

# **LIAR - Lügendetektor**



**Semesterprojekt  
Mobile Applications for Public Health  
03. Februar 2014**

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

# Inhaltsverzeichnis

<b>1 Projektplanung</b>	<b>4</b>
1.1 Meilensteine . . . . .	5
1.1.1 Einarbeitung . . . . .	5
1.1.2 Aufbau eines User Interfaces . . . . .	5
1.1.3 Anbindung des EEG Sensors . . . . .	6
1.1.4 Anbindung des Galvanic Skin Sensors . . . . .	6
1.2 fiktive Zeitungsmeldung . . . . .	7
1.3 Produktverpackung . . . . .	8
<b>2 Personas und Anwendungsszenarien</b>	<b>9</b>
2.1 Persona 1 - „Der Kontrollfreak“: Elisa Schubert (20) . . . . .	9
2.1.1 Soziodemografische Daten . . . . .	9
2.1.2 Vorlieben, Hobbys, Abneigungen . . . . .	10
2.1.3 Nutzererwartung an das Produkt . . . . .	10
2.1.4 Anwendungsszenario . . . . .	10
2.2 Persona 2 - „Der Wissenschaftler“: Frank Bollwerker (36) . . . . .	11
2.2.1 Soziodemografische Daten . . . . .	11
2.2.2 Vorlieben, Hobbys, Abneigungen . . . . .	11
2.2.3 Nutzererwartung an das Produkt . . . . .	12
2.2.4 Anwendungsszenario . . . . .	12
2.3 Persona 3 - „Der Angeber“: Jonas Keppler (29) . . . . .	12
2.3.1 Soziodemografische Daten . . . . .	12
2.3.2 Vorlieben, Hobbys, Abneigungen . . . . .	13
2.3.3 Nutzererwartung an das Produkt . . . . .	13
2.3.4 Anwendungsszenario . . . . .	13
<b>3 Anforderungen</b>	<b>14</b>
<b>4 priorisierte User Stories</b>	<b>15</b>
4.1 User Stories mit hoher Priorität: . . . . .	15
4.2 User Stories mit mittlerer Priorität: . . . . .	15
4.3 User Stories mit niedriger Priorität: . . . . .	15
<b>5 Risikobetrachtung</b>	<b>16</b>
<b>6 Systemüberblick und Systemarchitektur</b>	<b>17</b>
<b>7 Entwurf / Mockup des User-Interface</b>	<b>18</b>

---

## ABBILDUNGSVERZEICHNIS

---

<b>8 Implementierungen</b>	<b>25</b>
8.1 Arduino . . . . .	25
8.1.1 eHealth - Plattform . . . . .	25
8.1.2 Wireless + Bluetooth Shield . . . . .	26
8.1.3 Endprodukt . . . . .	28
8.2 Android . . . . .	29
8.2.1 EEG: NeuroSky Brainwave Headset . . . . .	29
8.2.2 Arduino Bluetooth . . . . .	31
8.2.3 Spielaufbau . . . . .	34
8.2.4 Datenpersistenz . . . . .	38
8.2.5 Datenvisualisierung . . . . .	38
<b>9 Verifizierung, Evaluation, Tests</b>	<b>39</b>
9.1 Testszenario 1: Singleplayer . . . . .	39
9.2 Testszenario 2: Multiplayer . . . . .	39
9.3 Auswertung der Testszenarios . . . . .	39
<b>10 Geschäftsmodell</b>	<b>39</b>
10.1 Business Model Canvas - Persona-Typ: Wissenschaftler . . . . .	42
10.2 Business Model Canvas - Persona-Typ: Angeber . . . . .	43
10.3 Business Model Canvas - Persona-Typ: Kontrollfreak . . . . .	44
10.4 Fazit zum Geschäftsmodell . . . . .	45

## Abbildungsverzeichnis

1 Zeitungsmeldung . . . . .	7
2 Produktverpackung . . . . .	8
3 Systemarchitektur . . . . .	17
4 Mockup Hauptbildschirm . . . . .	18
5 Mockup Bildschirm für neues Spiel . . . . .	18
6 Mockup Spielbildschirm . . . . .	19
7 Mockup Bildschirm Fragenauswertung Wahrheit . . . . .	19
8 Mockup Bildschirm Fragenauswertung Lüge . . . . .	20
9 Mockup finaler Bildschirm zur Fragenauswertung . . . . .	20
10 Mockup highscore-Bildschirm . . . . .	21
11 Mockup Badge-Bildschirm . . . . .	21
12 Mockup Bildschirm Eigenanalyse . . . . .	22
13 Mockup Bildschirm zum Laden gespeicherter Daten . . . . .	22

