

LIAR - Lügendetektor

Semesterprojekt im Fach Mobile Applications for Public Health

Björn Ahlfeld, Patrick Borck, Jens Grundmann,
Sebastian Lun, Daniel Pinkpank, Phillippe Wels

2. Dezember 2013

Inhaltsverzeichnis

1	Projektplanung	5
1.1	Meilensteine	6
1.1.1	Einarbeitung	6
1.1.2	Aufbau eines User Interfaces	6
1.1.3	Anbindung des EEG Sensors	7
1.1.4	Anbindung des Galvanic Skin Sensors	7
1.2	fiktive Zeitungsmeldung	8
1.3	Produktverpackung	9
2	Personas und Anwendungsszenarien	10
2.1	Persona 1 - „Der Kontrollfreak“: Elisa Schubert (20)	10
2.1.1	Soziodemografische Daten	10
2.1.2	Vorlieben, Hobbys, Abneigungen	11
2.1.3	Nutzererwartung an das Produkt	11
2.1.4	Anwendungsszenario	11
2.2	Persona 2 - „Der Wissenschaftler“: Frank Bollwerker (36)	12
2.2.1	Soziodemografische Daten	12
2.2.2	Vorlieben, Hobbys, Abneigungen	12
2.2.3	Nutzererwartung an das Produkt	13
2.2.4	Anwendungsszenario	13
2.3	Persona 3 - „Der Angeber“: Jonas Keppler (29)	13
2.3.1	Soziodemografische Daten	14
2.3.2	Vorlieben, Hobbys, Abneigungen	14
2.3.3	Nutzererwartung an das Produkt	14
2.3.4	Anwendungsszenario	14
3	Anforderungen	15
4	priorisierte User Stories	16
4.1	User Stories mit hoher Priotität:	16
4.2	User Stories mit mittlerer Priorität:	16
4.3	User Stories mit niedriger Priorität:	16

Abbildungsverzeichnis

5	Risikobetrachtung	17
6	Systemüberblick und Systemarchitektur	18
7	Entwurf / Mockup des User-Interface	19
8	Implementierungen	21
8.1	Arduino	25
8.1.1	eHealth - Plattform	25
8.1.2	Wireless + Bluetooth Shield	27
8.1.3	Endprodukt	28
8.2	Android	30
8.2.1	EEG: NeuroSky Brainwave Headset	30
8.2.2	Arduino Bluetooth	32
8.2.3	Spielaufbau	34
8.2.4	Datenpersistenz	34
8.2.5	Datenvisualisierung	34

Abbildungsverzeichnis

1	Zeitungsmeldung	8
2	Produktverpackung	9
3	Systemarchitektur	18
4	Mockup Hauptbildschirm	19
5	Mockup Bildschirm für neues Spiel	19
6	Mockup Spielbildschirm	20
7	Mockup Bildschirm Fragensauswertung Wahrheit	20
8	Mockup Bildschirm Fragensauswertung Lüge	21
9	Mockup finaler Bildschirm zur Fragensauswertung	21
10	Mockup highscore-Bildschirm	22
11	Mockup Badge-Bildschirm	22
12	Mockup Bildschirm Eigenanalyse	23
13	Mockup Bildschirm zum Laden gespeicherter Daten	23
14	Mockup Infobildschirm Hautleitwert	24
15	Mockup Infobildschirm EEG	24

Tabellenverzeichnis

16	Mockup Anleitungsbildschirm	25
17	Arduino-eHealth-Stack	26
18	Das Bluetooth Modul	27
19	Der Ardunio-Bluetooth-Stack	28
20	Der vollständige Stack	29
21	Das NeuroSky Brainwave Headset	31

Tabellenverzeichnis

1	Risikoanalyse	17
---	-------------------------	----

1 Projektplanung

Die zu entwickelnde Android Applikation LIAR ist ein Gesellschaftsspiel auf Basis eines Lügendetektors. Die Applikation soll es ermöglichen, ein Nutzer A einem anderen Nutzer B Fragen stellt, welcher dieser beantwortet. Dabei wird mittels eines EEG-Sensors und mit Hilfe eines Galvanic Skin Sensors die Anspannung des Nutzers evaluiert. Im Anschluss erlaubt die LIAR App eine Aussage zum Wahrheitsgehalt der Antwort des Nutzers.

1.1 Meilensteine

Meilensteine repräsentieren Zwischenergebnisse in der Programmentwicklung die von besonderer Bedeutung sind. Sie eignen sich zur Arbeitsaufteilung in einer Gruppe.

