

Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Konzept und Umsetzung eines Datenanalysetools für eine physiologische Uhr

Concept and implementation of a data analysis tool for a
physiological watch

Bachelorarbeit

im Studiengang Informatik

von

Nino Baumgart

Prüfer: Prof. Dr. rer. nat. Kurt Schneider
Zweitprüferin: Dr. rer. nat. Jil Ann-Christin
Klunder

Betreuerin: Hannah Luca Deters

Hannover, 25.09.2023

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 25.09.2023

Nino Baumgart

Zusammenfassung

Durch die immer weiter steigende Komplexität von Software-Systemen steigt auch immer mehr die Frage nach guter Erklärbarkeit. Der Mangel oder falsche Einsatz von Erklärbarkeit kann beim Nutzer zu Frustration oder gar zum Wechsel der Software führen. Um dies zu vermeiden, muss also möglichst die richtige Menge und der richtige Zeitpunkt gefunden werden, an denen ein Nutzer Erklärungen benötigt. Dafür ist es nötig, zu wissen, wann ein Nutzer Erklärungsbedarf hat.

Im Rahmen dieser Arbeit wurde eine Software entwickelt, welche mit einer smarten Armbanduhr gesammelten Bioparameter einliest, verarbeitet und visualisiert. Zu diesen Parametern gehören unter anderem Puls, Temperatur und Hautleitfähigkeit. Anschließend wurde eine Nutzerstudie mit drei Teilnehmern durchgeführt. Bei dieser Studie sollten die Teilnehmer eine ihnen unbekannt Software mit einigen Unvorhersehbarkeiten testen. Parallel dazu wurden die Bioparameter der Teilnehmer mit der Uhr aufgezeichnet. Die gesammelten Daten wurden dann mithilfe der Software verarbeitet und ausgewertet.

Die Ergebnisse der Studie zeigten, dass es bei einigen dieser Parameter Ausschläge gab, welche zeitlich zu Ereignissen während der Studie passen. Somit ist es möglich, mithilfe der Uhr und der Software herauszufinden, ob und wann ein Nutzer Probleme mit einer Software hat. Daraus können sich gegebenenfalls Vermutungen über den Erklärungsbedarf aufstellen lassen.

Abstract

Concept and implementation of a data analysis tool for a physiological watch

Due to the ever-increasing complexity of software systems, the question of good explainability is also rising. The lack or wrong use of explainability can lead to frustration or even a change of software. To avoid this, it is necessary to find the right amount and the right time when a user needs explainability. To do this, it is necessary to know when a user needs an explanation.

In the context of this work, software was developed that reads, processes and visualises bioparameters collected with a smart wristwatch. These parameters include pulse, temperature and skin conductivity. A user study was then conducted with three participants. In this study, the participants were to test a software that they were not familiar with and which contained certain unexpected elements. In parallel, the participants' bioparameters were recorded with the watch. The collected data was then processed and analysed using the software.

The results of the study showed that there were spikes in some of these parameters that matched events during the study. Thus, with the help of the clock and the software, it is possible to find out if and when a user has problems with a software. From this, it may be possible to make assumptions about the need for explanation.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	2
1.2	Lösungsansatz	2
1.3	Struktur der Arbeit	3
2	Grundlagen	5
2.1	Erklärbarkeit	5
2.2	Empatica Watch	5
2.2.1	Elektrodermale Hautleitfähigkeit	6
2.2.2	Blut Volumen Puls und Herzrate	8
2.2.3	Avro Format	8
2.3	Dependency Injection	9
2.4	Verwandte Arbeiten	10
3	Forschungsfragen	13
4	Entwicklung des Analysetools	15
4.1	Anforderungen	15
4.2	Framework	16
4.3	Konzept und Planung	16
4.4	Umsetzung	18
4.5	Abweichungen	22
4.6	Probleme bei der Entwicklung	22
5	Nutzerstudie	25
5.1	Planung	25
5.1.1	Teilnehmer	25
5.1.2	Ablauf	25
5.1.3	Testobjekt	26
5.2	Ablauf	27
5.3	Probleme bei der Studie	28

6	Auswertung und Diskussion	31
6.1	Auswertung der Teilnehmerdaten	32
6.1.1	Teilnehmer 1	32
6.1.2	Teilnehmer 2	35
6.1.3	Teilnehmer 3	37
6.2	Beantwortung der Forschungsfragen	40
6.2.1	Beantwortung von Forschungsfrage 1	40
6.2.2	Beantwortung von Forschungsfrage 2	41
7	Zusammenfassung und Ausblick	43
7.1	Zusammenfassung	43
7.2	Ausblick	44
A	Konzeptentwürfe	45

Kapitel 1

Einleitung

Durch die immer steigende Komplexität und Vielschichtigkeit von Software-Systemen steigt auch die Anforderung an gute Qualität, insbesondere an die Erklärbarkeit. Gerade auch im Bereich des maschinellen Lernens und den damit verbundenen Algorithmen und Ausgaben gibt es einen großen Bedarf an Erklärbarkeit [4]. Dabei ermöglichen Erklärungen Verständnis, fördern Vertrauenswürdigkeit, verbessern Benutzerfreundlichkeit, helfen bei der Lokalisierung von Fehlerquellen und können die Wahrscheinlichkeit menschlichen Versagens minimieren [16]. Erklärbarkeit kann auch Nachteile mit sich bringen, wie zum Beispiel dass Hinweise für die Benutzung einer Software an falschen Stellen oder zu falschen Zeitpunkten angezeigt werden. Dies könnte Frustration bei Nutzern auslösen und im schlimmsten Fall sogar dazu führen, dass sie zu der Software eines anderen Anbieters wechseln. Für Softwareentwickler und Designer ist es also wichtig, diese Erklärbarkeit im richtigen Maß, zu den richtigen Zeitpunkten und an den richtigen Stellen einzubinden, um dem Nutzer Vertrauen in die Software zu ermöglichen.

Ein Ansatz, dieses Problem anzugehen oder es zumindest für die Entwickler zu vereinfachen, war es, in einem Test den Nutzer selbst entscheiden zu lassen, wann und welche Erklärung er haben möchte. Daraus wurden dann Erkenntnisse gezogen, wann und bei welchen Aspekten die Nutzer Erklärungsbedarf hatten. Allerdings ist dieser Ansatz dadurch beschränkt, dass eine relativ große Anzahl an Nutzern nötig ist, um ein repräsentatives Ergebnis zu erhalten. Dieses Ergebnis ist dann auch nur repräsentativ für die eine bestimmte Software [11]. Das Problem dieses Ansatzes ist es, dass nur gezeigt werden kann, wann ein Nutzer Erklärungsbedarf hat. Da aber jede Person individuellen Bedarf besitzt, liefert diese Methode nicht immer für jeden das gewünschte Ergebnis.

Um den richtigen Einsatz von Erklärbarkeit effektiver zu ermitteln, kann ein anderer Ansatz gewählt werden, bei dem das Biofeedback, also Daten wie zum Beispiel Puls und Stress aufgezeichnet werden. Dies geschieht mithilfe einer dafür vorgesehenen Uhr im Rahmen eines Softwaretests.

Anschließend können die gesammelten Daten interpretiert werden, um daraus Erkenntnisse bezüglich des Erklärungsbedarfs gewinnen zu können. Das übergeordnete Ziel ist es, mit diesen Erkenntnissen eine Methode zu entwickeln und die Bioparameter in Echtzeit auszulesen und zu verarbeiten, um den Nutzern sofort Hilfestellungen und Erklärungen zu der Software zu geben.

1.1 Problemstellung

In der Einleitung wurde bereits kurz erläutert, dass Erklärbarkeit eine wichtige Rolle bei komplexen Software-Systemen spielt und dass der richtige Einsatz dieser nicht immer einfach ist. Es ist also wichtig herauszufinden, wann und an welchen Stellen Erklärungen eingesetzt werden sollten. Hierfür können mit einer Armbanduhr, welche verschiedenste Bioparameter des Menschen, wie beispielsweise Puls und Temperatur auslesen kann, Daten gesammelt und ausgewertet werden. Die Empatica Watch, welche für diese Auswertung verwendet werden soll, gibt die diese Daten nur in rohen binary Format aus. Vom Hersteller Empatica gibt es eine Software, welche die Daten nur in einminütigen Schritten auswertet. Die Uhr soll aber für Studien eingesetzt werden, wo die Daten im Sekundenbereich ausgewertet werden müssen. Dabei stellt sich die Frage, inwieweit die Parameter, welche die Uhr misst und ausgibt überhaupt geeignet sind festzustellen, ob Erklärungsbedarf besteht. Die Daten müssen dabei mit einem Video aus einer Studie verbunden werden, um überhaupt die Aussagekraft dieser ermitteln zu können.

1.2 Lösungsansatz

Um die Daten synchronisiert mit dem Video zu visualisieren, muss zunächst eine Software entwickelt werden, die die Rohdaten, welche die Empatica Watch ausgibt, auslesen, verarbeiten und anschließend grafisch darstellen kann. Zunächst werden mithilfe der Software die Daten verarbeitet und anschließend grafisch dargestellt, wobei getestet wird, ob die Software die Anforderungen an diese Aufgaben erfüllt. Anschließend soll eine kleine Nutzerstudie stattfinden, in welcher die Nutzer eine Software zum testen bekommen, in der sie eine Sequenz von Aufgaben ausführen müssen, die sie mit unerwarteten Situationen konfrontieren. Diese gesammelten Daten aus der Studie werden im Anschluss mithilfe der Software analysiert und ausgewertet, um überprüfen zu können, ob und welche Biomarker schnell genug ausschlagen, um überhaupt eine Erkenntnis aus ihnen ziehen zu können.