## TABELLENVERZEICHNIS

---

14	Mockup Infobildschirm Hautleitwert . . . . .	23
15	Mockup Infobildschirm EEG . . . . .	23
16	Mockup Anleitungsbildschirm . . . . .	24
17	Arduino-eHealth-Stack . . . . .	26
18	Das Bluetooth Modul . . . . .	27
19	Der Ardunio-Bluetooth-Stack . . . . .	27
20	Der vollständige Stack . . . . .	28
21	Das NeuroSky Brainwave Headset . . . . .	30
22	The Honest Skin . . . . .	36
23	Der Workflow . . . . .	37
24	Androidplotdiagramm: Attention . . . . .	38

## Tabellenverzeichnis

1	Risikoanalyse . . . . .	16
2	Testszenario 1: Singleplayer . . . . .	40
3	Testszenario 2: Multiplayer . . . . .	40
4	Testszenario: Allgemein . . . . .	41
5	Business Model Canvas - Persona-Typ: Wissenschaftler . . . . .	42
6	Business Model Canvas - Persona-Typ: Angeber . . . . .	43
7	Business Model Canvas - Persona-Typ: Kontrollfreak . . . . .	44
8	Business Model Canvas - Pro/Contra Auswertung . . . . .	45

## **1 PROJEKTPLANUNG**

---

### **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 einer Aussage zum Wahrheitsgehalt der Antwort des Nutzers.

## **1 PROJEKTPLANUNG**

---

### **1.1 Meilensteine**

Meilensteine repräsentieren Zwischenergebnisse in der Programmierung, 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 besonderen Leistung [z.B. ehrliche Haut oder Lügner des Monats]) und ermöglicht das Teilen des Ergebnisses auf Facebook

## **1 PROJEKTPLANUNG**

---

### **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 PROJEKTPLANUNG

### 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 Spielemarkt

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, Gehirnscans mit einem EEG-Messgerät und die Hautoberflächen Spannung mittels Galvanic Skin Sensor zu messen und mit bekannten Werten zu ver-

gleichen. Das zusätzliche Verwenden von Gehirnscans ist neuer, aber mittlerweile kein unbekanntes Verfahren mehr. „Bei der Ermittlung einer Lüge sind wesentlich mehr Gehirnteile aktiv im Vergleich zu einer wahren Antwort“ so Phillippe Wels, Entwickler bei XYZ. Auf die Frage, warum gerade ein Gehirnstromsensor 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 Selbstlerngruppen“. „Wir wollen einen Prototypen schaffen für neue Innovationen und Anwendungsbereiche, möchten aber das noch nicht weiter konkreti-

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 Game 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 Gerätesteuerung oder im Multimedia-Bereich“ meint Patrick Bork, ebenfalls Entwickler.

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



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

**Abbildung 1:** fiktive Zeitungsmeldung

## 1 PROJEKTPLANUNG

### 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 PERSONAS UND ANWENDUNGSSZENARIEN**

---

### **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 PERSONAS UND ANWENDUNGSSZENARIEN**

---

### **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 Jour-

## **2 PERSONAS UND ANWENDUNGSSZENARIEN**

---

nalistenschule 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**

Im Folgenden Abschnitt werden die Anforderungen für die LIAR Android App aufgelistet.

- Die App soll mit dem mobilen Betriebssystem Android entwickelt werden.
- Ein EEG- und eine Galvanic Skin Sensor (Hautleitwert) sollen Daten an das Android Smartphone senden können.
- Die Messwerte des EEG-Sensors und der Hautleitwert des Benutzers sollen auf dem Smartphone ausgelesen werden können.
- Die Messwerte des Benutzers können ausgewertet und angezeigt werden.
- Es kann ein Benutzerprofil erstellt werden.
- Das Speichern der Messwertauswertung je Benutzer ist möglich.
- 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 lässt sich eine neue Spielsession (Spielsession, Spieldauer, Spieleranzahl) erstellen.
- Eine Spielsession kann durchgeführt werden.
- Die Spielergebnisse werden angezeigt und können gespeichert werden.
- Die 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 Guest 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.

## 5 Risikobetrachtung

### 5 RISIKOBETRACHTUNG

---

Produkt: Lügendetektorspiel „LIAR“	Team: G4	Ursachen des Fehlers	Folgen des Fehlers	Ausgangs- zustand	Risikominderungs- maßnahmen	Verbesserter Zustand			
Produkt-Komponente	potentielle Fehler			A B	E RPZ	A B	E	RPZ	
	Wie sieht die Fehler aus?	Wodurch entsteht der Fehler?	Was kann passieren? Was können die Auswirkungen?		Welche Maßnahmen können getroffen werden?	0			
<b>Beispiel</b>									
<b>Sensor</b>	Anzeige falscher Messwerte	Erfassung fehlerhafter Messwerte durch falsche Sensor-Position	Überbeanspruchung durch nicht-tägiges Training	8	10	9	720	Tutorial Sensor-Positionierung, konstruktive Maßnahmen	3 10 3 90
<b>Beispiel 1</b>									
<b>Sensor</b>	Fehlinterpretation der Messdaten	Anwender ist durch äußere Umstände oder Ablenkung in Aufregung.	Wahrheit wird als Lüge detektiert, da Messgerät nur Ausschlag der Datenströme misst unabhängig von deren Ursache.	5	6	150	und ablenkungsfreien Umfeldes / Befragungssituation.	Schaffen eines ruhigen und ablenkungsfreien Umfeldes / Befragungssituation.	2 2 1 4
<b>Beispiel 2</b>									
<b>Sensor</b>	Unzureichende Messgenauigkeit	Schlechte Kalibrierung des Sensors.	Die durch eine Lüge resultierende gesteigerte EEG-Aktivität oder Aufregung wird falsch durch den Sensor gemessen bzw. Abgebildet.	3	8	7	168	Kalibrierung der Messgeräte.	2 8 5 80
<b>Beispiel 3</b>									
<b>Software/Treiber</b>	Messdaten vom Sensor werden falsch berechnet.	Softwarefehler, Bugs, Treiberprobleme.	Fehlerhafte Interpretation: falsche Ergebnisse.	10	9	720	Wartung der Software. Finden und Beheben des Fehlers.	1 2 1 2	