1.1.1 Einarbeitung

- Android SDK installieren
- Android Tutorial absolvieren
- Einarbeitung in Mindwave Mobile (EEG-Sensor)
- Einarbeitung in das SDK des Galvanic Skin Sensors

1.1.2 Aufbau eines User Interfaces

- Hauptbildschirm mit den Elementen(neues Spiel starten, Eigenanalyse, Ranglisten und Spielanleitung)
- Spielstartbildschirm auf dem die Anzahl der zu spielenden Runden, die Anzahl der Spieler und deren Namen eingetragen werden können.
- Eigenanalysebildschirm: zeigt die Messwerte beider Sensoren in zwei Graphen an und erlaubt eine Aufzeichnung von Daten, eine Ansicht älterer Daten, sowie zwei Informationsbuttons mit Wissenswerten Informationen zum EEG und Galvanic Skin Sensor
- Auswertungsbildschirm: zeigt das Ergebnis der Fragerunde (Anzahl der wahrheitsgemäß beantworteten Fragen, Anzahl der vermeintlich nicht wahrheitsgemäß beantworteten Fragen, Batch-System für das Erreichen einer besonde-

ren Leistung [z.B. ehrliche Haut oder Lügner des Monats]) und ermöglicht das Teilen des Ergebnisses auf Facebook

1.1.3 Anbindung des EEG Sensors

- Einrichtung einer Bluetoothverbindung zum EEG-Sensor
- Auslesen der Daten des EEG-Sensors
- Interpretieren der Daten des EEG-Sensors
- grafische Darstellung der Daten des EEG-Sensors

1.1.4 Anbindung des Galvanic Skin Sensors

- Einrichtung einer Bluetoothverbindung vom Arduino Shield zum Galvanic Skin Sensor
- Einrichtung einer Bluetoothverbindung vom Smartphone zum Arduino Shield
- Auslesen der Daten der Messdaten
- Interpretieren der Messdaten
- grafische Darstellung der Messdaten

1.2 fiktive Zeitungsmeldung

Die folgende fiktive Zeitungsmeldung stellt aus Sicht der Entwickler die Perspektiven und Erwartungen an das LIAR Projekt dar.

Neues aus der Medizintechnik

Bild der Technik 2 / 3

Mit EEG und Galvanic Schwindler entlarven

Berliner Start-Up-Unternehmen will mit der „Wahrheit“ an den Spielmarkt

Das Berliner Start-Up-Unternehmen XYZ bestehend aus sechs Hochschulabsolventen der HTW-Berlin will Ende Februar 2014 mit einem EEG-Messgerät und Galvanic-Skin-Sensor ein neuartiges Spiel auf den Markt bringen. Dabei handelt es sich um ein Lügendetektor, der über ein handelsübliches Smartphone angesprochen wird. Derzeit nur für Android-Geräte verfügbar, aber man arbeitet bereits an einer iPhone bzw. iPad-Version.

Wir sprachen mit den Newcomern über ihr neues Produkt in Berlin: Laut ihrer Vision soll es ein neuartiges Gesellschaftsspiel werden, dass es noch nicht in diesem Umfang gegeben hat. Die Idee besteht darin, Gehirnschans mit einem EEG-Messgerät und die Hautoberflächenspannung mittels Galvanic Skin Sensor zu messen und mit bekannten Werten zu ver-

gleichen. Das zusätzliche Verwenden von Gehirnschans ist neuer, aber mittlerweile kein unbekanntes Verfahren mehr. „Bei der Ermittlung einer Lüge sind wesentlich mehr Gehirnanteile aktiv im Vergleich zu einer wahren Antwort“ so Phillippe Wels, Entwickler bei XYZ. Auf die Frage, warum gerade ein Gehirnströmesensor verwendet wird meint Herr Wels weiter „die Ergebnisse sind genauer, im Vergleich zur Nutzung eines einzelnen Sensors“.

„Wir wollen ein Produkt für Jedermann - kein eingeschränktes Medizinprodukt“ führt Herr Wels weiter aus, aber „es gibt Nutzungsmöglichkeiten für klinische / medizinische Einsätze, z.B. für Psychiatrien oder Selbsthilfegruppen“. „Wir wollen einen Prototypen schaffen für neue Innovationen und Anwendungsbereiche, möchten aber das noch nicht weiter konreti-

sieren“ so Phillippe Wels.

Auf die Frage, welche Bedarfe die Anwender haben, meint Herr Wels: „wir wollen die wissenschaftliche Neugier des Kunden wecken und Nutzungsmöglichkeiten der Medizin in die Haushalte bringen. Des Weiteren haben wir Vorüberlegungen, das Ganze als Open-Source-Projekt in Form eines Frameworks zur Verfügung zu stellen.“ „Es gibt aber auch Bestrebungen bzw. Nutzungspotenziale im Privatbereich, z.B. bei der Geräteteuerung oder im Multimedia-Bereich“ meint Patrick Borck, ebenfalls Entwickler.

Welche Wirkung das Unternehmen mit ihrem Produkt auf dem Spielmarkt erreichen wird ist kontrovers diskutiert. Somit bleiben nur die Verkaufszahlen abzuwarten. dpa ■

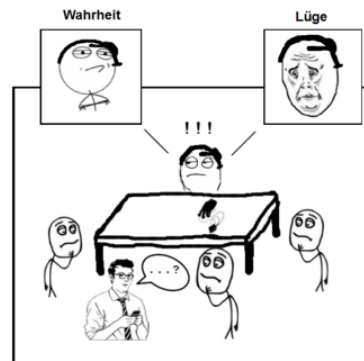


Abbildung: Das Spiel - ein Teilnehmer wird gefragt und seine Daten werden zur Wahrheit oder Lüge ausgewertet

Abbildung 1: fiktive Zeitungsmeldung

1.3 Produktverpackung

Die folgende Abbildung stellt die auf die Nutzergruppe angepasste Produktverpackung der LIAR-App dar.