1.3 Struktur der Arbeit

Die Struktur dieser Arbeit ist in folgende Kapitel eingeteilt. Zunächst werden in Kapitel 2 dieser Arbeit die Grundlagen, wie Erklärbarkeit und die verschiedenen Parameter der Uhr erläutert. Außerdem wird Abgrenzung zu verwandten Arbeiten vorgenommen. Dann werden kurz die Forschungsfragen 3 erläutert. In Kapitel 4 wird dann auf das Konzept, die Umsetzung und die Probleme der Software eingegangen, die im Rahmen dieser Bachelorarbeit entwickelt wurde. Kapitel 5 befasst sich mit der kurzen Nutzerstudie und anschließend folgt die Auswertung in Kapitel 6. Abschließend werden die Ergebnisse dieser Arbeit in Kapitel 7 zusammengefasst und ein Ausblick gegeben.

Kapitel 2

Grundlagen

2.1 Erklärbarkeit

Für die Definition von Erklärbarkeit wird in dieser Arbeit die Definition von Chazette et al. [3] im Kontext des Requirements Engineering übernommen. Sie definieren Erklärbarkeit dabei wie folgt.

A system S is explainable with respect to an aspect X of S relative to an addressee A in context C if and only if there is an entity E (the explainer) who, by giving a corpus of information I (the explanation of X), enables A to understand X of S in C .

Dabei kann der Adressee A mehrere Stakeholder, wie Nutzer, Tester und Entwickler, beinhalten. Beispielsweise erhält ein Nutzer (A) beim Bedienen einer Software (C) eine Information oder Hilfestellung (I), um einen Aspekt der Software (X) zu verstehen. Köhl et al. [16] definieren Erklärbarkeit dabei ebenfalls noch als nicht funktionale Anforderung. Dadurch, dass nicht-funktionale Anforderungen oft subjektive Eigenschaften haben, ist auch das Bedürfnis nach Erklärbarkeit von Person zu Person unterschiedlich. Droste et al. haben dabei vier unterschiedliche Personas identifiziert, welche unterschiedliche Ansprüche an Erklärbarkeit haben. Zum Beispiel gibt es den “Minimalist”, welcher nur wenig Erklärbarkeit benötigt, da dieser lieber selbst die Software entdeckt [7]. Hinzu kommt, dass Erklärungen auch von jedem unterschiedlich interpretiert und verstanden werden wie Chazette et al. in einer Umfrage herausgefunden haben [4].

2.2 Empatica Watch

Die Empatica EmbracePlus ist eine smarte Armbanduhr, die in dieser Bachelorarbeit verwendet wurde. Sie wurde speziell zur Erfassung physiologischer Daten entwickelt und ist mit einer Vielzahl an Sensoren ausgestattet, um verschiedene Parameter aufzuzeichnen. Dazu gehören *Accelerometer*,

Gyroscope, *EDA*, Temperatur, *Tags*, *BVP*, *systolic Peaks* und Schritte [9]. Der *Accelerometer* misst die Beschleunigung der Uhr in den X-, Y- und Z-Richtungen und kann somit die Bewegungen des Arms messen. Dazu gehört auch der Gyro-Wert, der die Rotationsgeschwindigkeit anzeigt. Der Gyro-Sensor kann außerdem Schritte erkennen und aufzeichnen. Die Hauttemperatur wird als Temperatur in Grad Celsius gemessen. Die *Tags* sind Zeitstempel, die gesetzt werden, sobald ein Knopf an der Uhr etwas länger gedrückt wird. Diese Zeitstempel eignen sich dazu, ein Zeitfenster für eine Messung aufzuzeichnen oder bestimmte Ereignisse zu markieren. Für diese Arbeit sind die elektrodermale Aktivität (*EDA*), der Blutvolumen Puls (*BVP*), die Temperatur und die systolischen Peaks relevant, da diese Rückschlüsse auf Empfindungen zulassen [12]. Diese Werte werden von den Sensoren der Uhr mit unterschiedlichen Abtastraten aufgenommen. Dabei hat die Temperatur eine Abtastrate von 1Hz, *BVP* 64Hz und *EDA* 4Hz. *Accelerometer* und *Gyroscope* haben in einem Softwaretest aufgrund des geringen Bewegungsumfangs keine Bedeutung. Der *EDA*-Wert, der *BVP*-Wert und die dazugehörigen systolischen Peaks des *BVP*-Wertes werden in den Abschnitten 2.2.1 und 2.2.2 näher erläutert.

Vom Hersteller gibt es die Software *Empatica Care* [8], mit der die gesammelten Daten der Uhr ausgelesen werden können. Allerdings ist diese Software nur für Langzeitstudien geeignet, da die Werte lediglich in einminütigen Abständen abgerufen werden können.

2.2.1 Elektrodermale Hautleitfähigkeit

Elektrodermale Aktivität (*EDA*) oder auch Hautleitfähigkeit wurde erstmals von Johnson und Lubin (1966) eingeführt. *EDA* ist dabei ein gemeinsamer Begriff für alle elektrische Phänomene in der Haut, einschließlich aller aktiven und passiven elektrischen Eigenschaften, die auf die Haut und ihre Bestandteile zurückgeführt werden können [2]. Psychologen verwenden Messungen der elektrodermalen Aktivität seit der Jahrhundertwende [20]. Dabei ist *EDA* ein Maß für die elektrische Aktivität der Haut aufgrund der Schwankungen beim Schwitzen des menschlichen Körpers und hängt dabei von der Intensität der Emotionen ab, wobei die Veränderungen bei hoher Erregung und emotionaler Intensität deutlicher sind [12]. Die Messung der *EDA* ist standardisiert und erfolgt über ein zwei Elektrodensystem, zwischen welchen dann die Spannung gemessen werden kann. Unterschieden wird dabei in passive (tonische) und aktive (phasische) elektrische Eigenschaften. Die phasische Aktivität wird dabei auch als Antwort bezeichnet und ist normalerweise nach circa 1-2 Sekunden mess- beziehungsweise erkennbar. Diese wird hauptsächlich zur Messung und Bestimmung von *EDA* verwendet und in Mikrosiemens angegeben. Die tonische Aktivität wird dabei aus der phasischen Aktivität abgeleitet, was bedeutet, dass die phasische Aktivität vorher bekannt sein muss. Die tonische Aktivität hat in der Praxis eine

weniger signifikante Rolle, da sie weniger empfindlich auf Veränderungen der Versuchsbedingungen reagieren. Daher wird sie meist nicht weiter ausgewertet [2]. Die am häufigsten benutzte Methode zur Auswertung von EDA ist die Analyse der Amplituden. Betrachtet wird dabei die Veränderung zum Durchschnittswert und wie stark ein Anstieg ist. Es gibt dabei mehrere Arten der Verläufe einer Amplitude. Diese werden eingeteilt in einphasig, zweiphasig und dreiphasig (Abbildung 2.1). Betrachtet wird dabei auch die Recoveryzeit, bis der Wert wieder dem Durchschnittswert von vor dem Anstieg gleicht. Diese kann in den in den meisten Fällen aber nicht genau bestimmt werden. Da die mathematische Auswertung von phasischen Aktivitäten nicht genau und sehr kompliziert ist, wird dazu geraten diese per Hand und durch Betrachten auszuwerten [2]. Als Richtwert für einen Anstieg wird üblicherweise ein Unterschied von $0,05\mu\text{S}$ über fünf Sekunden genommen[15]. Es gibt einige Faktoren, welche Einfluss auf Genauigkeit der Messung und Korrektheit der Ergebnisse nehmen. Dazu gehören beispielsweise Altersunterschiede der Personen, Geschlecht, Klima, Verdunstung von Wasser und Feuchtigkeit der Haut, Hauttemperatur und Blutfluss. Einfluss und Auswirkung dieser Faktoren können in allen Fällen allerdings nicht generalisiert werden [2]. Messergebnisse und Tests sollten dennoch in Hinsicht auf diese Faktoren berücksichtigt werden.

