

Pflichtenheft

Baby-Blooz Baby Überwachung

Autor:
Skandar, Mohamed, Aziz, Abdallah
Letzte Änderung: 03.05.2019
Dateiname: Pflichtenheft Baby-Blooz.docx

Version: 0.4

Tabellenverzeichnis

Versionshistorie	3
Vorhandene Dokumente	3
Überblick	4
Hauptziele	5
Annahmen und Abgrenzungen	5
Basic Workflow	6
Allgemein Workflow	7
Funktionalität	8
Überblick	8
Login	8
Sensordaten anzeigen	10
WLAN Einstellung	11
Benachrichtigung anzeigen	12
History anzeigen	12
Sensordaten speichern	13
Sensordaten Lesen(Lilypad)	13
Sensordaten an die API schicken	14
Sensordaten Lesen(API)	14
Sensordaten verarbeiten	15
Verarbeitete Daten an die App weiterleiten	15
7. Backlog	16
8. Lizenz	17

Abbildungsverzeichnis

Versionshistorie

Version:	Datum:	Verantwortlich	Änderung
0.1	02.05.2019	Skandar	Anpassen, Erweitern und Korrigieren
0.2	03.05.2019	Mohamed	Anpassen, Erweitern und Korrigieren
0.3	04.05.2019	Aziz	Anpassen, Erweitern und Korrigieren
0.4	05.05.2019	Abdallah	Anpassen, Erweitern und Korrigieren
0.5	09.05.2019	Mohamed,Aziz	Anpassen, Erweitern und Korrigieren
0.6	17.05.2019	Skanaer,Mohamed,Aziz Abdallah	Anpassen, Erweitern und Korrigieren

Tabelle 1: Versionshistorie

Vorhandene Dokumente

Dokument	Autor	Datum
Lastenheft	Aziz,Skandar, Mohamed,Abdallah	26.04.2019

Tabelle 2: Liste der relevanten Dokumente

1 Überblick

Im Rahmen des fachübergreifenden Projektes unseres Studiengangs absolvieren wir das Projekt, dessen Namen Baby Blooz ist, was um eine Software für Babyüberwachung geht.

Als Antwort auf das, am 26.04.2019, erhaltene Lastenheft werden wir mit diesem Dokument die Ziele, Haupt-Workflows und die Rahmenbedingungen des Projektes erläutern.

Es wird ein Modell eines smart Baby Suits erstellt, in dem ein Arduino Lilypad Board sowie verschiedene Messsensoren genäht werden, zusätzlich wird eine Mobile Monitoring Anwendung sowie eine Web-API und eine Datenbank entwickelt, indem Das Board die Sensordaten liest und an die Web-API weiterleitet. diese Letztere steht bereit ankommende Daten zu verarbeiten, in einer Datenbank zu speichern und an die ansprechende Client-Anwendung weiterzuleiten. Die Client-Anwendung steht ständig bereit, Daten zu erhalten, zu verarbeiten, auf einer Oberfläche anzuzeigen und nur die nötigen Push Nachrichten zu erstellen.