Abbildung 2: Produktverpackung Liar Android App

2 Personas und Anwendungsszenarien

Dieses Kapitel befasst sich mit Personas und deren Anwendungsszenarien. Eine Persona ist eine Person, die eine Gruppe von zukünftigen Nutzern der Applikation beispielhaft repräsentiert. Personas haben konkrete Eigenschaften und zeigen in einem Anwendungsszenario, wie sie mit dem Produkt umgehen wollen.

2.1 Persona 1 - „Der Kontrollfreak”: Elisa Schubert (20)



2.1.1 Soziodemografische Daten

Elisa Schubert ist 20 Jahre alt. Sie ist ledig und zur Zeit in einer festen Beziehung. Vor einem Jahr hat sie ihr Abitur gemacht. Aktuell studiert sie die Fächer Deutsch und Biologie an der Universität zu Köln.

2.1.2 Vorlieben, Hobbys, Abneigungen

Frau Schubert singt in einem Gospel-Chor, geht gern auf Partys und engagiert sich in ihrer Freizeit ehrenamtlich bei der Naturschutzorganisation WWF. Sie ist notorisch eifersüchtig auf jede Frau, die sich Ihrer Jugendliebe Bernd nähert. Da Bernd auch dafür bekannt ist, „mehrgleisig“ unterwegs zu sein, will sie immer wieder Gewissheit, dass er nur sie liebt. Frau Schubert hasst es belogen zu werden. Sie ist ein Kontrollfreak und liest heimlich die SMS von Bernd. Für ihre Zukunft hat sich Elisa vorgenommen, mit Bernd eine Familie gründen und Ihr Studium erfolgreich zu beenden.

2.1.3 Nutzererwartung an das Produkt

Elisa ist neugierig was andere Menschen, insbesondere ihre Freunde, über sie denken. Sie erwartet von der LIAR App, dass sie Bernd besser kontrollieren kann und erhofft sich Einblicke in die verborgene Gedankenwelt ihrer Freunde.

2.1.4 Anwendungsszenario

An einem gemütlichen Samstag Abend spielen Elisa, Bernd und einige Freunde Karten- und Gesellschaftsspiele. Elisa hat die LIAR App, samt den dazugehörenden Sensoren mitgebracht und stellt sie ihren Freunden vor. So ist sie in der Lage auf unauffällige Art und Weise Bernd Fragen zu stellen und seine Antworten auf den Wahrheitsgehalt hin zu überprüfen. Die anderen Freunde stellen einander ebenfalls Fragen und haben dabei viel Spaß.

2.2 Persona 2 - „Der Wissenschaftler”: Frank Bollwerker (36)



2.2.1 Soziodemografische Daten

Frank Bollwerker ist 36 Jahre alt, verheiratet und zur Zeit noch kinderlos. Er hat sein Physikstudium abgeschlossen und arbeitet seit dieser Zeit als wissenschaftlicher Mitarbeiter im Studiengang Luft- und Raumfahrttechnik an der Universität Stuttgart.

2.2.2 Vorlieben, Hobbys, Abneigungen

Frank ist ein gewissenhafter Forscher, der seit einem halben Jahr nach einem passenden Thema für seine Doktorarbeit sucht. Er überlegt, ob er erweiterte Tests zur Auswahl der zukünftigen Raumfahrer entwickeln sollte, welche neben den kognitiven und körperlichen Aspekten auch den Wahrheitswert von Antworten auf Fragen untersucht. Frank spielt in seiner Freizeit Bowling und geht gerne Wandern.

2.2.3 Nutzererwartung an das Produkt

Herr Bollwerker erwartet ein Produkt, was höchsten wissenschaftlichen Ansprüchen gerecht wird. Er erhofft sich mit der LIAR App signifikante und eindeutige Aussagen zu den Antworten von zukünftigen Raumfahrern zu erhalten.

2.2.4 Anwendungsszenario

Während der Fahrt mit den öffentlichen Verkehrsmitteln zu Universität findet Herr Bollwerker die LIAR-App im Google Play Store und beliest sich zum Thema Lügendetektor. Er beschließt die App zu kaufen und auf Verwendbarkeit für seine wissenschaftliche Arbeit zu testen.

2.3 Persona 3 - „Der Angeber“: Jonas Keppler (29)



2.3.1 Soziodemografische Daten

Jonas Keppler ist 29 Jahre alt und ledig. Eine feste Freundin hat nicht, da diese wie die Hemden wechselt. Er nach seinem Germanistikstudium eine Journalistenschule in Bonn besucht. Nebenbei hat er in einem kleinen Softwareunternehmen als Webentwickler gearbeitet um seinen monatlichen Ausgaben zu decken. Seit 6 Monaten arbeitet als Redakteur im Ressort "Digital" bei der Süddeutschen Zeitung in München.