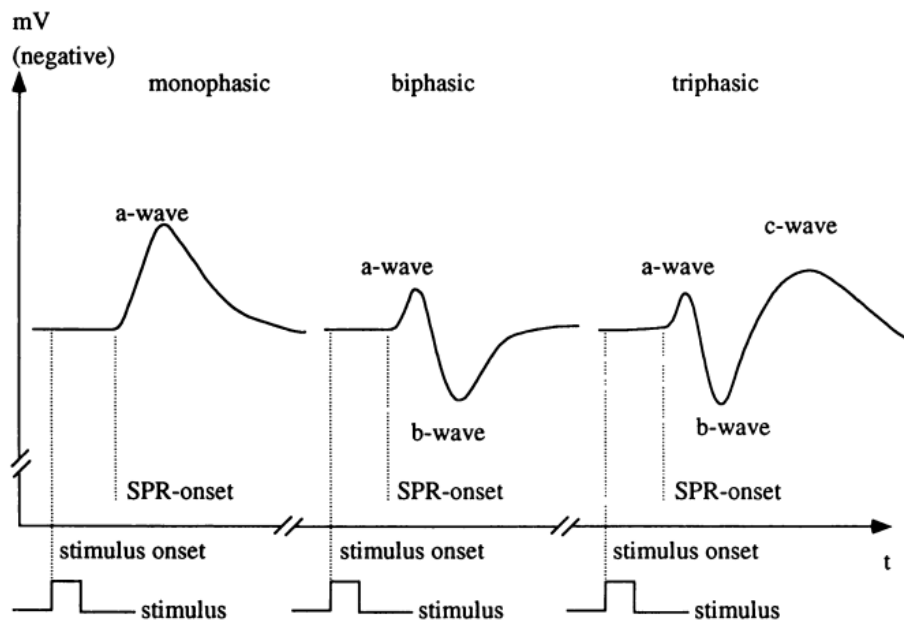


Abbildung 2.1: EDA Amplituden Arten [2]

2.2.2 Blut Volumen Puls und Herzrate

Blood Volume Pulse (BVP) bezieht sich auf die Veränderungen des Blutvolumens in den Gefäßen und wird mit einem einzigen, nicht-invasiven optischen Sensor gemessen, der die Veränderungen in der Lichtabsorptionsdichte der Haut und des Gewebes erfasst, wenn diese beleuchtet werden [15]. Die dadurch entstehende Wellenform bezeichnet man auch als Photoplethysmography (PPG) [6]. Aus dieser Wellenform lassen sich mehrere Dinge ableiten. Diese sind der Durchschnitts BVP-Wert, die Bereiche vom niedrigsten zum höchsten Wert sowie der zeitliche Abstand der Spitzen, welcher auch als Herzrate (Puls) bezeichnet wird [15]. Die Empatica Watch gibt diese Spitzenwerte (*Sytolic Peaks*) separat zum BVP-Wert aus, sodass eine Berechnung dieser nicht mehr notwendig ist und direkt die Herzrate ermittelt werden kann. Sowohl der BVP-Wert selbst als auch Herzraten und Herzratenvariabilität lassen zu, Rückschlüsse auf die Emotionen und Stress zu führen [5]. Auch auf die Messung und Ergebnisse des BVP-Werts nehmen einige Faktoren Einfluss, wie zum Beispiel Stoffwechsel, Gefäßdynamik, arterieller Blutdruck und Fluss sowie die Oberflächentemperatur. Das Ausmaß dieser Faktoren ist dabei auch individuell und kann nicht generalisiert werden [15].

2.2.3 Avro Format

Die Daten im Avro-Format sind ähnlich strukturiert wie bei einer *JSON*-Datei. Das bedeutet, dass alle Daten und Informationen verschachtelt sind. Das Schema besteht aus einem großen Array mit allen Feldern. Dies enthält zunächst einige Informationen über die Uhr, wie zum Beispiel Format und Schemaversion. Der für diese Arbeit relevante Teil ist der Abschnitt "raw data". In diesem sind die einzelnen Parameter wie Temperatur und Beschleunigungssensor auch wieder in separate Bereiche aufgeteilt. In Abbildung 2.2 ist ein Auszug aus dem Schema dargestellt, welcher die Struktur eines der Parameter zeigt. Zunächst wird der Name des Parameters angegeben, gefolgt von einem Feldtyp und anschließend einer Dokumentation, die eine kurze Erklärung des Parameters enthält. Eine Erklärung lautet dabei zum Beispiel: "Data from the electrodermal activity sensor expressed as microsiemens [μ S]. Uses a galvanic skin response sensor.". Diese Erklärung beschreibt dabei nur, wie der Wert gemessen wurde und nicht was er aussagt. Im Typ-Array wird der Typ selbst, sowie der Name angegeben. Nun werden die relevanten Abschnitte für diese Arbeit, die Felder, aufgeführt. Dazu zählen die Timestamps des Parameters, die Sampling Frequenz, das Array mit den Werten selbst und nochmal eine kurze Erklärung. Beim Auslesen hat jedes Fields-Array den Typ Generic-Record. Ein Generic-Record ist eine Instanz eines Record-Schemas, bei dem die Felder über Namen oder Index erreichbar sind [1]. Um die Werte auszulesen, müssen passende Felder aus mehreren Generic-Records herausgesucht werden. Zu

jedem Parameter muss der Typ angegeben werden, in dem das Array mit den Values ausgelesen wird. Beispiele dafür sind Floats für *BVP* und *EDA*, sowie Longs für *Systolic Peaks* und *Tags*.

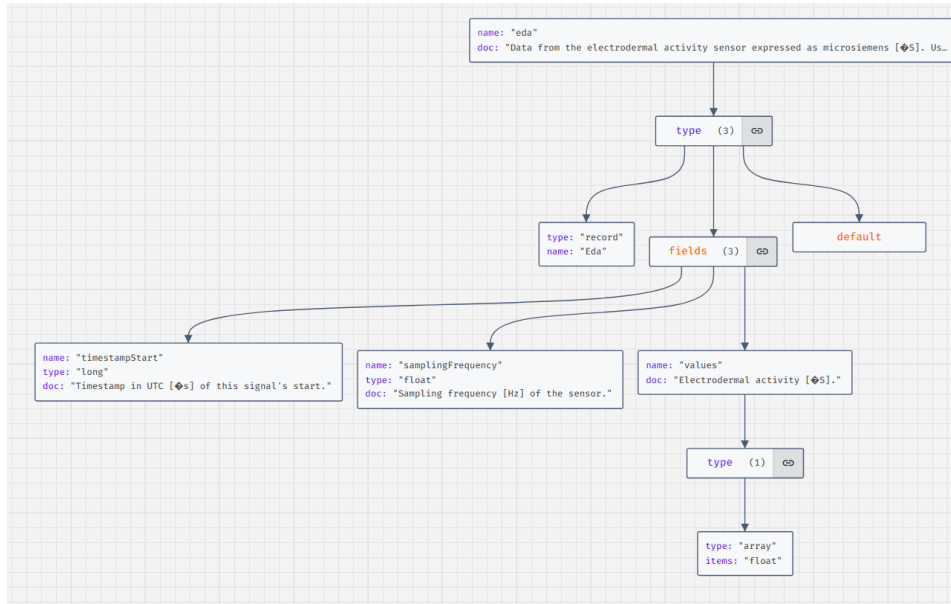


Abbildung 2.2: EDA Schema

2.3 Dependency Injection

Für das in dieser Arbeit entwickelte Programm wurde das Software-Design-Pattern Dependency Injection (DI) genutzt, genauer Interface Injection. Dependency Injection beschreibt ein Design Pattern in der Softwareentwicklung, welches auf dem Inversion of Control Prinzip basiert [19]. Das Pattern bietet viele Vorteile für die Softwarequalität. Dazu gehören zum Beispiel Wiederverwendbarkeit, Erweiterbarkeit und Testbarkeit. Dies ist dadurch möglich, da Klassen unabhängig voneinander sind, weil die Abhängigkeiten getrennt wurden [21]. Bei der DI werden Instanzen von Objekten oder Klassen als Parameter in den Konstruktor anderer Klassen übergeben und dort der Membervariable zugewiesen, statt direkt in der Klasse initiiert zu werden. Dabei werden die meisten Objekte, wenn das Programm nur eine Instanz von dieser Klasse benötigt, in der Main-Methode initialisiert und an die davon abhängigen anderen Klassen injected. Bei der Interface Injection werden dabei Parameter in den Konstruktoren als Interfaces von Klassen erwartet, um die Abhängigkeit weiter zu verringern.

2.4 Verwandte Arbeiten

Girardi et al. [12] verwendeten die Empatica Watch zur Analyse von biometrischer Daten. Sie berichten in ihrem Paper *Emotions and Perceived Productivity of Software Developers at the Workplace* darüber, wie Emotionen kognitive Fähigkeiten bei der Softwareentwicklung beeinflussen. Insbesondere sind sie dabei auf Kreativität und Problemlösungsfähigkeiten eingegangen und haben dies anhand empirischer Beweise nachgewiesen. Hierfür wurde eine Feldstudie mit fünf professionellen Entwicklern aus verschiedenen Unternehmen durchgeführt. Sie stellen zudem eine Taxonomie der Auslöser für positive und negative Emotionen von Entwicklern vor, die durch eine qualitative Analyse der täglichen Berichte der Entwickler erstellt wurde. Girardi et al. stellten schließlich fest, dass ein Minimum an nicht-invasiven biometrischen Daten als Indikator für Emotionen verwendet werden kann, vorausgesetzt das Training wird individuell durchgeführt.

Sie diskutieren in ihrer Arbeit, ob Biomarker zur Analyse von Emotionen eingesetzt werden können, gehen aber nicht auf die Frage ein, inwiefern diese sich auf Software oder Erklärbarkeit beziehen lassen.