**Tabelle 1:** Risikoanalyse

### 6 Systemüberblick und Systemarchitektur

Im Folgenden soll ein Überblick der verwendeten Systemkomponenten geben 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 zum einen lokal in einer Datenbank abgelegt werden oder auch mit anderen Freunden auf einer Social Media Plattform geteilt werden.

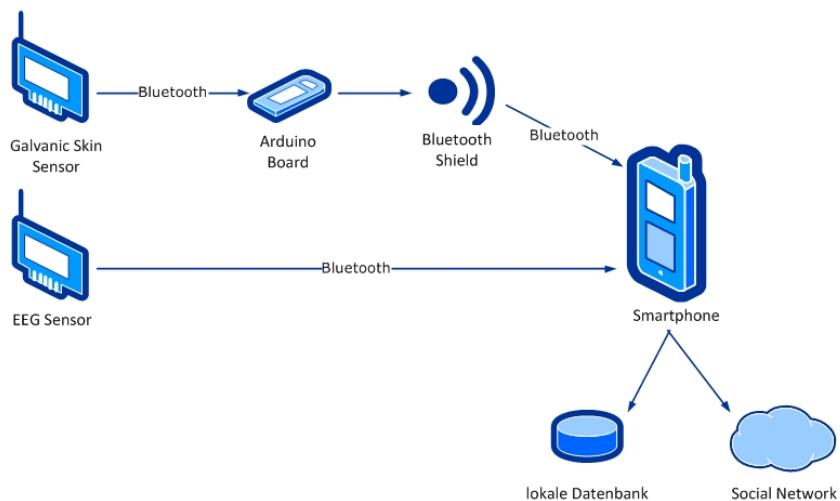
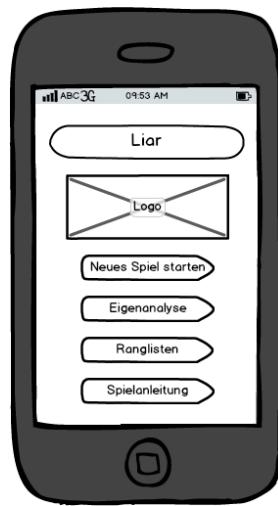


Abbildung 3: Systemarchitektur

## 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.

## 7 ENTWURF / MOCKUP DES USER-INTERFACE

---



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



**Abbildung 7:** Auswertungsbildschirm bei wahrheitsgemäßer Antwort

## 7 ENTWURF / MOCKUP DES USER-INTERFACE

---



Abbildung 8: Auswertungsbildschirm bei nicht wahrheitsgemäßer Antwort



Abbildung 9: finaler Auswertungsbildschirm eines Spiels

## 7 ENTWURF / MOCKUP DES USER-INTERFACE

---



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



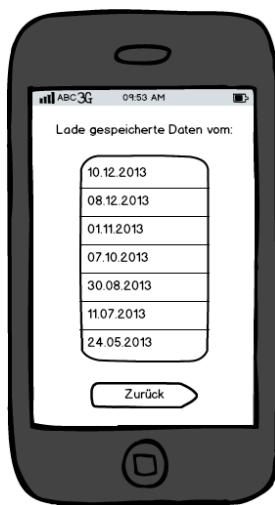
Abbildung 11: Der Bildschirm zeigt das Badge-System, bei dem Spieler für bestimmte Leistungen Auszeichnungen erhalten.

## 7 ENTWURF / MOCKUP DES USER-INTERFACE

---



**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.

## 7 ENTWURF / MOCKUP DES USER-INTERFACE

---



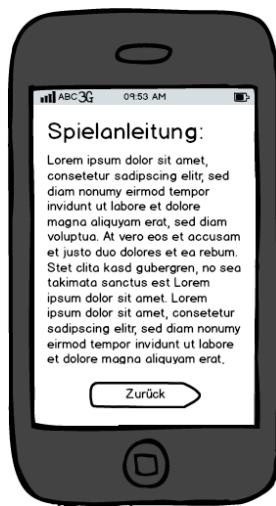
**Abbildung 14:** Der Informationsbildschirm Hautleitwert erklärt verständlich den genannten Sensor.