2.3.2 Vorlieben, Hobbys, Abneigungen

Herr Keppler ist sehr interessiert an neuer Technik und coolen Apps, die er dann stolz während der Mittagspause all seinen Arbeitskollegen präsentiert. Er steht gern im Mittelpunkt. Jeden Donnerstag geht ins Kino und schaut sich die neuesten Filme in der Sneak Preview an. Er immer der Erste, der etwas Neues ausprobiert. Jonas Keppler macht Yoga und achtet sehr auf seine Ernährung. Er kauft im Biomarkt ein und wird im nächsten Sommer erstmals selbst Gemüse auf seinem Grundstück anbauen.

2.3.3 Nutzererwartung an das Produkt

Jonas erwartet ein Produkt, was sich als Publikumsmagnet für die nächste WG-Party eignet. Es muss andere Leute neugierig machen und sollte ihn als Entertainer dastehen lassen.

2.3.4 Anwendungsszenario

Jonas Keppler im Internet von der LIAR-App gelesen und war der Erste, der sich den Prototyp bestellt hat. Den nächsten Werktag kann er gar nicht mehr abwarten, denn

er weiß, dass ihm die Aufmerksamkeit der Kollegen damit gewiss ist.

3 Anforderungen

- EEG und Hautleitwert des Benutzers können ausgelesen werden.
- Messwerte des Benutzers können ausgewertet und angezeigt werden.
- Es kann ein Lügendetektortest mit vorgegebenen Fragen absolviert werden.
- Es kann ein Lügendetektortest mit Fragen, die von einer zweiten Person gestellt werden, absolviert werden.
- Es kann ein Benutzerprofil erstellt werden.
- Das Speichern der Messwertauswertung je Benutzer ist möglich.
- Es lässt sich eine neue Spielsession (Spielsession, Spieldauer, Spieleranzahl) erstellen.
- Eine Spielsession kann durchgeführt werden.
- Spielergebnisse werden angezeigt und können gespeichert werden.
- Spielergebnisse können an soziale Netzwerke verteilt werden.

4 priorisierte User Stories

4.1 User Stories mit hoher Priorität:

- Der Anwender kann das Spiel über eine Smartphone-App öffnen.
- Der Anwender kann seinen Messwert sehen und speichern.

4.2 User Stories mit mittlerer Priorität:

- Ein Gast muss sich registrieren können
- Der Anwender kann Text eingeben, um einen Fragenkatalog zu erstellen.
- Anwender kann für das Spiel die Anzahl der Mitspieler und Fragen einstellen.
- Spieler erkennen Lüge oder Wahrheit nachdem eine Frage beantwortet wurde.

4.3 User Stories mit niedriger Priorität:

- Der Anwender soll die Anzahl der Mitspieler auswählen können.
- Jeder Mitspieler kann seinen Punktestand einsehen.
- Über ein Leaderboard ist es möglich sich mit anderen Spielern zu messen.
- Anwender können ihren Punktestand via Facebook teilen.
- Anwender kann Fragen beantworten.
- Gespeicherte Fragerunden können nochmal gespielt werden.

Tabelle 1: Risikoanalyse

6 Systemüberblick und Systemarchitektur

Im Folgenden soll ein Überblick der verwendeten Systemkomponenten gegeben werden. In Abbildung 2 ist der generelle Ablauf der Kommunikation zwischen den Komponenten dargestellt. Die emotionale Erregung des Nutzers soll über zwei Sensoren gemessen werden. Zum einen erfolgt eine Messung des elektrischen Widerstandes der Haut über einen Galvanic Skin Sensor. Der Galvanic Skin Sensor kommuniziert über eine Bluetooth-Verbindung mit einem Arduino Shield. Das Arduino Shield wiederum ist via Bluetooth mit dem Smartphone des Nutzers verbunden. Der zweite Sensor ermöglicht die Registrierung der Hirnströme und wird als Elektroenzephalografie, kurz EEG, bezeichnet. Der EEG-Sensor kommuniziert ebenfalls über das Bluetooth-Protokoll mit dem Smartphone. Im Smartphone werden die vom Nutzer gewonnenen Daten ausgewertet und verständlich dargestellt. Messergebnisse können lokal in einer Datenbank abgelegt werden oder auch mit anderen Freunden auf einer Social Media Plattform geteilt werden.