Fritz und Müller [17] zeigen in ihrem Artikel *Using (Bio)Metrics to Predict Code Quality Online* auf, wie biometrische Daten genutzt werden können, um Probleme bei der Code-Qualität von Entwicklern frühzeitig zu erkennen und zu beheben. Außerdem haben sie in einer Feldstudie über zwei Wochen mit zehn professionellen Entwicklern Daten gesammelt, um Code-Qualitätsprobleme zu ermitteln. Ihre Studienergebnisse zeigen, dass es möglich ist, Qualitätsprobleme größtenteils vorausszusagen, während ein Entwickler den Code entwickelt. Diese Ergebnisse wurden in einer unabhängigen Zweitstudie mit fünf weiteren Entwicklern bestätigt. Dies kann zu Senkungen der Entwicklungskosten führen. Als abschließendes Ergebnis zeigen die Studien, dass die Biometrie Potenzial hat, Probleme in der Code-Qualität vorherzusagen.

Nun stellt sich die Frage, ob es mithilfe dieser Technik auch möglich ist, Nutzern während der Verwendung einer Software in Echtzeit die richtige Erklärbarkeit bereitzustellen. Es ist wichtig, zunächst zu überprüfen, ob diese Biomarker innerhalb so kurzer Zeit ausschlagen und messbar sind, sodass rechtzeitig ermittelt werden kann, ob Erklärungsbedarf besteht.

Eren und Navruz [10] präsentieren in ihrem Artikel *Stress Detection with Deep Learning Using BVP and EDA Signals*, wie mithilfe Künstlicher Intelligenz und physiologischer Daten Stress erkannt und reduziert werden kann. Die Autoren verwenden dazu Daten der Empatica E4, welche BVP- und EDA-Daten sammelt, und lassen diese Daten von einem künstlichen Netzwerk analysieren. Das Netzwerk identifiziert Stress mittels binärer Klassifikation mit einer Genauigkeit von 92,26 Prozent.

Auch hier zeigt sich, dass Stress mithilfe der Empatica Watch und Biomarker erkannt werden kann. Allerdings stellt sich die Frage, ob dies auch kurzfristig und in Echtzeit möglich ist.

Kapitel 3

Forschungsfragen

Aus den bisherigen Kapiteln lassen sich schlussendlich zwei wichtige Forschungsfragen ableiten, welche es in dieser Arbeit zu klären gilt. Die Hintergründe dieser Fragen werden in diesem Kapitel erneut kurz zusammengefasst. Die Forschungsfragen werden anschließend unten aufgelistet.

Wie bereits in der Einleitung 1 erläutert, spielt die Frage nach der richtigen Menge an Erklärbarkeit und wie diese ermittelt werden kann eine wichtige Rolle. Dabei gilt es nun herauszufinden, ob mithilfe der Empatica Watch Bioparameter aufgezeichnet und ausgewertet werden können, sodass diese Parameter Rückschlüsse auf den Bedarf an Erklärungen schließen lassen. Die Frage, die sich daraus ergibt ist, ob und vor allem welche Parameter sich für diesen Zweck eignen. Aus der Forschungsfrage 1 lässt sich auch die Zweite ableiten, wenn man den Umstand betrachtet, der in der Problemstellung 1.1 bereits erwähnt wurde. Dabei lässt es die vom Hersteller der Empatica Watch bereitgestellte Software nicht zu, Analysen von Bioparametern von nur wenigen Minuten zu erstellen. Daraus resultiert, dass es um Forschungsfrage 1 beantworten zu können, notwendig ist eine Software zu erstellen, welche eine solche Analyse zulässt. Deshalb ist die dabei zu klärende Frage also, wie man die Daten der Uhr einlesen, verarbeiten und visualisieren kann, sodass sie anschließend ausgewertet werden können.

RQ 1 Welche Parameter der Empatica Watch eignen sich zur Auswertung des Erklärungsbedarfs?

RQ 2 Wie können die Daten der Empatica Watch zur Auswertung nach Erklärungsbedarf vereinfacht visualisiert werden?

Kapitel 4

Entwicklung des Analysetools

4.1 Anforderungen

Um die in den Kapiteln 1.1 und 1.2 genannten Punkte umsetzen zu können, wurden einige Anforderungen an die zu entwickelnde Software aufgestellt.

Die erste Anforderung (1) ist die parallele Einbettung eines Videos zur grafischen Darstellung der Daten. Das Video soll eine Bildschirmaufnahme eines Softwaretests zeigen und gegebenenfalls auch die testende Person in Aktion. Dadurch können eventuelle Reaktionen auf die Software erfasst und mit den Daten abgeglichen werden. Zusätzlich muss das Video synchron mit der grafischen Auswertung der Daten laufen, was eine weitere Anforderung (2) darstellt. Die Software musste außerdem die folgenden Anforderungen erfüllen: (3) Die Daten der Uhr und Video sollten in einem Ordner gespeichert und von der Software eingebunden werden können. Außerdem sollte es möglich sein, eine Zip-Datei im Avro-Format der Uhr in den Ordner zu legen. (4) Eine weitere Anforderung bestand darin, dass das Programm die Zip-Datei direkt entpacken können sollte. Abschließend ergeben sich noch zwei weitere logisch-funktionale Anforderungen, die für die Funktionalität unerlässlich sind. Erstens müssen die Daten im Rohformat in eine dafür vorgesehene Klasse eingelesen werden (5) und zweitens müssen sie grafisch dargestellt werden (6). Eine wichtige nicht funktionale Anforderung (7) war, dass die Daten und Ausgaben der Software ebenfalls mit ausreichend Erklärbarkeit versehen sind, sodass dem Nutzer jederzeit eine Erklärung zur Verfügung steht, was die Parameter und Werte aussagen beziehungsweise wie sie berechnet wurden. Abschließend sollte die Benutzeroberfläche mit ausreichend Usability designt werden und dadurch eine intuitive und benutzerfreundliche Bedienung ermöglichen (8).

Basierend auf diesen Anforderungen wurde ein Konzept zur Umsetzung der Software entwickelt, welches im Folgenden detailliert beschrieben und erläutert wird.

Tabelle 4.1: Anforderungen

Nr.	Anforderung	Art
1	Parallel zu den Daten ein Video einbetten	Funktional
2	Video Synchron zur Auswertung der Daten	Funktional
3	Daten aus passendem Ordner auslesen	Funktional
4	Zip-Datei mit Rohdaten entpacken	Funktional
5	Daten in passende Klasse einlesen	Funktional
6	Daten grafisch darstellen	Funktional
7	Erklärbarkeit der Daten und Ausgabe	Nicht-Funktional
8	Benutzeroberfläche intuitiv und benutzerfreundlich (Usability)	Nicht-Funktional

4.2 Framework

Für die Umsetzung des Tools wurde das Framework Kotlin Compose Multiplatform von JetBrains verwendet. Das Framework basiert auf der Programmiersprache Kotlin, welche ebenfalls von JetBrains entwickelt wurde und dem in Kooperation mit Google für Android entwickelten Framework Jetpack Compose. Die Nutzung des Frameworks hat mehrere Vorteile. Zum einen ist es sehr fortschrittlich, da es erst vor kurzem entwickelt wurde und auch Kotlin erst seit 2011 existiert. Zum anderen vereinfacht das Framework die Erstellung Benutzeroberflächen mithilfe des Komponenten-Designs zu bauen. Außerdem bietet Kotlin eine hervorragende Interoperabilität zu Java, sodass sowohl Java Code in Kotlin Projekten, als auch ganze Java Libraries problemlos genutzt werden können. Diese ermöglicht den Zugriff auf eine riesige Menge an bereits bewährten Libraries [13].

4.3 Konzept und Planung

Damit die Anforderungen in der Entwicklung richtig umgesetzt werden können, wurden zunächst zwei Konzeptideen in Form von digitalen Papierprototypen, für die grafische Oberfläche entwickelt. Nachdem die erste Konzeptidee erstellt wurde, gab es eine Evaluation, ob die Anforderungen richtig verstanden wurden. Aufgrund dieser Evaluation wurde dann die zweite überarbeitete Version erstellt. In Abbildung 4.1 sieht man die Zeichnung der ersten Entwurfsidee. Zunächst ist oben links (A) ein Dropdown Menü-Button, über welchen Sprache, Einstellung und das Beenden des Programms erreicht werden können. Oben rechts (B) sollte dann ein kleines Textfeld

sein, über das dieses Programm dem Nutzer Feedback, wie zum Beispiel das erfolgreiche Einlesen der Daten oder auch Fehlermeldungen geben kann. Unter dem Menü Button (C) befinden sich vier verschiedene Buttons. Der Erste mit dem Label “open folder” dient dazu, den Ziel-Ordner für die Daten aus der Uhr und das Video im Dateimanager zu öffnen, da es sonst schwierig sein könnte den Ordner zu finden. Direkt daneben befindet sich der “load data”-Button, welcher nach dem Hinzufügen der Dateien in den entsprechenden Ordner, geklickt wird, um die Daten in das Programm einzulesen, damit diese anschließend angezeigt werden können. Der dritte Button ist ein weiteres Dropdown-Menü, um die Anzahl der Graphen im unteren Bereich der Anwendung zu ändern. Als letztes war noch ein Start-Button vorgesehen, der die Synchronisation beziehungsweise den Ablauf des Programms starten soll.