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

## 7 ENTWURF / MOCKUP DES USER-INTERFACE

---



**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*<sup>1</sup> 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, Kör-pertemperatur 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 entsprechenden e-Health Libraries herunterzuladen und unter dem Standardinstallationsordner der Arduino-IDE abzulegen, z.B.:

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

---

<sup>1</sup>aufeinanderstecken verschiedener Module - vergleichbar mit Lego-Prinzip

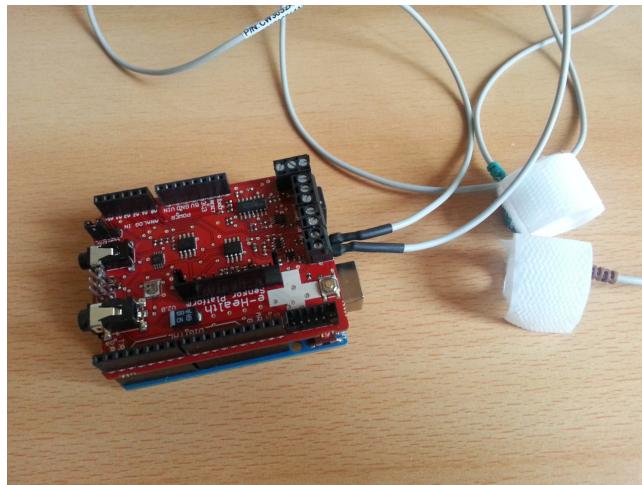


Abbildung 17: Arduino-eHealth-Stack

- 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 Modelbezeichnung HC-06.

## 8 IMPLEMENTIERUNGEN

---



Abbildung 18: Das Bluetooth Modul

Die Modelbezeichnung ist ausschlaggebend dafür, ob ein solches Modul im Master<sup>2</sup>, Master+Slave oder nur Slave<sup>3</sup> Modus arbeitet. In diesem Fall bestand mit diesem Modul die einfache Slave-Funktionalität zur Verfügung.



Abbildung 19: Der Arduinio-Bluetooth-Stack

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.

<sup>2</sup>kann aktives Pairing zu anderen Geräten übernehmen (Serverfunktionalität)

<sup>3</sup>kann kein Pairing übernehmen

## 8 IMPLEMENTIERUNGEN

---

In der `loop()`-Funktion ist dann der entsprechende zu übertragende Wert mit `Serial.print(< value >)` über die serielle Schnittstelle auszugeben. Zusätzlich ist nach jedem Wert die Zeichenkette „\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.neuroskey.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(blueoothAdapter, handler);`. Dem `tgDevice` wird der Default-`BluetoothAdapter` und eine `handler` zugewiesen.

Der `handler` wird mit 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;`

## 8 IMPLEMENTIERUNGEN

---

Folgende Daten können vom EEG ausgelesen werden:

- Verbindungsstatus: STATE\_CONNECTING, STATE\_CONNECTED, STATE\_NOT\_FOUND, STATE\_NOTE\_PAIED, 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.



**Abbildung 21:** Das NeuroSky Brainwave Headset

Das Anbinden des EEG an die Android-Applikation erfolgt über Bluetooth. Vor dem ersten Starten der Anwendung sollte das Android-Gerät mit dem EEG gepaart werden. Dann kann die eigene Anwendung gestartet werden. Sollte keine Verbindung in der Anwendung angezeigt werden, so sollte zusätzlich auf der EEG-Rückseite der „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
- SPP<sup>4</sup> UUID<sup>5</sup> für Verbindung festlegen<sup>6</sup>

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.“<sup>7</sup>. Auch für die Arduino Bluetooth zu Android Verbindung muss ein Handler erzeugt werden (vgl. EEG-Handler auf Seite 29). In diesem Fall musste der eingehende Datenstrom in ein byte-Array umgewandelt werden: `byte[] readBuf = (byte[])msg.obj;`. Ausgehend von 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

---

<sup>4</sup>serial port profile

<sup>5</sup>128-bit universally unique identifier

<sup>6</sup>wird für Bluetooth Socket benötigt

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

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 starte die Methode `startActivityForResult()` um das Anschalten des Bluetooth-Mo-duls zu erzwingen.

### Änderungen zu Android 4.3

Während der Entwicklung wurde von Goolge Inc. die neue Version des Android-OS freigegeben: Version 4.3. Damit verbunden sind auch einige Änderungen

## 8 IMPLEMENTIERUNGEN

---

bzgl. des Verbindungs aufbaus zwischen den Devices und dem Smartphone (in diesem Fall Samsung Galaxy S3) über die Bluetooth-Schnittstelle. Die Änderungen betrafen aber nur den Verbindungs aufbau mit dem EEG-Headset. Dazu muss te die Methode `setupEegBluetooth()` geändert werden.