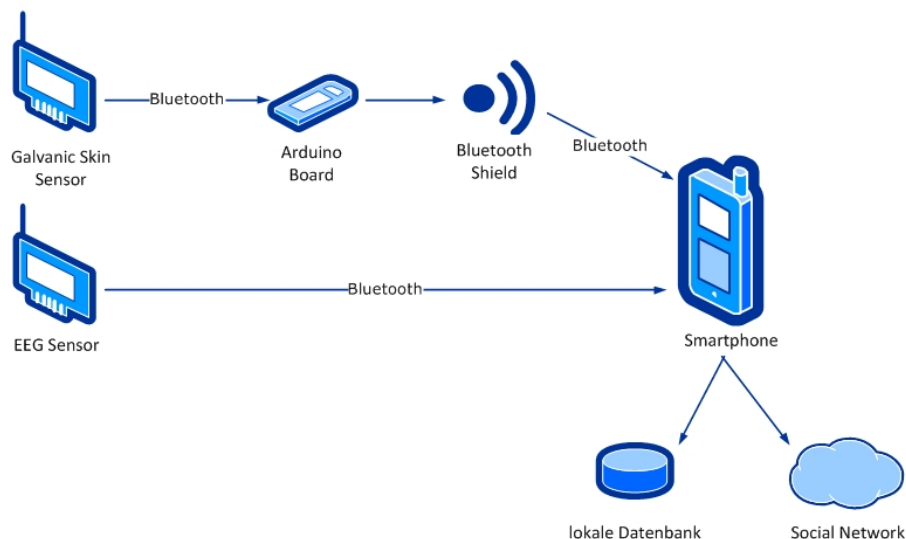


Abbildung 3: Systemarchitektur

7 Entwurf / Mockup des User-Interface

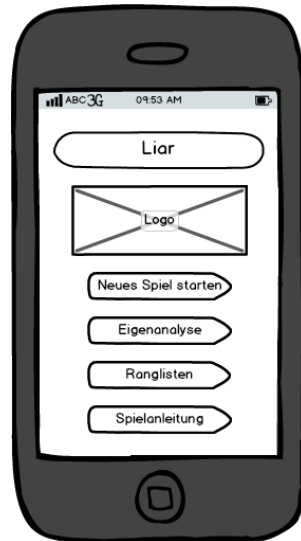


Abbildung 4: Der Hauptbildschirms der LIAR-App umfasst die Menüpunkte: neues Spiel starten, Eigenanalyse, Ranglisten sowie Spielanleitung.



Abbildung 5: Der Bildschirm für ein neues Spiel erlaubt das Festlegen der Spielernamen, der Spieleranzahl und der zu spielenden Runden.



Abbildung 6: Der Spielbildschirm zeigt eine Frage, deren Antwortmöglichkeiten, die verstrichene Zeit, sowie den grafischen Verlauf der Sensordaten.



Abbildung 7: Auswertungsbildschirm bei wahrheitsgemäßer Antwort



Abbildung 8: Auswertungsbildschirm bei nicht wahrheitsgemäßer Antwort



Abbildung 9: finaler Auswertungsbildschirm eines Spiels

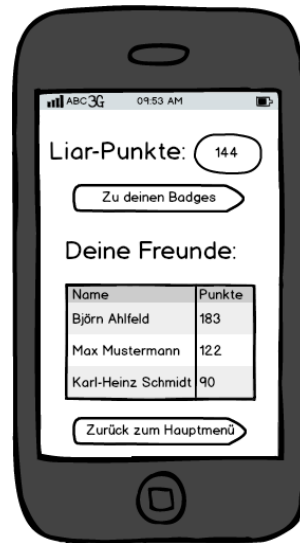


Abbildung 10: Der Ranking-Bildschirm zeigt die highscore-Einträge der Spieler

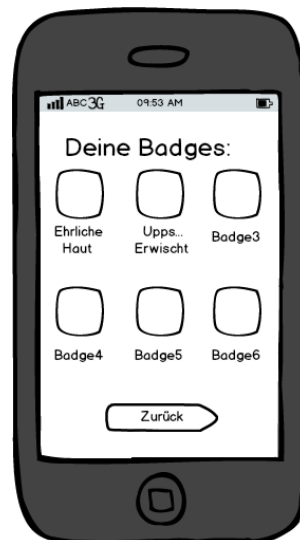


Abbildung 11: Der Bildschirm zeigt das Badge-System, bei dem Spieler für bestimmte Leistungen Auszeichnungen erhalten.



Abbildung 12: Der Eigenanalyse-Bildschirm veranschaulicht die Messwerte aller Sensoren, erlaubt das Abspeichern von Daten, den Zugriff auf alte Daten und bietet Hilfetexte zu den verwendeten Sensoren an.

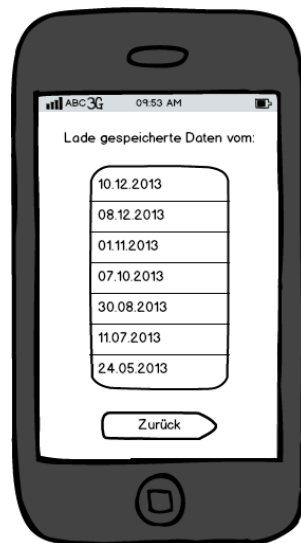


Abbildung 13: Der Bildschirm zum Laden gespeicherter Daten erlaubt den Zugriff auf vergangene Eigenanalysen.



Abbildung 14: Der Informationsbildschirm Hautleitwert erklärt verständlich den genannten Sensor.



Abbildung 15: Der Informationsbildschirm EEG erklärt verständlich den genannten Sensor.