Der Hauptbestandteil des Konzepts sind die wiederverwendbaren *Graph Objekte*, die aus vielen einzelnen Komponenten bestehen und sich in mehrfacher Ausführung, wie oben beschrieben, anzeigen lassen sollen. Dabei besteht jedes erzeugte Objekt aus einer Box (D). Über dieser Box (E) befindet sich ein erneut ein Dropdown-Menü, mit welchem man zwischen den verschiedenen Werten, wie Puls und Temperatur, wechseln können soll. Daneben werden dann der x-Wert und der y-Wert des aktuellen Werts angezeigt. Die Box mit dem Graphen selbst hat eine gestrichelte Linie (F), die die aktuelle Position synchron mit den anderen Graphen und dem Video anzeigt.

Nach der Besprechung zur ersten Konzeptidee wurde die zweite Konzeptidee (Abbildung A.1) mit ein paar Änderungen erstellt. Zunächst wurde statt eines Graphen ein Bereich (A) für das Video reserviert, da es zum synchronen Ansehen benötigt wird. Dadurch ergibt sich eine Änderung in Bezug auf die Anzahl der Graphen. Diese kann jetzt nicht mehr wie ursprünglich geplant in Zwischenschritten geändert werden, da das Video einen festen Platz hat. Deshalb wurde statt über das Dropdown Menü auswählen zu können, die Möglichkeit geschaffen, bis zu vier weitere Graphen über einen \oplus -Button hinzuzufügen, sodass sich zwei Reihen mit insgesamt fünf Objekten ergeben. Damit die Flexibilität gegeben ist, wie viele Graphen angezeigt werden, hat ein Graph nun auch einen \otimes -Button (B), um diesen wieder entfernen zu können. Zusätzlich gibt es bei jedem Graphen auch einen “Info”-Button, welcher ein Pop-up-Fenster öffnet, in dem nähere Informationen über den ausgewählten Wert und die Berechnung zu finden sind, um die Erklärbarkeit von Anforderung 1 zu gewährleisten. Da der x-Wert bei allen Graphen gleich ist, wurde dieser als Time Anzeige (C) im oberen Drittel hinzugefügt, damit die Zeit vom Video auch besser ersichtlich ist. Außerdem wurden die Buttons im oberen Drittel in zwei Reihen gepackt und der Start Button wechselt nun nach dem Klicken in einen “Stop”-Button.

Anhand dieses Konzepts begann anschließend die Entwicklung der Software.

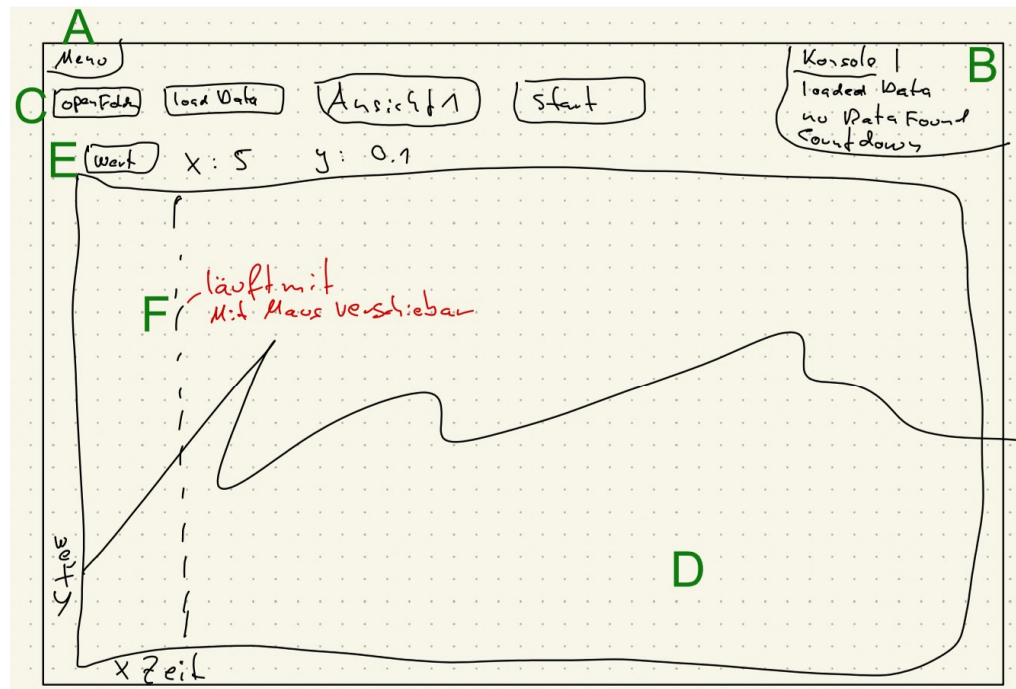


Abbildung 4.1: Seite 1 des Ersten Konzeptentwurfs

4.4 Umsetzung

Bei der Umsetzung war es zunächst wichtig herauszufinden, ob und wie die entscheidendsten und technisch kritischen Anforderungen realisiert werden können. In diesem Fall das Einbetten des Videos in die Software, das Einlesen der Daten in eine geeignete Klasse und das Plotten der Daten als Graph. Davon hing ab, ob das Framework verwendet werden konnte, oder doch zu einem Framework mit stabilerem und besserem Support für Libraries gewechselt werden musste.

Daten

Zunächst musste eine Data-Class geschrieben werden, um Anforderung 7 zu erfüllen, welche die Werte der einzelnen Parameter, also Puls und Hautleitfähigkeit, als Array oder Liste speichert, damit sie anschließend verarbeitet werden können. Data Classes sind Klassen in Kotlin, welche viele Zeilen unnötigen Code vermeiden. Sie benötigen nur Membervariablen. Methoden wie Getter, Setter und "toString" werden zwar nicht automatisch generiert, aber trotzdem kann auf diese zugegriffen werden [14]. Dadurch wird die Wartbarkeit und Lesbarkeit des Codes erhöht.

Der nächste Schritt der Entwicklung war, die Binary-Dateien, welche nach Anforderung 4 in dem dafür vorgesehenen Ordner liegen, auszulesen. Dafür sind mehrere Schritte notwendig, denn wie im Kapitel zum Avro-

Format 2.2.3 bereits erklärt, ist der Zugriff auf die Daten verschachtelt. Dies bedeutet, dass zunächst die richtige URL zur Avro Datei gefunden werden muss, um dann mithilfe eines Java File Readers die Datei öffnen beziehungsweise lesen zu können. Anschließend wird daraus ein Generic-Record extrahiert, aus welchem dann im nächsten Schritt, wie im Schema erkenntlich, die "rawData" auch als Generic-Record extrahiert wird. Somit liegen die rohen Daten vor, in welchen die verschiedenen Parameter der Empatica Watch gespeichert sind. Um nun die einzelnen Parameter auslesen zu können, muss aus der "rawData" der Generic-Record des jeweiligen Paramertyps gezogen werden. Daraus können jetzt in Form eines Generic-Data-Arrays die einzelnen Werte des Parameters in passendem Typ extrahiert und in eine Liste hinzugefügt werden. Diese Listen müssen jedoch noch verarbeitet werden, da zunächst noch die einzelnen Arrays der Parameter unterschiedlich viele Werte beinhalten. Wie bereits oben erklärt, hängt die Anzahl der Werte mit der Samplingrate der Sensoren zusammen. Deshalb mussten die Samplingrate der Parameter noch mit den Werten des Arrays verrechnet werden, um hinterher in jeder Liste gleich viele Werte, also pro Sekunde einen Wert, zu erhalten. Dafür werden beispielsweise bei dem *EDA*-Wert, mit einer Samplingrate von ungefähr 4Hz, immer vier Werte miteinander addiert und durch vier dividiert, um so den Mittelwert zu erhalten und diese vier Werte zu einem Wert zusammenzufassen. Die gleich langen Listen können nun in die DataClass geschrieben und dann weiterverarbeitet werden.

Für Anforderung 8 müsste geprüft werden, ob und wie die Werte als Graph geplottet werden können. Dafür war es notwendig, zunächst einen Rahmen für den Graphen in Form eines Canvas festzulegen, um die Höhe und Breite des Graphen variieren zu können. Dann mussten die minimalen und die maximalen Werte aus den Listen betrachtet werden, damit später die Skalen der x- und y-Achsen passen. Über eine "drawIntoCanvas"-Funktion wurden dann die beiden Achsen mit den richtigen Offsets gezeichnet und dann ein Pfad generiert, welchen die einzelnen Werte aus der Liste übergeben bekommt. Dieser Pfad wurde dann auch in das Canvas gezeichnet, um den Graphen so in der Oberfläche sichtbar zu machen.

Video

Als Drittes sollte das Video in die Software für Anforderung 3 eingebettet werden. Dafür wurde eine Library auf Basis des VLC Players genutzt. Diese hat die Anforderung, dass der VLC Media Player auf dem System installiert sein muss, um die Möglichkeit zu bieten das Video einzubetten. Damit gute Kontrolle über den Player möglich ist. Dafür wird zunächst ein "EmbeddedMediaPlayerComponent" erstellt. Von diesem erhält das Programm dann das nutzbare Player-Objekt. Diesem wird einen URL, also der Pfad zu der im Resource-Ordner liegenden Videodatei übergeben und dann kann diese bereits abgespielt werden. Somit bietet der Player auch einfache Controls wie Start, Stop und das Ändern der Position.

