

Earables (Pflichtenheft)

uuoof

14. November 2019

Projekt: Earables
Auftraggeber: Karlsruhe Institute of Technology (Teco)
Auftragnehmer: wir

Version	Datum	Autor(en)
1.0	14. November 2019	uuoof

Inhaltsverzeichnis

1	Einleitung	1
2	Zielbestimmung	1
2.1	Musskriterien	1
2.2	Wunschkriterien	1
2.3	Abgrenzungskriterien	2
3	Produkteinsatz	2
3.1	Zielgruppe	2
3.2	Anwendungsbereiche	2
3.3	Betriebsbedinugen	2
4	Produktumgebung	2
4.1	Hardware	2
4.2	Software	3
5	Funktionale Anforderungen	3
5.1	Mussanforderungen	3
5.1.1	Bibliothek	3
5.1.2	Erweiterungsmodul	3
5.1.3	App	3
5.2	Wunschanforderungen	4
5.2.1	Erweiterungsmodul	4
5.2.2	App	4
6	Produktdaten	5
7	Nichtfunktionale Anforderungen	5
7.1	Laufzeitverhalten	6
7.2	Speicherplatz	6
7.3	Stabilität	6
8	Systemmodelle	6
8.1	Architekturdiagramm	6
8.2	Use-Case-Diagramme	6
9	Benutzeroberfläche	6
10	Qualitätszielbestimmungen	6
11	Globale Testfälle und Szenarien	6
11.1	Globale Testfälle	6
11.1.1	Mussanforderungen	6
11.2	Szenarien	7
11.2.1	Bibliotheksnutzung	7
12	Entwicklungsumgebung	7
13	Glossar	8

1 Einleitung

Earables gehören zu den Wearable Computings. Sie sind also nichts anderes als intelligente Kopfhörer. Je nach dem wie sie ausgestattet werden bringen sie die unterschiedlichsten Funktionen mit sich. Neben der klassischen Ausstattung von Lautsprechern, Mikrophon und Bluetooth Low Energy besitzen sie in unserem Fall zusätzlich ein 6-Achsen IMU (Beschleunigungssensor und Gyroskop) . Mit seiner Hilfe ist man in der Lage die (Kopf-)Bewegung des Nutzers zu tracken um mit den gewonnenen Messdaten beispielsweise die Atemfrequenz zu ermitteln. Earables bilden also eine Schnittstelle von Mensch und Computer. Mit ihnen kann man unauffällig medizinische Daten im Alltag sammeln. Da sie kaum von normalen Bluetooth Kopfhörern zu unterscheiden sind haben sie bereits ein hohe soziale akzeptanz in der Gesellschaft erlangt, im Gegensatz zu beispielsweise den smart Glasses. Mit der Entwicklung und Forschung dieser Art von Earables beschäftigt sich das globale eSense Projekt von Nokia Bell Labs in Zusammenarbeit mit dem Telecooperation Office (TECO). Hier wird gemeinsam nach möglichen Anwendungsfällen der Earables gesucht, wodurch auch dieses Projekt seinen Weg in diese Welt gefunden hat.

2 Zielbestimmung

Die Ziele dieses Projektes lassen sich in drei Bereiche gliedern:

1. Es soll eine Cross-Plattform Bibliothek (Android/IOS) für die Earables der Plattform eSense entwickelt werden mit der es möglich ist, die Messdaten des Gerätes aufzuzeichnen und verschiedene Steuerungsparameter zu verändern.
2. Es soll ein Erweiterungsmodul entwickelt werden, mit der es Möglich ist gewisse Daten der Kopf auszuwerten, beispielsweise ob der Nutzer gerade läuft oder steht.
- 3.

