagsensor:

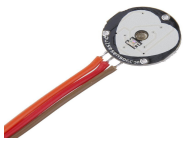


Abbildung 13 : Herzschlagsensor

Mit Pulssensoren kann auf Ihren Finger oder Ihr Ohr ein Lämpchen gelegt werden, um Ihre Herzfrequenz zu testen, indem die angeschlossene Leitfähigkeit mit dem Arduino verbunden wird. Studenten, Künstler, Sportler, Ersteller, Entwickler oder Mobile Endgerät können interaktive Arbeiten entwickeln, um die Herzfrequenz zu bestimmen.

Es hat auch eine Open-Source-App, mit der Ihr Herzfrequenzdiagramm in Echtzeit angezeigt werden kann.

Die Versorgungsspannung: 3,3 V ~ 5 V

#### \*\*Knopfzellen-Batteriehalter:



Abbildung 14 :Knopfzellen-Batteriehalter

verfügt über einen kleinen Schiebeschalter auf der Platine, mit dem Sie Ihr Projekt ausschalten und Batterien sparen können.

. Dieser Schiebeschalter ist der gleiche wie der auf der Lilypad-Schiebeschalter Platine, so dass ein versehentliches Aus- oder Einschalten nicht einfach ist.

### \*\*Temperatursensor:

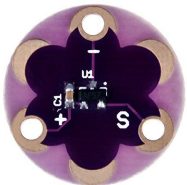


Abbildung 15 :Temperatursensor

445/5000

Dies ist ein kleiner Thermistor-Temperatursensor MCP9700. Wie wir alle wissen, war das Erkennen von Temperaturänderungen nie einfacher. Mit diesem Sensor können Sie die lokale Umgebungstemperatur ermitteln, indem Sie eine Analog-Digital-Wandlung auf der Signalleitung durchführen. Mit diesem kleinen Sensor kann er auch physische Berührungen anhand von Körperwärme und Umgebungsbedingungen erkennen. Dieser Sensor gibt 0,5 V bei 0 ° C, 0,75 V bei 25 ° C und 10 mV pro Grad C aus.

### \*\*LilyPad Arduino USB – ATmega32U4 Board:

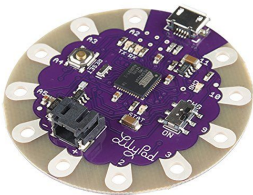
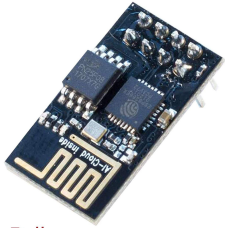


Abbildung 16 :Lilypad

Das LilyPad USB wird offiziell auch in der Arduino IDE ab Version 1.0.2 unterstützt.

Diese Tafeln sind so konzipiert, dass Sie Ihr nächstes Nähprojekt mit der Einstellung „Einfach halten“ erledigen können. Die einzige zusätzliche Hardware, die Sie für das LilyPad USB benötigen, ist ein Micro-USB-Kabel, da das neue IC über eine integrierte USB-Unterstützung verfügt.

\*\*esp8266 01 WLAN/WiFi Modul :



[P2-Delivery](#)

*Abbildung 17:ESP8266 01 WLAN Modul*

ESP8266 01 dank seines leistungsstarken 80-MHz-Prozessors und eines großen 1-MB-Speichers auch eigenständig ohne Arduino in C / C ++ Kann entweder direkt über das Arduino IDE-Programm gemountet werden.