Graph

Der nächste Schritt war die Umsetzung des Graphen mit den dazugehörigen Objekten, beziehungsweise zuerst musste die Logik für die zwei Reihen an Graph-Objekten und das Video geschrieben werden. Dafür wurden die Lazy-Rows des Frameworks genutzt, welche Objekte aus einer Liste mit Abstand untereinander dynamisch nach Anzahl anordnen können, ähnlich wie Flexboxen in CSS. Somit mussten die Graphen nur initiiert und in eine Liste hinzugefügt werden, um dynamisch in dem GUI angezeigt zu werden. Diese ist in Abbildung 4.2 zusehen. Der Graph besteht dabei aus dem oben bereits beschriebenen Plot der Werte und den beiden Achsen. Dazu kommt noch der kleine Button mit dem ⓘ-Button, welcher ein Pop-Up mit Erklärungen zu den Werten und dem Graphen anzeigt und Anforderung 1 erfüllt. Um selbst entscheiden zu können, welcher Graph wo angezeigt wird, kann über ein Dropdown Menü bei jedem Graphen frei entschieden werden, welcher angezeigt wird. Wenn gewünscht, ist es auch möglich, sich mehrfach den Gleichen anzeigen zu lassen.

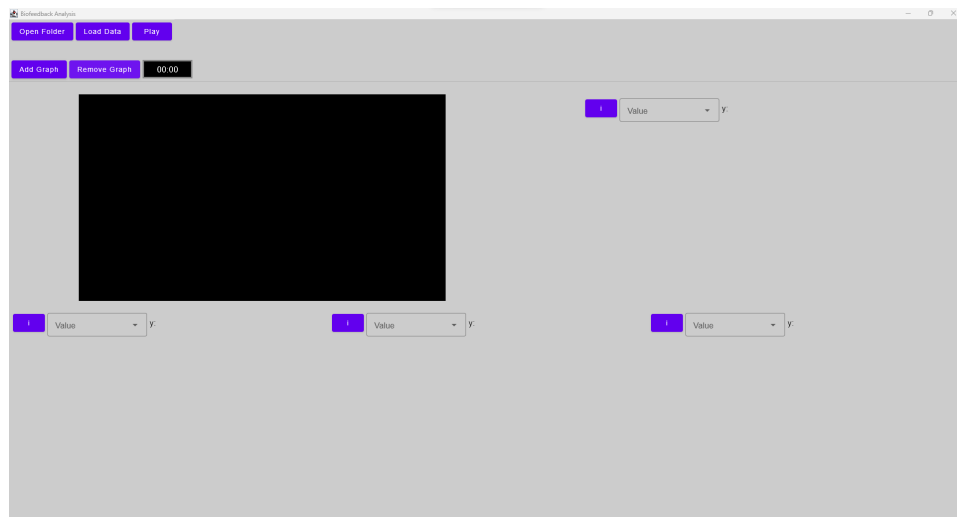


Abbildung 4.2: Programm vor Auswahl der Werte

Um auch klar erkennbar zu machen, an welchem Punkt des Graphen gerade der Wert angezeigt wird, war im Konzept eine gestrichelte Linie vorgesehen, welche pro Sekunde einen Schritt weiter wandert. Auch war es geplant durch einfaches Klicken auf irgendeine Stelle des Graphen, diese Linie dorthin zu setzen und den x- sowie y-Wert anzuzeigen. Dafür musste innerhalb des Canvas eine “detectTapGestures”-Methode verwendet werden, welche, einen prozentualen Float Wert zurückgibt. Über diesen kann dann ermittelt werden, an welcher Stelle die Linie angezeigt werden muss und welcher der zugehörige Wert aus den Daten-Arrays ist.

Intervall

Eine weitere Herausforderung war es, aus den Daten, die die Uhr ausgibt, das richtige Intervall zu finden, in dem die Nutzerstudie stattfand. Wie im Kapitel 2.2.3 bereits beschrieben hat jeder Parameter, welchen die Uhr misst einen Zeitstempel, zu dem die Aufnahme gestartet ist. Außerdem existiert noch ein Parameter *Tags*, welcher den Zeitstempel während des Drückens der Knöpfe an der Uhr speichert. Mit diesem Zeitstempel galt es nun, das Intervall einzugrenzen. Zunächst kam es vor, dass die Uhr sehr viele Dateien ausgab, da sie aufnimmt, sobald sie eingeschaltet wird. Das bedeutet, dass viele der Dateien überflüssig sind und beim Einlesen rausgefiltert werden müssen. Dafür wird mit dem ersten *Tags* Zeitstempel und einem Zeitstempel aus einem der Werte von der zu prüfenden Datei abgeglichen und überprüft, ob der *Tags* Zeitstempel zwischen dem Zeitstempel der Datei und dem der nächsten Datei liegt. Dadurch werden die richtigen Dateien eingegrenzt. Um innerhalb der Datei das richtige Intervall des Zeitraums zwischen dem Drücken der Knöpfe an der Uhr ermitteln zu können, musste der Zeitraum vom Anschalten der Uhr, bis zum ersten *Tags* Zeitstempel ermittelt werden. Mit diesem Zeitraum kann dann der erste Index im Array gefunden werden. Der zweite Index ergibt sich nun, indem man die Differenz der beiden *Tags* Werte in Sekunden errechnet und diesen auf den ersten Index addiert. Somit kann eine Sublist des Arrays zurückgegeben werden, welche das für eine Nutzerstudie korrekte Intervall enthält. Um dann das Video besser synchronisieren zu können, wurde noch ein fester Puffer von fünf Sekunden eingebaut, sodass das Video circa fünf Sekunden vor Drücken der Uhr geschnitten werden kann.

Synchronisation

Der letzte wichtige Schritt war nun die Synchronisation von Video und Graphen. Das Video zu starten und gleichzeitig auch die Linie auf dem Graphen pro Sekunde weiterzuschieben, erwies sich als nicht einfach umsetzbar. Das genaue Problem wird in Kapitel 4.6 noch detailliert erläutert. Die Lösung des Problems war es, einen globalen Synchronisationswert zu initialisieren, welcher sobald der Start-Button geklickt wurde, also das Video lief, jede Sekunde um Eins erhöht wurde. Dieser Wert wurde dann an alle Stellen, wo dieser gebraucht wurde, weitergereicht. Somit konnte auch die gestrichelte Linie im Graphen weiter gesetzt und die Werte aus den Arrays ermittelt werden. Bei Klick auf den Graphen wird auch eben dieser Wert geändert und so auch das Video an die entsprechende Stelle gesetzt.

Anforderung 5 war es, die Daten der Uhr als Zip Datei in den Zielordner legen zu können, weil die Uhr gelegentlich die Daten in mehreren Dateien ausgibt. Damit die Zip Datei entpackt werden konnte, wurde die Java Zip Library genutzt. Diese bekommt dabei einen Inputstream, welcher so lange nach neuen Entries in einer Schleife durchläuft, bis keine mehr vorhanden sind. Dann wird für jeden Entry eine neue Datei erstellt und mittels eines Buffers der Inhalt des Inputstreams in einen Outputstream für die Datei

geschrieben.

Anschließend mussten lediglich noch kleinere Features, wie das Umrechnen des Synchronisationswerts, in ein lesbares Zeitformat umgesetzt werden. Dafür bot es sich an, die bereits existierende Java `formatTime` Library zu verwenden, in welche lediglich der Integerwert übergeben und das Zielformat angegeben werden muss. Es wurden außerdem noch die Logik für das Hinzufügen und Entfernen eines Graphen aus den Reihen im unteren Teil der Software in Form von zwei Buttons implementiert.

4.5 Abweichungen

Aufgrund der technischen Gegebenheiten des Frameworks mussten nochmal einige Änderungen am Konzept vorgenommen werden, welche im Folgenden beschrieben werden. Der erste große Unterschied ist, dass die Textbox oben rechts im Konzept entfernt wurde, weil es schwierig war diese in vernünftiger und lesbarer Größe zu platzieren, dass sie nicht zu viel Platz einnimmt, beziehungsweise Lücken an anderen Stellen schafft. Genaueres zu dem Problem des Layouts findet sich im nächsten Kapitel 4.6. Da diese Textbox nun entfernt wurde, musste ein anderer Ansatz gefunden werden, um dem Nutzer Feedback und Erklärungen zu geben. Deshalb wurden daraufhin Pop-Up-Fenster programmiert, welche dann zum Beispiel das erfolgreiche Laden der Daten oder Fehlermeldungen anzeigen. Das Feld mit der aktuellen Laufzeit des Programms wurde aus den gleichen Gründen auch ein wenig verschoben. Der zweite große Unterschied besteht darin, dass der \oplus -Button zum Hinzufügen eines Graphen umgeändert wurde und mit einem "remove"-Button im oberen Teil zu finden ist und nicht mehr in der Reihe der Graphen. Geplant war, dass beim Klicken der Buttons an dessen Stelle ein Graph erscheint und der Button einen Platz in der Reihe weiter rückt. Dies war aber aufgrund der Funktionsweise der genutzten `LazyRows` nicht optimal umsetzbar. Deshalb gibt es nun zwei einfache Buttons, mit denen Graphen hinzugefügt und wieder entfernt werden können.

