

Qualitätssicherung HTW Berlin

Ihr Testdokument ist qualitativ nicht in Ordnung. Neben Formatierungsfehler etc. sind die Inhalte immer noch viel zu grob und nicht konkret genug. Sie geben sich keine richtige Mühe ein sauberes, leserliches und vollständiges Dokument zu liefern.

Wertung: 3,5 Punkte



Autor: Skander Baraket,Khalledi Aziz,Abdallah Tahboub,Mohammed ben Amar Letzte
Änderung: 28.06.2019
Dateiname:..Qualitätssicherung

Version: 2.0

Inhaltsverzeichnis

1	Testplan	4
2	Testfälle	4
2.1	QR_Code generator	4
2.2	Erste Implementierung der Datenbank	4
2.3	Authentifizierungsvorgang implementierung	5
2.4	Implementierung der Web API-Endpoints	5
2.5	Arduino Code	6
2.6	Sendung Konfigurationsdatei an Lilypad	6
2.7	Verarbeitung der Sensordaten	7
2.8	WLAN Einstellung	7
2.9	Push-Benachrichtigungen	7
2.10	History Page	8
2.11	Verbindung zwischen Lilypad und Sensoren	8
2.12	Sendung der Sensordaten an die API	8
2.13	Verbindung zwischen Lilypad und App	9
3	Testprotokoll	9
4	Anhang	10

Version Historie

<i>Version:</i>	<i>Datum:</i>	<i>Verantwortlich</i>	<i>Änderung</i>
0.1	06.06.2019	Baby Blooz team	Version 1
0.2	28.06.2019	Baby Blooz team	Version 2

Vorhandene Dokumente

Alle für die vorliegende Spezifikation ergänzenden Unterlagen müssen hier aufgeführt werden

Dokument	Autor	Datum
Lastenheft	Baby Blooz team	28.04.2019
Pflichtenheft	Baby Blooz team	17.05.2019
Technische Spezifikation	Baby Blooz team	07.06.2019

1 Testplan

2 Testfälle

2.1 QR_code generator

Testfall	Beschreibung
Testfall-Nummer	001
Teststart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	QR-Code generator
Testziel	QR-Code scannen
Testvoraussetzungen	QR-Code muss vorhanden sein .
Testfalldaten	QR-Code scannen
Erwartetes Verhalten	Der Link bekommen von dem gescannten QR-Code

QR_code beispiel:



Link : <http://hostexample/api/id?lilypadid=lilypadidexample>.

2.2 Erste Implementierung der Datenbank

Testfall	Beschreibung
Testfall-Nummer	002
Teststart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Erste Implementierung der Datenbank
Testziel	CRUD testen
Testvoraussetzungen	MySql server
Testfalldaten	Lesen der Sensordaten Update der Sensordaten Erstellen der Sensordaten Löschen der Sensordaten
Erwartetes Verhalten	Die Sensordaten werden gelesen Die Sensordaten werden aktualisiert Die Sensordaten werden gestellt Die Sensordaten werden gelöscht

Beispiel der Sensordaten ,die von der Lilypad geschickt sind:

Datas example sended by the lilypad to the api :

```
{
  "lilypadId": "lilypadidexample",
  "Movement":{
    "Xacceleration" : 30,
    "Yacceleration" : 60,
    "Zacceleration" : 50
  },
  "Heartbeat":70,
  "sound":452,
  "temperature":37
}
```

2.3 Authentifizierungsvorgang implementierung

Testfall	Beschreibung
Testfall-Nummer	003
Testart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Authentifizierungsvorgang implementieren
Testziel	Prüfen ob der QR-code vorhanden ist. Speichern der QR code in der APP.
Testvoraussetzungen	QR-code muss schon vorhanden sein.
Testfalldaten	QR-code prüfen QR-code speichern
Erwartetes Verhalten	QR-code wird geprüft. QR-code wird gespeichert.

Link der QR_code besipiel: <http://hostexample/api/id?lilypadid=lilypadidexample>

Bei der Prüfen sucht der API das Wort „Lilypad“ in dem Link und weiss er dann ,ob es richtig ein QR_code von der Lilypad ist oder nicht.

2.4 Implementierung der Web API-Endpoints

Testfall	Beschreibung
Testfall-Nummer	004
Testart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Implementierung der Web API-Endpoints
Testziel	Post und get von der Sensordaten.
Testvoraussetzungen	Fake Sensordaten
Testfalldaten	Sensordaten auslesen und zurückgeben.
Erwartetes Verhalten	API liest die geposteten Sensordaten und gibt die gefragte Sensordaten zurück.