2 Hauptziele

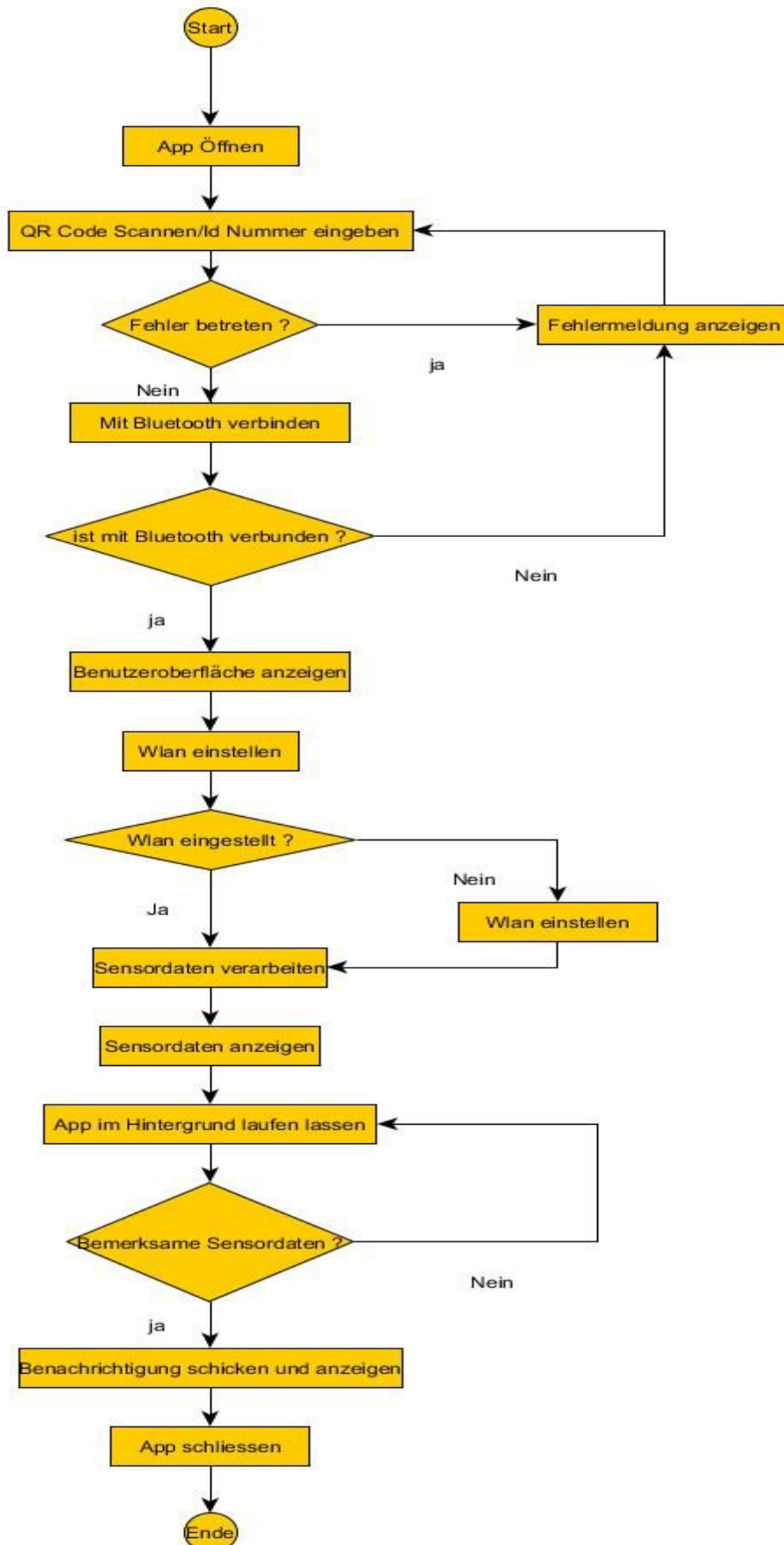
#	Ziel	Beschreibung
1	Korrekte und eindeutige daten der Sensoren darstellen	Sensoren zum Aufnehmen der Messwerte, und verarbeitung für die Darstellung
2	Messwerte der Sensoren Anzeigen	App-GUI zum Anzeigen
3	Werte der Sensoren in Datenbank speichern	Datenbank: mysql
4	Fast Echtzeitübertragung und stetige Aktualisierung der Daten	Baby-blooz Backend
5	Nur benötigte Push Nachrichten anzeigen.	Akustische und optische Benachrichtigung nach eindeutiger verarbeitung der Daten
6	Angenehmheit des Babys	Kein störendes Babysuit