Der durch Android und der App erzeugte EEG-Bluetooth-Adapter musste auf sich selbst eine `startDiscovery()`-Methode aufrufen, die es dann dem Adapter ermöglicht, zwölf Sekunden nach verfügbaren Geräten in Einzugsbereich zu suchen. Sollte darunter ein TGDevice sein, so wird die Verbindung mittels `tgDevice.connect(true);` hergestellt und der Datenabruf mit `tgDevice.start();` gestartet. Alles Weitere übernimmt der für dieses Device entwickelte `HandleeegHandler`.

Zur Verwunderung musste an der Verbindungseinstellung für das Arduino-Bluetooth-Shield keine Anpassung bzgl. des Versionswechsels durchgeführt werden.

### 8.2.3 Spielaufbau

#### Die Oberfläche (GUI)

Die Anwendung startet mit der Möglichkeit der Auswahl einer von vier Punkten:

1. Neues Spiel starten
2. Eigenanalyse
3. Ranking vorhergehender Spiele
4. Kleine Hilfe / Instruktionen

Am Entscheidendsten sind die Punkte „Neues Spiel starten“ und „Eigenanalyse“.

#### „Neues Spiel starten“

Ein neues Spiel kann über diese Activity erstellt werden. Zuerst müssen alle teilnehmenden Spieler mit Namen angelegt werden, dann muss die Anzahl der Runden (die Anzahl der Fragen) festgelegt werden.

Jetzt sind alle Vorbereitungen getroffen und es kann über „Spiel starten“ losgehen.

#### Die Spiellogik

Die Logik des Spiels setzt sich aus den Parameter:

1. Attention: Aufmerksamkeit
2. Meditation: Ruhewert
3. Blinks: Augenblinzeln

### 4. Galvanic: Hautwiderstand

zusammen. Für jede Frage wird für jeden o.g. Parameter ein int-Array erzeugt. Dieses Array wird pro Sekunde mit den aktuellen Werten des Sensors gefüllt. Für die Aufmerksamkeits-, Ruhewert- und Hautwiderstandswerte wird im Anschluss die Standardabweichung gebildet. Diese entscheidet darüber, ob es Änderungen im Bereich der gemessenen Sensordaten gibt oder nicht. Vor jeder neuen Fragerunde wird für die Spieler eine Kalibrierung durchgeführt. Diese Kalibrierung ermittelt jeweils einen Referenzwert für die Aufmerksamkeits-, Ruhewert- und Hautwiderstandswerte. Nach jeder Frage kann dann überprüft werden, ob die neu berechneten Standardabweichungen für Aufmerksamkeits-, Ruhewert- und Hautwiderstandswerte mit den Referenzwerten aus der Kalibrierung übereinstimmen oder in einem Bereich von - bis liegen. Sollten die neuen Werte über den Grenzwerten liegen, so wird eine Lüge (erhöhte Anspannung) vermutet.

Zusätzlich werden für jede Runde die Anzahl der Augenblinzler gezählt. Auch diese werden während der Kalibrierung gezählt und im Anschluss bei jeder Frage verglichen. Gerade wenn mehr Augenblinzler in einer Fragerunde gezählt werden wie im Vergleich zur Kalibrierung, so kann man von einer Lüge sprechen.

Alle vier Parameter werden dann in einer Methode miteinander verglichen. Die Methode liefert dann entsprechend einen boolschen Wert (true/false) zurück, ob eine Lüge vorliegt oder nicht.

```

private boolean theHonestSkin(){
    boolean liar = false;

    boolean att_lie = false;
    if(std_res_att <= (after_calib_att += after_calib_att*LIE_FACTOR_0_1)){
        att_lie = false;
    }else{
        att_lie = true;
    }

    boolean med_lie = false;
    if(std_res_med >= (after_calib_med -= after_calib_med*LIE_FACTOR_0_1)){
        med_lie = false;
    }else{
        med_lie = true;
    }

    boolean res_lie = false;
    if(std_res_resis <= (after_calib_resis += after_calib_resis*LIE_FACTOR_0_1)){
        res_lie = false;
    }else{
        res_lie = true;
    }

    boolean blinks_lie = false;
    if(blinkCounter > after_calib_blinks){
        blinks_lie = false;
    }else{
        blinks_lie = true;
    }

    if((att_lie && med_lie && res_lie) || (att_lie && med_lie && blinks_lie) ||
       (att_lie && res_lie && blinks_lie) || (med_lie && res_lie && blinks_lie)){
        liar = true;
    }
    eegAttentionFinishedFlag = false;
    eegMeditationFinishedFlag = false;
    galvanicFinishedFlag = false;

    answerQuestion(liar);
    return liar;
}
}