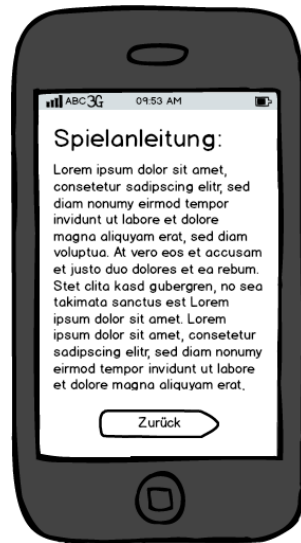


Abbildung 16: Der Anleitungsbildschirm erklärt die wichtigsten Sachverhalte zu App und Sensoren.

8 Implementierungen

Das folgende Kapitel der *Implementierungen* gibt einen Überblick über die verwendeten Technologien für das Produkt. Es werden genutzte (offene) Schnittstellen genannt, vorgestellt und deren Funktionsweise beschrieben. Zusätzlich werden selbst entwickelte Komponenten beschrieben und deren Implementierungen (sofern nötig) erörtert.

8.1 Arduino

Die Arduinoplattform ermöglicht eine *stackable*¹ Aneinanderkopplung verschiedener technischer Module. Unter anderem wurden in diesem Projekt ein e-Health-Modul, zum Auslesen von Galvanic Skin Daten, und zum Anderen die Anbindung an ein Wireless + Bluetooth Shield verwandt und eingesetzt. Die einzelnen Komponenten werden wie folgt beschrieben und durch die Beschreibung des Endprodukts vervollständigt.

8.1.1 eHealth - Plattform

Das e-Health-Modul ermöglicht es Android-Nutzer auf biometrische Daten zuzugreifen. Zu diesen Daten zählen unter anderem Puls-Messung, Blutdruck, EKG, Körpertemperatur und auch die Hautleitfähigkeit (GSR $\hat{=}$ galvanic skin response).

Das e-Health-Modul ist *stackable* und lässt sich daher einfach auf ein Arduino Uno „aufsetzen“. Hier ist darauf zu achten, dass es nicht zu verbogenen oder abgebrochenen Metall- / Verbindungsstücken kommt.

Zur Programmierung eines Arduino-Programms ist neben der Installation der entsprechenden Arduino IDE und der Treiberinstallation der Arduino Hardware, die

¹ aufeinanderstecken verschiedener Module - vergleichbar mit Lego-Prinzip

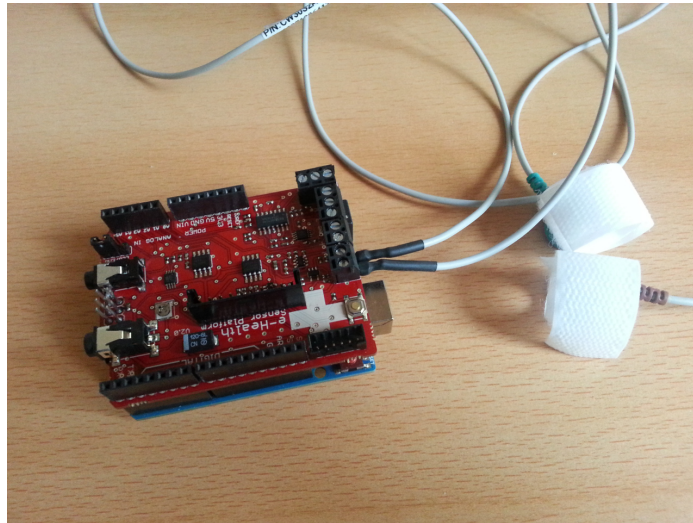


Abbildung 17: Arduino-eHealth-Stack

entsprechenden e-Health Libraries herunterzuladen und unter dem Standardinstallationsordner der Arduino-IDE abzulegen, z.B.:

- C:\Arduino\arduino-1.0.5\libraries\eHealth
- C:\Arduino\arduino-1.0.5\libraries\PinChangeInt

Danach kann in der Arduino IDE das gewünschte Programm erstellt werden. Dazu sollte zuerst das e-Health Package eingebunden werden, dann kann man auf den / die gewünschten Sensor / -en zugreifen:

1. `#include < eHealth.h >`
2. ...
3. `float resistance = eHealth.getSkinResistance();`

Die Variable *resistance* sollte in der loop-Funktion deklariert und definiert werden, um sicher zu stellen, dass die Werte immer „frisch“ in die Variable geschrieben werden.

Für weitere Information zur e-Health-Plattform im besonderen zur GSR verweisen wir auf die [e-Health-Website](#).

8.1.2 Wireless + Bluetooth Shield

Das Wireless + Bluetooth Shield es ebenfalls stackable und kann auf das Arduino Uno aufgesetzt werden. Im folgenden Fall stand neben dem Wireless Shield das Zusatzmodul BlueTooth Bee von iteadstudio zur Verfügung. Das Bluetooth-Modul hatte die Spezifikation V2.0 und Modellbezeichnung HC-06.