3 Annahmen und Abgrenzungen

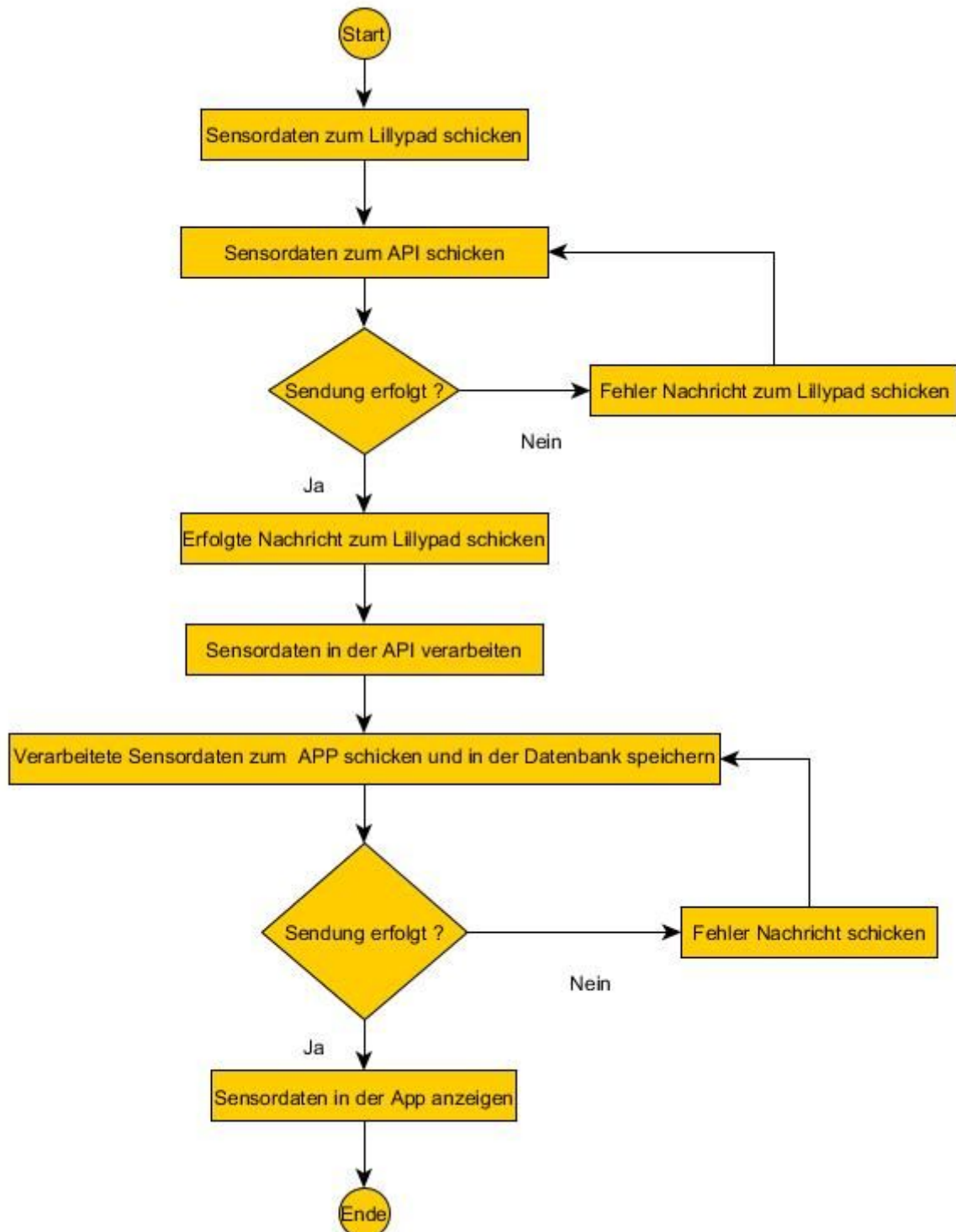
#	Annahmen
1	Mind. 2 Lilypad Boards oder das Budget dafür
2	Mind. 2 * Die für Lilypad passenden Sensoren oder das Budget dafür.
3	Thermosensor, Geräusche Sensor Accelerometer ausschließlich zu beachten