4.6 Probleme bei der Entwicklung

Bei der Entwicklung gab es einige Schwierigkeiten und Probleme. Eine große Schwierigkeit stellte der Mangel an Tutorials, Dokumentation und Forenbeiträgen zum Framework dar. Trat ein Error oder eine Fehlermeldung auf, so musste in den meisten Fällen auf Foren zum verwandten Framework `Jetpack Compose` zurückgegriffen werden, wobei die Antworten dort teilweise sehr Android spezifisch waren, was das Finden einer Lösung sehr erschwert hat.

Aus dem bereits genannten Problem folgte, dass erst recht spät in der Entwicklungsphase das Konzept der States vom Framework verstanden

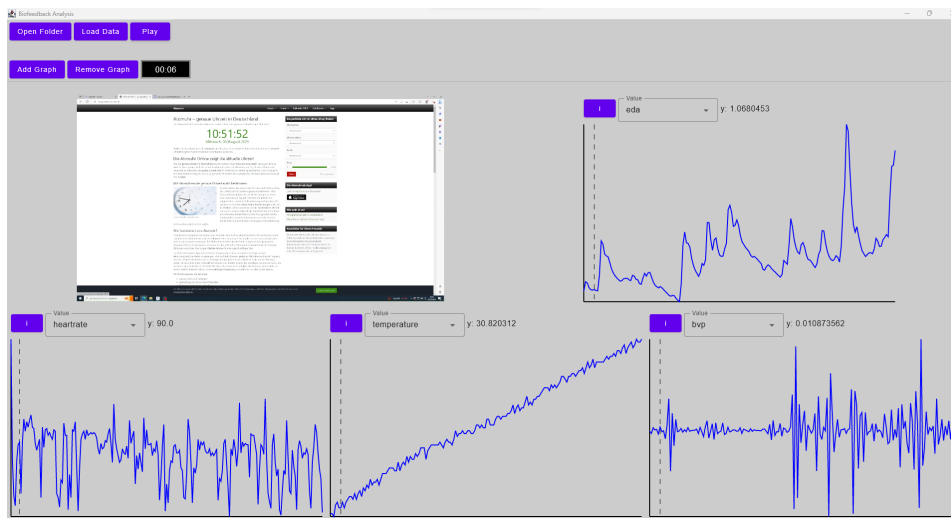


Abbildung 4.3: Finales Layout der Software

wurde, welche essenziell für das Umsetzen der Synchronisation waren. Dies machte es zunächst schwierig eine gute Struktur für den Code und die Klassen zu finden, wenn nicht klar war, wie diese miteinander kommunizieren können. Um also das Problem lösen zu können, musste das Konzept der States verstanden werden. States funktionieren dabei so, dass sie als Variablen oder Objekte gesetzt werden. Diese States können mithilfe des "remember" Keywords gespeichert bleiben. Das bedeutet, dass wenn dieser Wert geändert wird, aktualisieren sich grafische Objekte, in denen dieser Wert benutzt wird, automatisch. Durch das Compose Konzept können UI Objekte und deren Logik in andere Methoden und Klassen ausgelagert werden, indem den Methoden die Annotation "Composable" gegeben wird. So können diese Funktionen in anderen Compose Objekten aufgerufen werden und der Inhalt wird somit an der Stelle eingesetzt. Die Frage war also, wie man den State einer Variable an andere Klassen und Methoden weitergibt, sodass dieser verwendet und geändert werden kann. Hierfür musste die zu übergebene Variable als "remember mutable state" initialisiert und dann als Parameter an die Methoden übergeben werden. Die Methoden müssen dann als Übergabeparameter-Typ einen Mutablestate und den Typ der Variable haben (`MutableState<Float>`). Somit konnte also überall, wo benötigt auf den Synchronisationswert zugegriffen und sobald sich dieser ändert, auch direkt auf der Oberfläche angezeigt werden. Nun konnte das Video mit dem Rest des Programms synchronisiert und eine sinnvolle Codestruktur entwickelt werden, da nun klar war, wie man ausgelagerte Klassen und Objekte miteinander kommunizieren lässt.

Das Video einzubetten erwies sich wie in Kapitel 4.4 erwähnt, zunächst als sehr schwierig, da Kotlin noch keine geeignete Library anbietet. Die von

Jetbrains selbst entwickelte Library ist noch nicht vollständig und deshalb nicht einfach nutzbar. Auch andere Java- und Video Libraries für Android waren nicht geeignet, da diese größtenteils nicht in diesem Framework nutzbar waren. Schlussendlich wurde die in dem experimentellen Videoplayer von Jetbrains verwendete VLCJ Library verwendet und in ähnlicher Form wie dort genutzt. Diese erfordert zwar, dass der VLC Media Player auf dem System installiert ist, bietet aber dafür die für die Software notwendigen Funktionen.

Da die Empatica Watch ihre Daten teilweise mit unterschiedlich langen beziehungsweise genauen Zeitstempeln ausgibt, war es ein Problem, die Zeitstempel der Uhr beim Finden des richtigen Intervalls auszulesen und zu vergleichen, da es keine genaue Dokumentation dazu gab. So sind zum Beispiel die Zeitstempel der *Systolic Peaks* im Nanosekundenbereich, da der Sensor mit einer Frequenz von 64Hz Daten sammelt. Diese dann beispielsweise mit den Zeitstempeln der anderen Parameter zu vergleichen oder den *Tags* des Knöpfe-Drückens, hat kurzzeitig zu Problemen geführt. Als das Problem erkannt wurde, wurde eine Funktion geschrieben, welche die Zeitstempel in lesbarem Format mit Uhrzeit und Datum ausgibt. Auch hierbei wurde wieder die Java Time Format Library genutzt. So konnten die Zeitstempel lesbar ausgegeben und das Entwickeln der Methode zum Finden des Intervalls vereinfacht werden.

Eine zusätzliche Herausforderung stellte das Layout, also wie die Objekte auf der Oberfläche angeordnet sind, teilweise dar. Beispielsweise brauchte die Textbox, die die Fehlermeldungen anzeigen sollte, eine gewisse Größe, um den Text darin gut lesbar zu machen. Daraus folgte, dass in die Row, in der sich Die Textbox befand, auch so hoch war wie die Textbox selbst war. Dadurch entstand zwischen den Buttons auf der linken Seite eine große Lücke, die viel Platz verschwendete. Da es also keine schöne oder optimale Lösung war diese Box einzusetzen, wurde sie durch passende PopUps ersetzt.

Für weitere Schwierigkeiten sorgte der Videoplayer, welcher während des Abspielens zwar zulässt an eine andere Position zu springen, aber das Video dann nicht mehr richtig abgespielt oder an die vorherige Stelle zurückgesetzt wird. Deshalb war es notwendig, das Ändern der Position nur bei gestoppten Videoplayer zuzulassen. Auch das Laden der Daten und vorheriges Starten des Videos führte zu Bugs, welche nicht behoben werden konnten. Deshalb muss auch hier dafür gesorgt werden, dass erst die Daten geladen werden müssen, bevor das Video gestartet werden kann.

Kapitel 5

Nutzerstudie

Das Ziel der Nutzerstudie bestand darin, mithilfe der Empatica Watch Daten zu sammeln, um die Funktionalität und die Anforderungen der entwickelten Software abschließend zu testen. Anschließend werden diese Daten mithilfe der Software ausgewertet und auf ihre Eignung zur Erhebung von Erklärungsbedarf untersucht. Des Weiteren wurden im Rahmen der Studie erste Erfahrungen mit der Uhr für zukünftige Projekte gesammelt.

5.1 Planung

5.1.1 Teilnehmer

Drei männliche Bachelor-Studierende des Fachs Informatik im Alter von Anfang 20 wurden als Teilnehmer für diese Studie ausgewählt, um ein gewisses Grundverständnis von Software sicherzustellen. Es wurde darauf geachtet, dass die Voraussetzungen aller Teilnehmer ähnlich waren, um möglichst vergleichbare Daten zu erhalten. Wie bereits in Kapitel 2.4 erwähnt, können Alter und Geschlecht Auswirkungen auf die Messergebnisse haben.

5.1.2 Ablauf

Bei der Studie sollten die Probanden mit angelegter Empatica Watch, eine ihnen unbekannt Software testen. Diese Software beinhaltete absichtlich eingebaute Fehler und begrenzte Erklärungen, um aussagekräftige Daten zu erlangen, indem Verwirrung und Frustration hervorgerufen wird. Zur Analyse der Daten wurde während der Studie eine Bildschirmaufzeichnung erstellt, die später synchron zur Auswertung der Daten in der Software betrachtet werden konnte. So konnten Handlungen parallel zur Veränderung der Daten verfolgt und somit geprüft werden, ob an den vorgesehenen Stellen ein Ausschlag in den Werten erkennbar ist. Aus Datenschutzgründen wurde bei dieser Studie darauf verzichtet, ein Video der Teilnehmer aufzunehmen.

In anderen Studien könnte eine Aufnahme jedoch Sinn ergeben, weil dieses Video bei der Auswertung gegebenenfalls mehr Rückschlüsse auf die Emotionen und die Gedanken der Teilnehmer geben könnte, da Mimik und Körpersprache hätten gedeutet werden können. Zusätzlich bekamen die Teilnehmer vor der Studie einen Fragebogen ausgehändigt, auf dem die unten genannten Use-Cases, Fragen zum Geschlecht, Alter sowie Platz für Anmerkungen vorgesehen sind. Dort können sie nach Beendigung der Studie noch Gedanken zur Studie oder zur Software notieren, was hinterher bei der Auswertung nützlich sein könnte.