2.1 Musskriterien

Das Ziel der Bibliothek ist es, Kern-Funktionen zu beinhalten, um Messdaten auszulesen und die Steuerungsparameter zu verändern. Besonders wichtig ist, dass man in der Lage sein soll die Daten des 6-Achsen IMU auszulesen. Bezüglich des Erweiterungsmoduls soll es in der Lage sein zu erkennen ob der Nutzer gerade läuft oder steht und dies, über eine ansprechende GUI, auszugeben. Außerdem soll die GUI so benutzerfreundlich gestaltet werden, so dass der Benutzer intuitiv weiß wie man die Steuerungsparameter der Earables verändern kann.

2.2 Wunschkriterien

Als Wunschkriterium soll die App die Schrittfrequenz des Nutzers anzeigen und das im besten Fall in Echtzeit. Mit einer kurzen Verzögerung von ein paar Sekunden wird jedoch gerechnet. Zusätzlich soll die Anzahl der Schritte ausgegeben werden wodurch man die zurückgelegten Distanz ermitteln kann. Diese wird ebenfalls ausgegeben. Des weiteren sind 3 verschiedene Modi geplant. Im Modus „Counter“ zählt die App wie viele Liegestützen oder Sit Ups der Nutzer macht. Über den Modus „Listen and Perform“ ist der Nutzer in der Lage seinen eigenen Trainingsplan individuell für sich zu erstellen. Dieser wird dann,

mit Hilfe von Text to Speech ausgegeben. Im letzten Modus Namens „Musik Run“ wird die Musik automatisch gestoppt, wenn der Nutzer steht und wieder gestartet, sobald der Nutzer damit beginnt weiter zu laufen. Die App wird auf deutsch sein, aber der Nutzer kann weitere Sprachen durch Sprachpaket-Plugins hinzufügen. Zum Abschluss soll es für den Nutzer möglich sein seine Trainingsdaten zu speichern, importieren und exportieren, um so mehrere „Projekte“ anzulegen.

2.3 Abgrenzungskriterien

Was wir nicht umsetzen wollen ist die Speicherung von Rohdaten, da dies zur Folge haben könnte, dass sich einige Nutzer ausspioniert fühlen. Generell wird die Speicherung von Daten nur auf dem Smartphone erfolgen und nicht auf dem Speicher der Earables. Das Ziel, die Geschwindigkeit der Musik an die Schrittfrequenz des Nutzers anzupassen, wird nicht verfolgt. Im Modus Liegestützen oder Sit Ups zählen wird davon ausgegangen, dass der Nutzer wirklich nur Liegestützen oder Sit Ups macht und nicht versucht das System auszutricksen.

3 Produkteinsatz

3.1 Zielgruppe

- Bibliothek: Softwareentwickler
- Erweiterungsmodul: Softwareentwickler
- App: Hobbysportler

3.2 Anwendungsbereiche

- Bibliothek Softwareentwicklung für eSense Wearables
- Erweiterungsmodul Softwareentwicklung im Bereich Schritterkennung
- App Heimtraining, Sport,...

3.3 Betriebsbedinugen

TODO

- Bibliothek
- Erweiterungsmodul
- App

4 Produktumgebung

4.1 Hardware

Minimale Anforderungen: Smartphone mit Bluetooth LE Unterstützung.

4.2 Software

Betriebssystem: Android > 8.0 und iOS > 12.0

5 Funktionale Anforderungen

TODO Valle Koordination

5.1 Mussanforderungen

TODO alle: Man sollte hier funktionale Anforderungen **angeben und ergänzen**. Dabei nummeriert man in 10er Schritten.

5.1.1 Bibliothek

/F010/ Daten des Gyroskops auslesen und in Echtzeit zur Verfügung stellen.

/F020/ Daten des Beschleunigungssensors auslesen und in Echtzeit zur Verfügung stellen.

/F030/ Messparameter (Abtastrate etc.) ändern.

/F040/ Datenaufnahme des IMU starten und stoppen.

/F050/ BLE Charakteristiken unterstützen

5.1.2 Erweiterungsmodul

Auswertung der ausgelesenen Daten:

/F060/ Schritterkennung (Stehen/Laufen des Nutzers).