#	Abgrenzungen
1	Ausreichende Fehlermeldungen
2	Bluetooth verbindung
3	Einstellung der WLAN ohne Kabel

4 Basic Workflow



5 Allgemein Workflow



6 Funktionalität

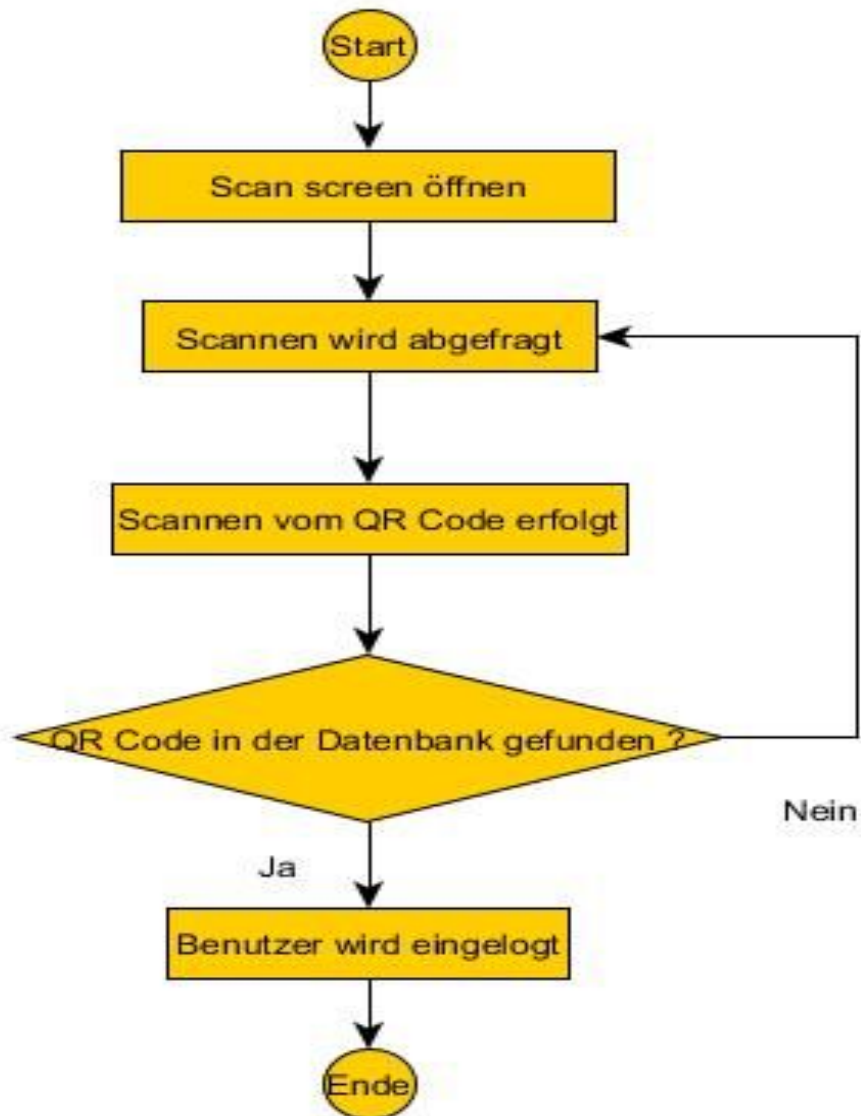
6.1 Überblick

Das Baby soll das Suit tragen. In jeder Suit gibt es ein QR Code und damit kann man das Suit mit der App verbinden. Das Suit hat 3 Sensoren und der Benutzer kann in jeder Zeit die aktualisierten gelesenen Werte von Sensoren anzeigen. Die Sensoren messen die Werte jede 30S und werden die Daten übers Lilypad zum Server schicken und dann werden sie zum App weitergeleitet. Die Daten werden immer in der App Aktualisiert und wenn der Benutzer die alten Werte sehen möchte, kann er mit Hilfe der App das abfragen und die Daten vom Server bekommen, die schon in einer Datenbank gespeichert sind.

Wenn die Daten sehr wichtig sind, dass es eine Gefahr das Baby betrifft, dann zeigt die App eine Benachrichtigung an.