Abbildung 18: Das Bluetooth Modul

Die Modellbezeichnung ist ausschlaggebend dafür, ob ein solches Modul im Master², Master+Slave oder nur Slave³ Modus arbeitet. In diesem Fall bestand mit diesem Modul die einfache Slave-Funktionalität zur Verfügung.

Die Implementierung der Bluetooth-Verbindung läuft im Slave-Modus über die serielle Verbindung. Dazu wird im Setup ein `Serial.begin(< baud_rate >)` aufgerufen. Die Baudrate richtet sich nach der Übertragungsgeschwindigkeit mit der das Bluetooth-Modul arbeiten soll und mit der die entsprechende Gegenstation (Master) arbeitet.

In der `loop()`-Funktion ist dann der entsprechende zu übertragende Wert mit `Serial.print(< value >)` über die serielle Schnittstelle auszugeben. Zusätzlich ist nach

²kann aktives Pairing zu anderen Geräten übernehmen (Serverfunktionalität)

³kann kein Pairing übernehmen

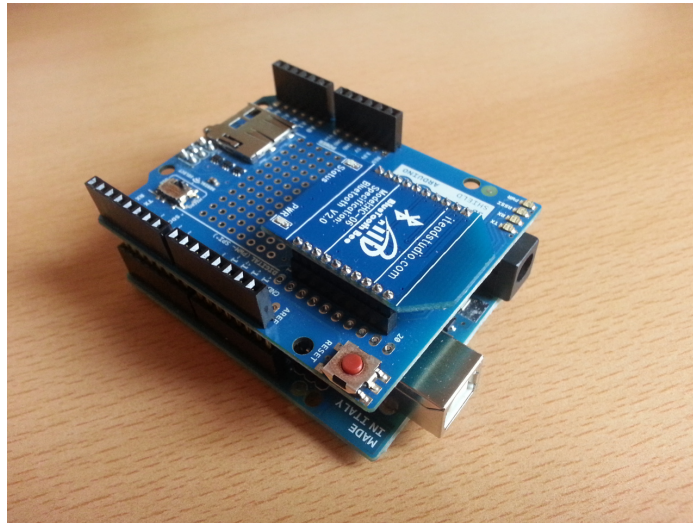


Abbildung 19: Der Arduino-Bluetooth-Stack

jedem Wert die Zeichnkette „\r\n“ mit *Serial.print()*; zu übergeben. Sie signalisiert den Abschluss eines Datensatzes.

8.1.3 Endprodukt

Beim Zusammenfügen der einzelnen Komponenten musste die Reihenfolge:

- Oben: Wireless + Bluetooth - Shield
- Mitte: e-Health-Modul
- Unten: Arduino Uno

eingehalten werden, da sonst das e-Health-Modul nicht mit Strom versorgt wird, wenn Wireless und e-Health miteinander getauscht werden.



Abbildung 20: Der vollständige Stack

8.2 Android

Bei der Android-Entwicklung wurde auf eine Unterstützung mit git entwickelt. So konnte der Quellcode sowohl für Android- als auch Arduino- und LaTeX-Quellcode versioniert verwaltet werden.

8.2.1 EEG: NeuroSky Brainwave Headset

Um das NeuroSky EEG Headset in Android anbinden zu können, bedarf es einiger Vorbereitungen:

1. Kauf und Download des Developer Tool für Android vom [NeuroSky Store](#)
2. Erzeugen eines lib-Ordners im Android-Projekt, sofern nicht schon vorhanden
3. Ablegen der ThinkGear.jar-Datei im lib-Ordner
4. Importanweisung: „import com.neurosky.thinkgear.*;“ in der entsprechenden Activity

5. AndroidManifest.xml: es ist die Bluetooth.Permission zu setzen

Des Weiteren müssen ein Android-BluetoothAdapter und ein TGDevice instanziiert werden.

Der bluetoothAdapter wird mittels *BluetoothAdapter.getDefaultAdapter()* zugewiesen. Wenn dieses erfolgreich war, dann wird das tgDevice erzeugt: *tgDevice = newTGDevice(blueetoothAdapter,*
Dem tgDevice wird der Default-BluetoothAdapter und eine *handler* zugewiesen.

Der handler wird einer Handler-Class erzeugt. Dieser *handler* hat die Aufgabe, Daten, die das Gerät sendet abzufangen und in einer gewünschten Form zu verarbeiten. Exemplarisch für die Aufmerksamkeitswerte des EEG:

1. case TGDevice.MSG_ATTENTION:
2. eeg.att.setText(" Attention: " + msg.arg1 + "\n" + eeg.att.getText());
3. break;

Folgende Daten können vom EEG ausgelesen werden:

- Verbindungsstatus: STATE.CONNECTING, STATE.CONNECTED, STATE.NOT_FOUND, STATE.NOT_PAIRING, STATE.DISCONNECTED
- Auslesedaten: MSG_ATTENTION, MSG_Meditation, MSG_BLINK, MSG_HEART_RATE
- sonstige Status: MSG_LOW_BATTERY, MSG_POOR_SIGNAL

Entscheidend für den Lügendetektor sind die Werte aus MSG_ATTENTION, MSG_MEDITATION und MSG_BLINK. Anhand der Aufmerksamkeits- (attention) und Ruhewerte (meditation) kann man die Aufregung bzw. Entspannung bei der Testperson ablesen. Hinzu kommen die Augenblinzler (blink), die stark oder schwach ausgeprägt sein bzw. gezählt werden können und daher auf zusätzliches „unkontrolliertes“ / „nervöses“ Verhalten hinweisen.