5.1.3 App

/F070/ App starten: Der Nutzer kann die App über sein Smartphone starten.

/F080/ Modus auswählen: Auswahl eines Modus aus allen verfügbaren Modi (also mindestens der Modus „Laufen/Stehen anzeigen“).

/F090/ Modus wechseln: Wechseln zwischen Modi während ein Modus aktiv ist.

/F100/ Laufmodus: Liveanzeige, ob Nutzer gerade „läuft“ oder „steht“.

/F110/ Vorgang starten: Der Nutzer kann den modusspezifischen Vorgang starten. Dann wird der Modus nach seiner Beschreibung aktiv ausgeführt. Dies gilt für jeden Modus außer den Modus „Livedaten“.

/F120/ Vorgang stoppen: Der Nutzer kann den modusspezifischen Vorgang stoppen.

/F130/ Resultat anzeigen: Nach Stoppen Anzeigen des Vorgangsergebnisses bis neuer Vorgang gestartet wird.

5.2 Wunschanforderungen

5.2.1 Erweiterungsmodul

Weitere Datenauswertung:

/F140/ Schrittfrequenzerkennung: Erkennung der Schrittfrequenz des Nutzers.

/F150/ Distanzmessung: Erkennung der zurückgelegten Distanz des Nutzers.

/F160/ Schrittzahlgähler: Zählen der Schritte des Nutzers.

/F170/ Erkennung Situps: Erkennung von Situps.

/F180/ Erkennung Liegestütze: Erkennung von Liegestütze.

5.2.2 App

Weitere Modi:

/F190/ Modus Livedaten: (*versteckt*) Visualisieren der Sensorrohdaten als Graphen.

/F200/ Zählmodus: Zählen von Liegestützen oder Sit-ups.

/F210/ Start/Stopp Musikmodus: Musik stoppt wenn Nutzer stehen bleibt, läuft wenn der Nutzer läuft.

/F220/ Modus Lauschen&Agieren: Zusammenstellen eines Trainingsablaufs (Liegestütze, Sit-ups, Laufen).

/F230/ Modus Lauschen&Agieren: Sprachanweisungen für die nächste Übung während des Trainings.

/F240/ Modus Lauschen&Agieren: Anzeige der Zeitdauer jeder Übung nach Ablaufende, siehe F130.

/F250/ Einstellungen: Der Nutzer kann seinen Namen abgeben und ändern.

/F260/ Einstellungen: Der Nutzer kann die Sprache der App anpassen.

/F270/ Einstellungen: Der Nutzer kann die gespeicherten Trainingsdaten löschen.

/F280/ Einstellungen: Der Nutzer kann die Steuerungsparameter anpassen.

/F290/ Erstnutzung: Aufforderung der Angabe von Name und in-App Sprache bei Erstnutzung.

/F300/ Exportieren und Importieren von Trainingsdaten: Der Nutzer kann in der App seine gesamten Trainingsdaten exportieren und importieren.

/F310/ Speichern der Vorgänge: Speicherung aller Vorgänge die aktiv waren mit Datum, Uhrzeit und dem Wert, der der Funktion entspricht (keine Rohdaten).

6 Produktdaten

- /PD010/ Es werden keine rohen Messdaten gespeichert.
- /PD020/ Die Einstellungen (Sprache, Steuerungsparameter, Messeinstellungen) sind zu speichern.
- /PD030/ Es sind relevante Daten über den Nutzer, wie den Benutzernamen, zu speichern.
- /PD040/ Die Trainingsdaten (Bestenliste und letzte Trainingseinheit) sind zu speichern.
- /PD050/ Speicherung aller Vorgänge, die aktiv waren mit Datum, Uhrzeit und dem letzten Resultat (/F300/) (TODO abklären, ob umsetzen)