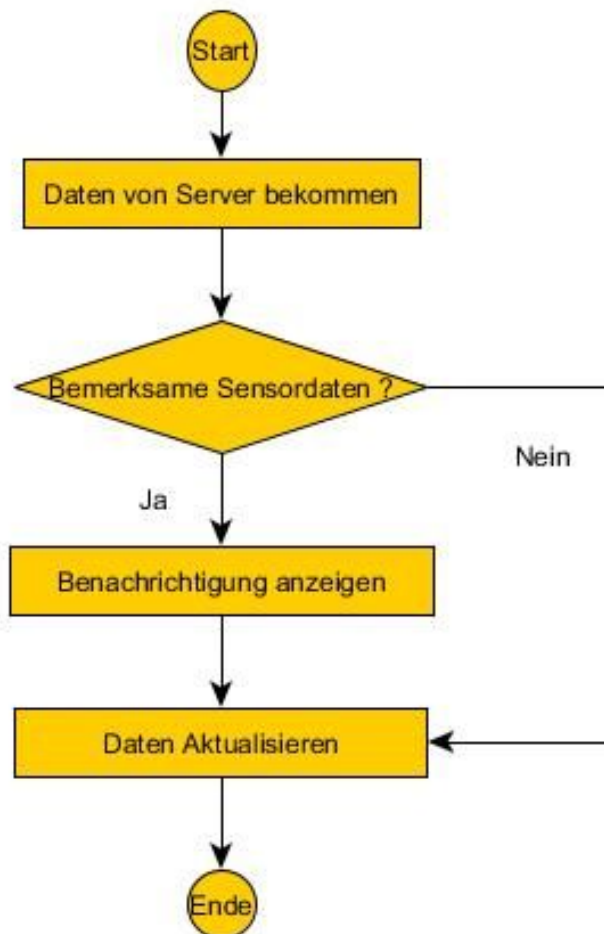
6.2 Login

Zweck/Ziel	Mit dieser Funktion kann sich der User ins System einloggen, um auf die Daten aus der Datenbank zugreifen zu können
Akteur/Auslöser	User
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none">• Verbindung zum Internet• App ist auf dem Handy installiert
Daten-Input	<ul style="list-style-type: none">• Benutzereingabe: QR code
Verarbeitungsschritte	<ol style="list-style-type: none">1. Scannen des QR-Codes oder UID eingeben2. Überprüfung ob QR-code vorhanden in der Datenbank3. Öffnen und Freischalten der Oberfläche der App wenn der Login erfolgreich ist.
Ergebnis	Erfolgreicher Login bedeutet, dass der User das WLAN vom LilyPad einstellen kann sowie die in der Datenbank gespeicherten Daten zugreifen kann.
Fehlerhandling	<ul style="list-style-type: none">• Warnung, wenn die Daten nicht ordnungsgemäß übertragen wurden.
Anforderung	#1, #18
Test Cases	<ul style="list-style-type: none">• das Einloggen mit korrektem QR-Code



6.3 Sensordaten anzeigen

Zweck/Ziel	Eindeutige anzeige der Sensordaten auf die Hauptaktivität der app.
Akteur/Auslöser	App
WF-Rererenz	
Vorbedingung	<ul style="list-style-type: none"> • Der User muss eingeloggt sein • Lilypad eingeschaltet • Lilypad mit internet verbunden • Daten verarbeitet vom Server
Daten-Input	
Verarbeitungsschritte	
Ergebnis	Eine Liste von den Sensordaten, die der User lesen und verstehen kann.
Fehlerhandling	<ul style="list-style-type: none"> • wenn ein Verbindungsfehler auftritt,dann den Benutzer benachrichtigen,
Anforderung	#6, #18, #21, #9
Test Cases	<ul style="list-style-type: none"> • Alle sensordaten in Diagrammen erstellen und analysieren



6.4 WLAN Einstellung

Zweck/Ziel	Ziel dieser Funktion ist, die einstellung des WLAN vom Lilypad
Akteur/Auslöser	User
WF-Rererenz	
Vorbedingung	<ul style="list-style-type: none"> • User eingeloggt • Verbindung zwischen dem Smartphone und Lilypad • Vorhandene WLAN Netzwerk
Daten-Input	<ul style="list-style-type: none"> • Netzwerk auswählen • Passwort eingeben
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Netzwerk Name und Passwort anhand der App eingeben 2. Senden des Passwortes und die MAC Adresse des Netzwerk über Kabel an das Lilypad 3. Lilypad versucht mit diesen Credential sich mit dem Netzwerk zu verbinden 4. Meldung über erfolgreicher oder fehlerhafte Verbindung an die App zurücksenden
Ergebnis	<ul style="list-style-type: none"> • Lilypad ist mit Internet über WLAN verbunden • Der Lilypad kann die Sensordaten an die API schicken
Fehlerhandling	
Anforderung	#11,#8, #14
Test Cases	<ul style="list-style-type: none"> • Ping request 8.8.8.8

6.5 Benachrichtigung anzeigen

Zweck/Ziel	Ein Push Benachrichtigung, wenn notwendig, anzeigen
Akteur/Auslöser	App
WF-Rererenz	
Vorbedingung	<ul style="list-style-type: none"> • User eingeloggt • Lilypad mit Internet verbunden • WebApi verfügbar
Daten-Input	
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Lesen der vom server ankommenden Daten 2. Wenn bemerkenswerte daten erhalten, Push Benachrichtigung ausführen. <p>PS: Die WebAPI entscheidet über die Wichtigkeit der Daten</p>
Ergebnis	Benachrichtigung über bemerkenswerte Daten
Fehlerhandling	<ul style="list-style-type: none"> • Dem Benutzer benachrichtigen, wenn ein Verbindungsfehler auftritt
Anforderung	#5, #7, #18, #21
Test Cases	<ul style="list-style-type: none"> • "dummy" wichtige Daten der App schicken. (mit Wichtig Flag) • "dummy" Akku leer Nachricht der App schicken.