```

Abbildung 22: The Honest Skin

### „Die Eigenanalyse“

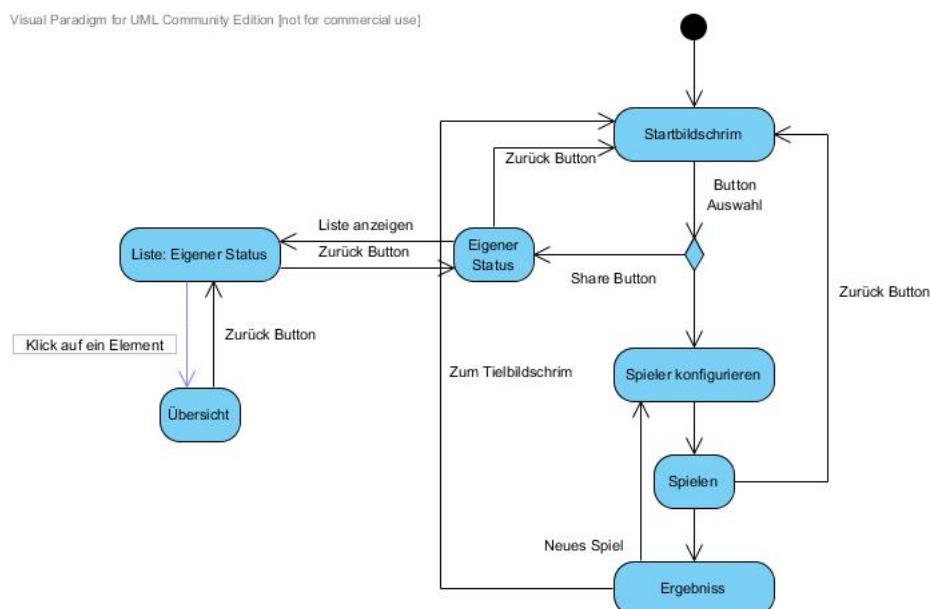
Die Eigenanalyse gibt dem Nutzer die Möglichkeit, dass er sich seine aktuellen Daten visualisiert anschauen kann. Dazu müssen alle Sensoren im Nutzer angebracht worden sein und die entsprechende Activity laufen.

Nach erfolgreichem Verbindungsaufbau über Bluetooth zwischen EEG-Headset und Arduino-Bluetooth-Shield, werden die übermittelten Daten ad-hoc auf dem Bildschirm dargestellt.

Die Verarbeitung der Datenvizualisierung ist auf Seite 38 nachzulesen. Eine Speicherung der Daten ist und wurde hier vorerst nicht vorgesehen.

### Der Workflow

Im Folgenden ist eine grobe Darstellung des Workflows und der einzelnen Activity-Zustände abgebildet:



**Abbildung 23:** Der Workflow

### 8.2.4 Datenpersistenz

### 8.2.5 Datenvisualisierung

Die Datenvisualisierung wird direkt durch die Handler vom Galvani Skin und EEG-Headset gesteuert. Es wurde dazu eine weitere Bibliothek hinzugefügt *libs/androidplot – core – 0.6.0.jar*. Auf der Grundlage dieser Androidplot-Bibliothek konnten auf der Eigenanalyse-Activity drei Diagramme hinzugefügt werden:

*eegAttentionPlot*, *eegMeditationPlot* und *galvanicPlot*.

Jeder Plot dient dazu, dass er das Diagramm mit seinen Achsen beschriftet, die Farben und Skalierungen bereitstellt und sich ggf. aktualisiert, wenn neue Daten hinzugekommen sind. So konnten für die Attention- und Meditation-Werte die y-Achse mit `< Plot >.setRangeBoundaries(0, 100, BoundaryMode.FIXED);` auf eine Skala von 0 bis 100 gesetzt werden.

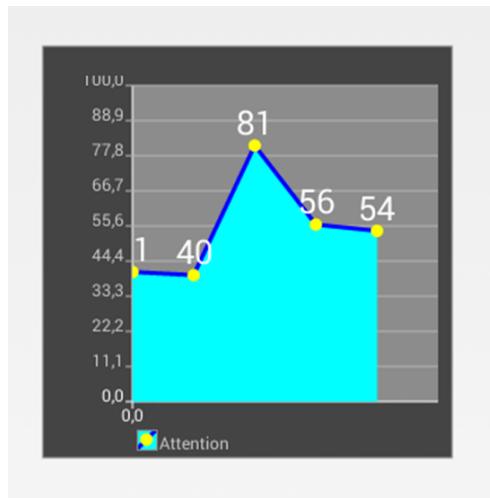


Abbildung 24: Androidplotdiagramm: Attention

Für jeden Plot und deren Sensor wurden daher eine Liste mit `newArrayList < Integer > ()`; erzeugt. Diese Liste dient dazu, neue Werte aufzunehmen und die letzten vier Werte anzuzeigen. Hierzu wurden nach jedem Update der Liste die oder das entsprechende Diagramm mit `redrawer.start()`; in der GUI aktualisiert.