7 Nichtfunktionale Anforderungen

TODO Jan Koordination Glossareintrag „live“: statt Echtzeit lieber „live“ verwenden, live := mit einer Verzögerung von maximal 3 Sekunden nur zum brainstorm alle Punkte zu denen es sachen geben könnte: Mussanforderungen Bibliothek Erweiterungsmodul App Wunschanforderungen Erweiterungsmodul App

- /NF010/ Beim Ausführen der Funktion /F030/ soll die Datenaufnahme der IMU gestoppt werden.
- /NF020/ Die Funktionen /F060/ und /F100/ (Laufmodus) soll maximal eine Verzögerung von zwei Sekunden aufweisen.
- /NF030/ Beim Wechseln zwischen Modi (/F090/) soll der aktuelle Modus terminiert werden.
- /NF040/ Der Start eines Modus nach seiner Auswahl (/F080/) soll nicht länger als zwei Sekunden benötigen.
- /NF050/ Die Einblendung des Vorgangsergebnisses (/F130/) soll nicht länger als zwei Sekunden benötigen.
- /NF060/ Nach Ausführung der Funktion Sprachänderung (/F260/) muss die App neu gestartet werden.
- /NF070/ Bei negative Anzahl Angaben (/F220/) wird das Starten des Vorgangs verhindert.
- /NF080/ Bei Namensänderung (/F250/) werden gespeicherte Daten (siehe ev. /F300/) nicht verändert.
- /NF090/ Bei Namensgebung sind nur Groß- und Kleinbuchstaben ohne Umlaute und Sonderzeichen erlaubt.

7.1 Laufzeitverhalten

7.2 Speicherplatz

7.3 Stabilität

8 Systemmodelle

TODO Jan

8.1 Architekturdiagramm

8.2 Use-Case-Diagramme

9 Benutzeroberfläche

TODO Benni auseinandersetzen

10 Qualitätszielbestimmungen

11 Globale Testfälle und Szenarien

TODO David

11.1 Globale Testfälle

11.1.1 Mussanforderungen

Schritterkennung beim Gehen

1. Das Smartphone wird per Bluetooth mit den Kopfhörern verbunden.
2. Die App wird gestartet.
3. Nach dem Startvorgang der App wird in den Modus „Laufmodus“ (/F100/) gewechselt.
4. Die Kopfhörer werden korrekt am Ohr des Nutzers angebracht.
5. Die App zeigt dem Nutzer den Status „stehend“ an.
6. Sobald der Nutzer anfängt zu gehen zeigt die App „gehend“ an.
7. Sobald der Nutzer wieder still steht ändert sich der Zustand wieder zurück zu „stehend“.

Start/Stop Musikmodus

1. Das Smartphone wird per Bluetooth mit den Kopfhörern verbunden.
2. Der Nutzer spielt Musik mit der vorinstallierten Musik-App ab.
3. Die App wird gestartet.
4. Nach dem Startvorgang der App wird in den Modus „Start/Stop Musikmodus“ (/F210/) gewechselt.
5. Die Kopfhörer werden korrekt am Ohr des Nutzers angebracht.
6. Der Nutzer beginnt zu gehen.
7. Der Nutzer startet den gewählten Modus.
8. Der Nutzer hört auf zu gehen.
9. Die Musik wird pausiert.
10. Der Nutzer geht weiter.
11. Die Musik startet automatisch wieder.

11.2 Szenarien

11.2.1 Bibliotheksnutzung

12 Entwicklungsumgebung

TODO valle

Wir arbeiten an dem Projekt mit Visual Studio 2019, sodass alle eine einheitliche Entwicklungsumgebung verwenden. Dabei kommt der .NET Standard 2.0 zum Einsatz, der C# 7.2 verwendet. Wir arbeiten außerdem mit Xamarin Forms.

Zur Versionskontrolle und zur Projektübersicht wird Git verwendet, das Repository liegt öffentlich auf Github¹.

¹<https://github.com/vlle1/earablesKIT>