6.6 History anzeigen

Zweck/Ziel	Der Benutzer ist immer in der Lage die gespeicherten Sensordaten an jedem Moment sich anzuschauen.
Akteur/Auslöser	App
WF-Rererenz	
Vorbedingung	<ul style="list-style-type: none"> • User eingeloggt • Lilypad mit Internet verbunden
Daten-Input	1. Datum und Zeit
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Datum und Zeit an die API schicken. 2. Die WebAPI schickt die Gespeicherten Sensordaten(beziehend des eingegebenen Datums und Zeit) an die APP 3. Die APP liest die Antwort und stellt die auf dem Bildschirm dar
Ergebnis	Die Geschichte der Sensordaten an jedem Moment abfragen und auf die UI anzeigen lassen
Fehlermeldung	<ul style="list-style-type: none"> • wenn ein Verbindungsfehler auftritt,dann den Benutzer benachrichtigen,
Anforderung	#21, #6
Test Cases	<ul style="list-style-type: none"> • "dummy" Daten abfragen und anzeigen.

6.7 Sensordaten speichern

Zweck/Ziel	Eindeutige Darstellung der Sensordaten in der Datenbank speichern
Akteur/Auslöser	WebAPI
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none"> • Daten von authentifiziertem Lilypad verfügbar • Korrektes Format der Daten • MySQL Server verfügbar und erreichbar • Verfügbare Datenbank
Data Input	
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Überprüfung des Formats der vom Lilypad angekommenen Sensordaten 2. Verbindung mit der Datenbank erstellen 3. Einfügen des Records in die Datenbank
Ergebnis	Daten sind in der Datenbank persistiert
Fehlermeldung	
Anforderung	#20, #6
Test Cases	<ul style="list-style-type: none"> • Musterdaten für die Temperatur und Herzschläge in der Datenbank speichern und danach abfragen.

6.8 Sensordaten Lesen(Lilypad)

Zweck/Ziel	Analoge Sensordaten lesen
Akteur/Auslöser	LilyPad
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none"> • Verbundene Sensoren mit der LilyPad Board • Lilypad eingeschaltet
Data Input	
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. PINS vom Lilypad initialisieren 2. PINS-Werten in einer Schleife lesen
Ergebnis	<ul style="list-style-type: none"> • Für jeden Sensor ein Wert gelesen
Fehlermeldung	
Anforderung	#2, #3, #4, #17
Test Cases	Werte mit den im Datenblatt definierten Intervallen vergleichen

6.9 Sensordaten an die API schicken

Zweck/Ziel	Analoge Sensordaten über tcp/ip schicken
Akteur/Auslöser	LilyPad
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none"> • Sensoren sind verbunden • Lilypad ist eingeschaltet • WLAN ist eingestellt • PINS sind initialisiert und funktionsfähig
Data Input	
Verarbeitungsschritte	1. Die gelesene Sensorwerten in Schleife jede 30 Sekunden an die API schicken in einer lesbar Format.
Ergebnis	<ul style="list-style-type: none"> • Erfolgsmeldung der Ermittlung von der API bekommen
Fehlermeldung	<ul style="list-style-type: none"> • Meldung über fehlerhaft ermittlung anzeigen
Anforderung	#11, #10
Test Cases	<ul style="list-style-type: none"> • Dummy POST request an die API senden und ein Erfolgsmeldung erwarten

6.10 Sensordaten Lesen(API)

Zweck/Ziel	Eindeutige Daten erstellen
Akteur/Auslöser	WebAPI
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none"> • Die Sensordaten sollen von authentifiziertes Lilypad verfügbar sein • Die Format der Daten soll Korrekt sein • Die Sensordaten sind vorhanden
Data Input	
Verarbeitungsschritte	1. empfangene Daten in einer lesebar Format umstellen
Ergebnis	<ul style="list-style-type: none"> • Verständliches Analyseprotokoll
Fehlermeldung	
Anforderung	#18, #20, #7
Test Cases	