## **9 Verifizierung, Evaluation, Tests**

### **9.1 Testszenario 1: Singleplayer**

Verbinde die Sensoren mit deinem Gerät und starte die App. Beginne mit der Eigenanalyse. Folge dazu den Anweisungen auf dem Bildschirm und lege die Sensoren korrekt am Körper an. Wenn du bereit bist, starte die Aufzeichnung deiner Werte. Lade nach der Sitzung einen gespeicherten Datensatz um ihn mit dem Aktuellen zu vergleichen. Beende die App nach erfolgreicher Analyse und entferne die Sensoren von Körper und Gerät.

### **9.2 Testszenario 2: Multiplayer**

Verbinde die Sensoren mit deinem Gerät und starte die App. Mache dich zunächst mit den Spielregeln vertraut. Dann befestige die Sensoren korrekt am Körper und starte ein neues Spiel mit 2 Mitspielern. Lege dazu 2 neue Profile an und stelle die Anzahl der Runden auf 3 ein. Beginne nun mit der Befragung. Bist du mit deinen Fragen am Ende tausche die Sensoren mit deinem Spielpartner und vollende die Sitzung. Schau dir danach noch einmal die Rangliste an um zu sehen wer mehr Punkte erreicht hat. Beende die App nach erfolgreicher Spielrunde und entferne die Sensoren von Körper und Gerät.

### **9.3 Auswertung der Testszenarios**

## **10 Geschäftsmodell**

Zur Beschreibung der Funktionsweise des zukünftigen Unternehmens und Evaluierung der optimalen Gewinnerwirtschaftung wurden drei Business Model Canvas erstellt. Dabei wurden die im Kapitel Personas und Anwendungsszenarien beschriebenen Zielgruppen betrachtet.

## 10 GESCHÄFTSMODELL

---

Frage	-	-	o	+	++
Konnten die Sensoren korrekt mit dem Gerät verbunden werden?					
Konnte die Eigenanalyse gestartet werden?					
Konnten die Sensoren korrekt am Körper angelegt werden?					
Konnten die Werte aufgezeichnet und abgespeichert werden?					
Konnte ein älterer Datensatz geöffnet werden?					
Konnten die Sensoren problemlos von Gerät und Körper entfernt werden?					

**Tabelle 2:** Testszenario 1: Singleplayer

Frage	-	-	o	+	++
Konnten die Sensoren korrekt mit dem Gerät verbunden werden?					
Konnten die Spielregeln gelesen werden?					
Konnten die Sensoren korrekt am Körper angelegt werden?					
Konnten 2 neue Profile angelegt werden?					
Konnte die Rundenzahl eingestellt werden?					
Konnten die Fragerunden absolviert werden?					
Konnte nach Abschluss des Spiels die Rangliste geladen werden?					
Konnten die Sensoren problemlos von Gerät und Körper entfernt werden?					

**Tabelle 3:** Testszenario 2: Multiplayer

## 10 GESCHÄFTSMODELL

---

Frage	-	-	o	+	++
Wie einfach war die Anbringung der Sensoren?					
Wie komfortabel war das Tragen der Sensoren während des Spiels?					
Wie intuitiv war die Bedienung der App?					
Wie gut war die Auswertung der Sensoren?					
Wie viel Spaß macht das Spiel?					

**Tabelle 4:** Testszenario: Allgemein

### 10.1 Business Model Canvas - Persona-Typ: Wissenschaftler

<b>Key Partners</b>	<b>Key Activities</b>	<b>Value Propositions</b>	<b>Customer Relationships</b>	<b>Customer Segments</b>
• Sensorikhersteller • Verbrauchsmaterialhersteller	• Softwareentwicklung • Support • Webseite zum Vertrieb des Produktes	• Verifizierungsmöglichkeit für Fragebögen	• Newsletter • Gewinnspiele	Wissenschaftler
			<b>Channels</b>	
			• Produktvorstellung auf Konferenzen, Messen, Tagungen • Produktdarstellung und Vertrieb auf eigener Webseite	
	<b>Key Resources</b>		• Produkt gelangt als Starterpaket per Post zum Kunden	
	• positiv validierte empirische Studie zum Beleg der Validität des Produktes			
	• Customer Support			
	• Programmierer • IT-Infrastruktur			
<b>Cost Structure</b>		<b>Revenue Streams</b>		
• Human Resources / Marketing • Vertrieb • Risikokosten / Eintritt Konferenzen • Logistik für Verbrauchsmaterial • Studie zum Beleg der Validität des Produktes		• Lizenzmodell (App, Sensoren, Support) • Verbrauchskosten (z.B. Sensorspads für das EEG und den Galvanic Skin Sensor)		