Beispiel der Sensordaten ,die von der Lilypad geschickt sind:

Datas example sended by the lilypad to the api :

```
{
  "lilypadId": "lilypadidexample",
  "Movement":{
    "Xacceleration" : 30,
    "Yacceleration" : 60,
    "Zacceleration" : 50
  },
  "Heartbeat":70,
  "sound":452,
  "temperature":37
}
```


2.5 Arduino Code

Testfall	Beschreibung
Testfall-Nummer	005
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Implementierung des Arduino codes
Testziel	Arduino code funktioniert fehlerfrei
Testvoraussetzungen	Arduino ist Verbunden
Testfalldaten	Schicken der Sensordaten Lesen der Sensordaten
Erwartetes Verhalten	Arduino code funktioniert fehlerfrei

2.6 Sendung Konfigurationsdatei an Lilypad

Testfall	Beschreibung
Testfall-Nummer	006
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Sendung Konfigurationsdatei an Lilypad
Testziel	Konfigurationsdatei bei Lilypad zu erreichen
Testvoraussetzungen	Konfigurationsdatei
Testfalldaten	
Erwartetes Verhalten	Konfigurationsdatei muss bei Lilypad erscheinen


2.7 Verarbeitung der Sensordaten

Testfall	Beschreibung
Testfall-Nummer	007
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Verarbeitung der Sensordaten
Testziel	Alle Sensordaten medizinisch verarbeiten
Testvoraussetzungen	Sensordaten Von Lilypad bekommen
Testfalldaten	Sleep detection Cry detection
Erwartetes Verhalten	Sensordaten (Temperatur, Geräusche, Bewegung,) medizinisch verarbeiten 

2.8 WLAN Einstellung

Testfall	Beschreibung
Testfall-Nummer	008
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	WLAN Einstellen
Testziel	APP mit Lilypad über WLAN verbinden
Testvoraussetzungen	Internet
Testfalldaten	W LAN mit Lilypad verbinden W LAN mit App verbinden
Erwartetes Verhalten	Kommunikation zwischen Lilypad und dem App über WLAN ist vorhanden

2.9 Push-Benachrichtigung

Testfall	Beschreibung
Testfall-Nummer	009
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Push-Benachrichtigungen
Testziel	Push-Benachrichtigungen werden dem Benutzer gezeigt, wenn nötig ist
Testvoraussetzungen	Verarbeitete Sensordaten in API ist durchgeführt
Testfalldaten	Wichtige verarbeitete Sensordaten dem Benutzer zeigen 

Erwartetes Verhalten	Push-Benachrichtigungen sind zu sehen, wenn nötig ist
----------------------	---

2.10 History Page

Testfall	Beschreibung
Testfall-Nummer	010
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	History Page
Testziel	Sensordaten 90 Tage in der Vergangenheit anzeigen
Testvoraussetzungen	Sensordaten sind vorhanden
Testfalldaten	
Erwartetes Verhalten	History Page von 90 Tagen in der Vergangenheit Page History in Segmenten anzeigen: 7 Tage, 30 Tage, oder 90 Tage .ES ist von dem Benutzer einstellbar

2.11 Verbindung zwischen Lilypad und Sensoren


Testfall	Beschreibung
Testfall-Nummer	011
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Verbindung zwischen Lilypad und Sensoren
Testziel	Sensoren sind mit dem Lilypad verbunden
Testvoraussetzungen	WILAN
Testfalldaten	Lesen der Sensordaten
Erwartetes Verhalten	Fehlerfreie Verbindung zwischen Sensoren und Lilypad

2.12 Sendung der Sensordaten an die API

Testfall	Beschreibung
Testfall-Nummer	012
Teststart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Sendung der Sensordaten an die API
Testziel	Sensordaten werden von der API übernommen

Testvoraussetzungen	Verbindung zwischen Lilypad und der API
Testfalldaten	Sensordaten an API schicken
Erwartetes Verhalten	Alle Sensordaten werden von API erhalten 

2.13 Verbindung zwischen App und Lilypad

Testfall	Beschreibung
Testfall-Nummer	013
Testart	Funktionstest/Automated
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Verbindung zwischen App und Lilypad
Testziel	Kommunikation zwischen App und Lilypad über W LAN
Testvoraussetzungen	Internet Lilypad ist mit der Internet verbunden
Testfalldaten	Sensordaten an die API schicken Sensordaten an die Datenbank schicken Sensordaten an die APP schicken 
Erwartetes Verhalten	Datenaustausch zwischen Lilypad und der App

1 Testprotokoll

TestfallNr.	Datum	Status	Schweregrad	Datum 2. Lauf	Status 2. Lauf
01		durchgeführt			
02		durchgeführt			
03		durchgeführt			
04		durchgeführt			



2 Anhang

2.1 Fehler Kategorien

Für die Abnahme des Systems sind folgende Fehlerklassen definiert:

- **3 = Schwerer Mangel** Produktivsetzung nicht möglich (Nachhaltige Störung des Softwareablaufes mit daraus resultierender Funktionsuntüchtigkeit des Systems bzw. Störung von Systemteilen, die zur Störung aller Arbeitsabläufe beim Auftraggeber führt.)
- **2 = Mittlerer Mangel** Produktivsetzung möglich aber mangelhafte Funktionen nicht nutzbar (Durch eine Störung treten in Teilen der Programmabläufe nicht unerhebliche Störungen auf, so dass Teile der Software nicht verwendbar sind.)
- **1 = Leichter Mangel** Produktivsetzung durch Workaround mit vertretbarem Zusatzaufwand möglich (Alle anderen als die in den vorstehenden Prioritätsgraden beschriebenen Störungsbilder)