6.11 Sensordaten verarbeiten

Zweck/Ziel	Eindeutige Daten erstellen
Akteur/Auslöser	WebAPI
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none"> Die Sensordaten sollen von authentifiziertes Lilypad verfügbar sein Die Format der Daten soll Korrekt sein
Data Input	
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Medizinischer Vergleich der Temperatursensor-daten: zwischen 36,5 und 37,5 normaler Bereich. Unter 36,5 / Über 37,5 = Unnormaler Bereich, als wichtige Daten merken. 2. Medizinischer Vergleich der Herzsensordaten: zwischen 125 und 144 Herzschlag : Normaler Bereich. sonst als wichtige Daten merken. 3. Ein normaler Bereich für das Geräusche und Bewegung mit dem Test Cases definieren. 4. Geräuschsensordaten und Bewegungssensordaten mit den definierten Bereich vergleichen : Wenn nicht inbegriffen, die Daten als wichtig merken. 5. Analyseprotokoll erstellen.
Ergebnis	<ul style="list-style-type: none"> Verständliches Analyseprotokoll
Fehlermeldung	
Anforderung	#18, #20, #7
Test Cases	<ul style="list-style-type: none"> Medizinischer Vergleich der Daten eines Sensors(manuell) und den normalen Wert für ein gesundes Baby. Medizinischer Vergleich der Daten aller Sensoren(manuell) und die normalen Werte für ein gesundes Baby. Vergleich der Ergebnisse mit dem erstellten Protokoll

6.12 Verarbeitete Daten an die App weiterleiten

Zweck/Ziel	Die Client-App bekommt eindeutige Sensordaten
Akteur/Auslöser	WebAPI
WF-Referenz	
Vorbedingung	<ul style="list-style-type: none"> Analyseprotokoll verfügbar Korrekte Format des Protokolls Verarbeitung der Sensordaten war erfolgreich
Data Input	
Verarbeitungsschritte	<ol style="list-style-type: none"> 1. Daten werden von API als objekt erstellt, dann werden diese Daten an die APP Front End geschickt .
Ergebnis	Erfolgsnachricht der Ermittlung von der App bekommen, wenn was bekommen wurde.
Fehlermeldung	
Anforderung	#5, #8, #10
Test Cases	

7. Backlog

Issue	Sprint	Assigned to	Due Date
Homepage der App Erstellen	1	Abdallah	31.05.2019
WebAPI-Endpoints bzw. Tests dafür Implementieren	1	Mohamed	31.05.2019
Erste Implementierung der Datenbank	1	Aziz	31.05.2019
Verbindung zwischen Lilypad und App Erstellen	1	Aziz,Mohamed,Skander	31.05.2019
Authentifizierung-Vorgang implementierung	1	Abdallah, Skander	31.05.2019
History Page Erstellen	2	Skander	22.16.2019
WebAPI Endpoint für Abfragen der History erstellen	2	Aziz	22.16.2019
Push-Benachrichtigunen implementieren	2	Mohamed	22.16.2019
Sensordaten Lesen(WebAPI)	2	Aziz	22.16.2019
Sensordaten in der Datenbank Speichern	2	Aziz	22.16.2019
Sensordaten Verarbeiten	2	Mohamed	22.16.2019
Sensordaten an die WebAPI Schicken	2	Abdallah, Skander	22.16.2019
LilyPad WLAN Einstellen	2	Abdallah	22.16.2019
Verbindung zwischen Lilypad und den Sensoren erstellen bzw. Werte Lesen	2	Mohamed, Aziz, Abdallah, Skander	22.16.2019
Test verarbeitung der Sensordaten(WebAPI)	3	Mohamed	12.07.2019
Test WLAN-Einstellung	3	Abdallah	12.07.2019
Test lesen der Sensordaten(WebAPI)	3	Aziz	12.07.2019
Test Pusch-Benachrichtigung	3	Mohamed	12.07.2019
Test Historypage	3	Skander	12.07.2019
Test Datenbank	3	Aziz	12.07.2019
Test Login	3	Abdallah	12.07.2019

8. Lizenz

Dieses Projekt ist **GNU GPL** lizenziert. Das erlaubt die Nutzung und die Änderung unsere Software.

Wir haben uns für die GNU Lizenz entschieden, da unser Programm weiter verwendet und entwickelt werden soll von jedem. Zudem ist durch diese Lizenz gewährleistet das auch in Zukunft Weiterentwicklungen oder Erweiterungen dieser Software unter der Open Source Lizenz laufen und somit für jeden Verfügbar sind.