**Tabelle 5:** Business Model Canvas für die Persona vom Typ Wissenschaftler

## 10.2 Business Model Canvas - Persona-Typ: Angeber

Key Partners	Key Activities	Value Propositions	Customer Relationships	Customer Segments
• Sensorikhersteller • Verbrauchsmaterialhersteller	• Softwareentwicklung • Support • Webseite zum Vertrieb des Produktes	• Anerkennung bei Mitmenschen • Entertainingtool	• Newsletter • Gewinnspiele	Angeber
Key Resources			Channels	
			• Social Media (youtube, facebook) • Produktdarstellung und Vertrieb auf eigener Webseite	
Cost Structure			Revenue Streams	
• Human Resources / Marketing • Logistik für Verbrauchsmaterial			• App kostenlos im Google Play Store • Einmalpreis beim Kauf der Sensoren (ohne zeitliche Limitierung) • Verbrauchskosten (z.B. Sensorspads für das EEG und den Galvanic Skin Sensor)	

Tabelle 6: Business Model Canvas für die Persona vom Typ Angeber

### 10.3 Business Model Canvas - Persona-Typ: Kontrollfreak

Key Partners		Customer Segments	
<ul style="list-style-type: none"> <li>• Sensorikhersteller</li> <li>• Verbrauchsmaterialhersteller</li> </ul>		<ul style="list-style-type: none"> <li>• eigenem Verlangen nach nach Kontrolle nachgehen</li> </ul>	
Key Activities		Value Propositions	
<ul style="list-style-type: none"> <li>• Softwareentwicklung</li> <li>• Support</li> <li>• Webseite zum Vertrieb des Produktes</li> </ul>		<ul style="list-style-type: none"> <li>• Newsletter</li> <li>• Gewinnspiele</li> <li>• postalisches Mailing zum Geburtstag</li> </ul>	
Key Resources		Channels	
<ul style="list-style-type: none"> <li>• Customer Support</li> <li>• IT-Infrastruktur</li> <li>• Programmierer</li> </ul>		<ul style="list-style-type: none"> <li>• Social Media (facebook, youtube)</li> <li>• Produktdarstellung und Vertrieb auf eigener Webseite</li> <li>• Produkt gelangt als Starterpaket per Post zum Kunden</li> </ul>	
Cost Structure		Revenue Streams	
<ul style="list-style-type: none"> <li>• Human Resources / Marketing</li> <li>• Logistik für Verbrauchsmaterial</li> </ul>		<ul style="list-style-type: none"> <li>• Einmalpreis beim Kauf ohne zeitliche Limitierung</li> <li>• Verbrauchskosten (z.B. Sensorpads für das EEG und den Galvanic Skin Sensor)</li> </ul>	

Tabelle 7: Business Model Canvas für die Persona vom Typ Kontrollfreak

## 10.4 Fazit zum Geschäftsmodell

Als Fazit ergab sich, dass der Persona-Typ Wissenschaftler ein deutlich erhöhtes Ausmaß an Investitionen, wie beispielsweise Kosten für eine Studie zum Beleg der Validität des Produktes, benötigt. Da zudem Beschaffungsprozesse an der Universität einen langen Zeitraum andauern können, ergibt sich für das Unternehmen ein hoher finanzieller Aufwand und eine längere Zeit ohne Einnahmen. Dies ergibt ein hohes Risiko für unser Unternehmen und aus diesem Grund schließen wir dieses Geschäftsmodell aus. Der Persona-Typ Angeber kann nicht langfristig an unser Produkt gebunden werden und scheidet aus diesem Grund ebenfalls als Geschäftsmodell aus. Wir haben uns für eine Spezialisierung auf den Persona-Typ Kontrollfreak entschieden, da eine langfristige Bindung an unser Produkt und geringere Anfangskosten als beim Persona-Typ Wissenschaftler zu erwarten sind.

	pro	contra
Wissenschaftler	<ul style="list-style-type: none"> <li>• Wissenschaftler bezahlen überdurchschnittlich viel Geld für ihr equipment.</li> <li>• Lizenzmodell bietet langfristige Einnahmequelle</li> </ul>	<ul style="list-style-type: none"> <li>• Beschaffungsprozesse an Universitäten dauern lang (viele Personen an Entscheidungen beteiligt)</li> <li>• Kosten für eine positiv validierte empirische Studie zum Beleg der Validität des Produktes</li> <li>• Kosten für Präsentationen auf Messen, Tagungen und Konferenzen</li> </ul>
Angeber	<ul style="list-style-type: none"> <li>• sorgt für einen begrenzten Zeitraum selbst für die Bekanntheit des Produktes</li> </ul>	<ul style="list-style-type: none"> <li>• kann nicht langfristig gebunden werden</li> <li>• Anfangsaufwand (Bestellung, Kosten) können dafür sorgen, dass der Kunde das Produkt nicht akzeptiert und nicht kauft</li> </ul>
Kontrollfreak	<ul style="list-style-type: none"> <li>• bietet permanente Einnahmequelle aufgrund seiner Persönlichkeit</li> </ul>	<ul style="list-style-type: none"> <li>• Die Kosten des Produktes dürfen nicht so hoch wie für Wissenschaftler sein.</li> </ul>

Tabelle 8: Pro/Contra Auswertung der Business Model Canvas