Das Anbinden des EEG an die Android-Applikation erfolgt über Bluetooth. Vor dem ersten Starten der Anwendung sollte das Android-Gerät mit dem EEG gepairt werden. Dann kann die eigene Anwendung gestartet werden. Sollte keine Verbindung in der Anwendung angezeigt werden, so sollte zusätzlich auf der EEG-Rückseite der



Abbildung 21: Das NeuroSky Brainwave Headset

„Pairing“-Knopf gedrückt werden.

8.2.2 Arduino Bluetooth

Zur Anbindung des Arduino Bluetooth Moduls kann man zweierlei vorgehen:

1. automatisierter (statischer) Verbindungsaufbau zwischen Android und Arduino
2. dynamischer Verbindungsaufbau durch scannen vorhandener Bluetooth-Geräte

Wir haben uns im Projekt dazu entschieden, dass wir die statische Methode wählen, da über das Scannen vorhandener Geräte der Verbindungsaufbau zu Fehlverbindungen geführt hatte und ggf. die Applikation geschlossen bzw. die built-in Bluetooth Funktionalität neugestartet werden musste.

Um die automatisierte Verbindung zu realisieren mussten einige Vorbedingungen erfüllt werden:

- MAC-Adresse des Bluetooth-Moduls scannen
- MAC-Adresse in der Applikation hinterlegen
- SPP⁴ UUID⁵ für Verbindung festlegen⁶

Zur Identifizierung des Arduino Bluetooth Moduls „linvor“ konnten wir mit der App *PowerBluetoothScanner* die MAC-Adresse „00:12:07:17:18:24“ auslesen. Beim Aufbau der Bluetooth Verbindung soll eine UUID hinterlegt werden. Android (Google Inc.) selbst schlägt vor: „If you are connecting to a Bluetooth serial board then try using the well-known SPP UUID 00001101-0000-1000-8000-00805F9B34FB.“⁷. Auch für die Arduino Bluetooth zu Android Verbindung muss ein Handler erzeugt werden (vgl. EEG-handler auf Seite 30). In diesem Fall musste der eingehende Datenstrom in ein byte-Array umgewandelt werden: `byte[] readBuf = (byte[])msg.obj`. Ausgehend von

⁴serial port profile

⁵128-bit universally unique identifier

⁶wird für Bluetooth Socket benötigt

⁷<http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html>

dem *msg.obj* das über die seriellen Schnittstelle übertragen wird, können diese Signale in ein byte-Array gecastet und der Variable *readBuf* zugewiesen werden. Aus dem byte-Array kann dann mittels *newString()* ein String erzeugt und weiter verarbeitet werden.

Die eigentliche Bluetooth-Verbindung wird in der *onResume()* Methode hergestellt. Hierzu wird einer Variable *btAdapter* (von *BluetoothAdapter*, siehe EEG) durch die Methode *createBluetoothSocket(device)* gesetzt. Der Parameter *device* ist eben die Verbindung (über die MAC-Adresse) zum Arduino Bluetooth Shield

```
BluetoothDevice device = btAdapter.getRemoteDevice(ADDRESS);
```

Mit der Erstellung des Bluetooth-Socket kann nun ein Verbindungs-Thread (*ConnectedThread*) erzeugt werden. Dieser sorgt dafür, dass die eingehenden Daten ordnungsgemäß (ge'threaded') empfangen (wenn gewünscht auch gesendet) werden können.

Mit *mConnectedThread = newConnectedThread(btSocket);* und *mConnectedThread.start();* wird der Thread gestartet.

Bluetooth-Status überprüfen

Da eine gültige Verbindung nur mit angeschaltetem Bluetooth-Modul des mobilen Geräts funktioniert, muss von der Anwendung überprüft werden, ob Bluetooth angeschaltet ist. Dazu wird die Methode *checkBTState()* genutzt. Sie wird in der Activity-Methode *onCreate()* gestartet und überprüft folgende Status:

1. wurde ein Bluetooth-Adapter angelegt, wenn nicht, dann beende und gib Fehlermeldung aus
2. Bluetooth-Adapter vorhanden und „enabled“, dann ist alles ok
3. Bluetooth-Adapter vorhanden aber nicht „enabled“, dann erzeuge und rufe einen Intent mit *BluetoothAdapter.ACTION_REQUEST_ENABLE* auf und star-

te die Methode *startActivityResult()* um das Anschalten des Bluetooth-Moduls zu erzwingen.

8.2.3 Spielaufbau

8.2.4 Datenpersistenz

8.2.5 Datenvisualisierung