5.1.3 Testobjekt

Für das in der Studie verwendete Programm wurde ein High-Fidelity Prototyp einer Software genutzt, der bereits im Institut entwickelt ¹ und anschließend den Anforderungen der Nutzerstudie angepasst wurde. Die Umsetzung erfolgte mit der Software Axure RP 10, welche die schnelle und einfache Erstellung von Mockups ermöglicht. Dabei wurde die Software an den in dieser Universität genutzten Online-Service für Noten (QIS), Bescheinigungen und Prüfungsübersicht angelehnt und überarbeitet. Der Hauptfokus der Software liegt auf der Übersicht der Prüfungen und Einsicht der Prüfungsergebnisse. Die Startseite mit den drei Bereichen ist in Abbildung 5.1 dargestellt.

In der Prüfungsübersicht können Nutzer ihre angemeldeten Prüfungen einsehen, sich für neue Prüfungen anmelden und die Notenübersicht betrachten. Bei der Prüfungsanmeldung können Nutzer mehrere Prüfungen auswählen, die noch nicht bestanden wurden, und im Anschluss die Anmeldung bestätigen. Diese Prüfungen können dann im Bereich angemeldete Prüfungen eingesehen werden. Eine Liste der bereits absolvierten Prüfungen und deren Noten ist im Bereich Notenübersicht einsehbar. An dieser Stelle wurden zwei Unvorhersehbarkeiten für die Teilnehmer eingebaut. Dabei wurde zum einen das Prüfungsergebnis der neusten Klausur, Logik und Formale Systeme, auf 5,0 und nicht bestanden geändert. Dadurch sollte der Teilnehmer, auch wenn er diese Prüfung nicht selbst geschrieben hat, erstaunt sein und sich dies in den Daten widerspiegeln. Zum anderen wurde bei der Anmeldung von Klausuren eine Fehlermeldung eingebaut. Diese weist den Nutzer darauf hin, dass die Prüfungen nicht angemeldet werden konnten, weil der Zeitraum für die Prüfungsanmeldung abgelaufen sei und er es erneut versuchen oder sich an das zuständige Prüfungsamt wenden soll.

Im Bereich Bescheinigungen und Studienverlauf können eine Übersichtstabelle mit den bisherigen Semestern eingesehen sowie die relevanten Dokumente wie Studienverlauf, Immatrikulationsbescheinigung und vorläufiges Semesterticket heruntergeladen werden. Außerdem gelangt man von dort

¹Dieser Prototyp wurde im Rahmen der Usability Engineering Labors im WiSe 21/22 erstellt

aus auch in den Bereich Notenübersicht.

Im letzten Bereich persönliche Daten können die Anschrift, Telefonnummer, E-Mail-Adresse und das Passwort geändert und eingesehen werden. Auch hier wurden in die Software zwei Unvorhersehbarkeiten eingebaut, auf die die Teilnehmer während der Studie stoßen sollten. Dabei wurde die Änderung der Anschrift so verändert, dass diese nicht gespeichert wird und zudem eine andere Hausnummer angezeigt wird. Dadurch sollte es bei den Teilnehmern zu Verwirrung kommen, was sich durch die Daten bestätigen lassen sollte. Zudem wurden beim Eingeben des Passworts die Textfelder so angepasst, dass die beiden neuen Passwörter übereinstimmen müssen, da sonst ein Hinweis angezeigt wird. Die Teilnehmer müssen dies dann erst korrigieren, bevor sie die Änderung speichern können.

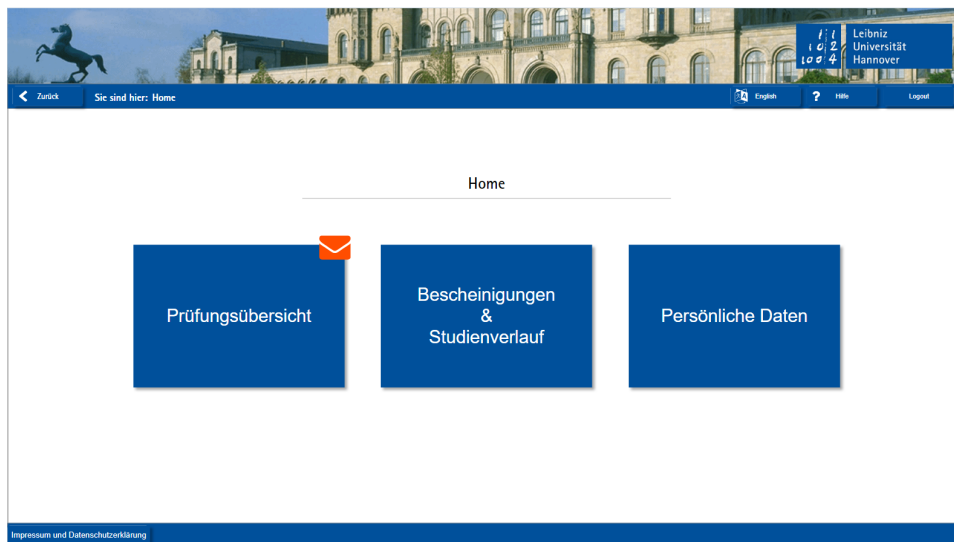


Abbildung 5.1: Startseite der Testsoftware

Aus den in die Software hinzugefügten Unvorhersehbarkeiten und Fehlermeldungen ergeben sich folgende Use-Cases für die Nutzerstudie. Diese Use-Cases zielen darauf ab, durch möglichst viele dieser Unvorhersehbarkeiten bei den Teilnehmern viele Emotionen hervorzurufen, welche sich in den Daten ablesen lassen.

5.2 Ablauf

Die Studien wurden im Abstand von 30 bis 45 Minuten mit einzelnen Teilnehmern durchgeführt. Zu Beginn der Studie wurde den Teilnehmenden zunächst eine Datenschutzerklärung vorgelegt, welche sie unterschreiben mussten. Anschließend wurde ihnen die Uhr umgelegt. Zuerst wurde die Bildschirmaufnahme gestartet und die Zeitstempel an der Uhr gesetzt.

Tabelle 5.1: Use-Cases

Nr.	Use-Case
1	Prüfungsergebnis zur Prüfung Logik und Formale Systeme einsehen
2	Anmelden zur Prüfung FPGA-Entwurfstechnik
3	Anchrift ändern zu Musterstraße 32a und Änderung prüfen
4	Passwort ändern

Nun sollte jeder Teilnehmer die Use-Cases abarbeiten. Während der Studie hatten die Teilnehmer die Möglichkeit, Rückfragen zu stellen. Gegebenenfalls wurden Hinweise gegeben, falls die Teilnehmer nicht wussten, was sie tun sollten. Nach der Studie wurde dann der Zeitstempel der Uhr erneut gesetzt und die Bildschirmaufnahme beendet. Anschließend wurden den Teilnehmern zwei Fragen gestellt, um gegebenenfalls mehr Rückschlüsse auf den Bedarf nach Erklärung und Hilfestellung während der Studie ziehen zu können. Frage 1 soll dabei darauf abzielen vom Teilnehmer zu erfahren, ob dieser überhaupt gemerkt hat, dass der Use-Case nicht wie vorgesehen funktioniert hat. Zu wissen, ob diese Fehlermeldungen oder Probleme in den Use-Cases bemerkt worden sind, kann für die Auswertung der Daten von Relevanz sein. Die zweite Frage dient dazu, zu klären, was die Teilnehmer sich für Erklärungen gewünscht hätten oder ob sie die Vorhandenen ausreichend fanden. Die Fragen sind in der folgenden Liste aufgeführt.

Frage 1 Hast du Fehlermeldungen bekommen?

Frage 2 Hätten mehr Erklärungen in den Fehlerfällen geholfen?

5.3 Probleme bei der Studie

Das erste Problem was auftrat war, dass der Hinweis, warum die Prüfungen nicht angemeldet werden konnten, ursprünglich weggelassen werden sollte, um so wenig wie möglich Erklärung zu liefern. Allerdings wurde diese Änderung in der Software vor der Nutzerstudie nicht mehr gespeichert.

In Use Case 1 war vorgesehen, dass die Teilnehmer auf das kleine orange-farbende Briefsymbol in 5.1 klicken, um direkt zur Notenübersicht zu gelangen. Keiner der Teilnehmer hat dies getan. Des Weiteren gab es einen Fehler bei der Umstellung der Software, sodass bei Teilnehmer 1 die Note für Logik und Formale Systeme als bestanden eingetragen wurde, da er über den Bereich Bescheinigungen und Studienverlauf zur Notenübersicht navigierte. Es gab ein Verlinkungsproblem innerhalb der Software, weshalb die nicht editierte Seite angezeigt wurde.

Alle Teilnehmer fragten bei Use-Case 3 nach dem alten Passwort. Es war softwareseitig irrelevant, ob ein Passwort eingetragen war und wie es lautete.

Das neue Passwort konnte trotzdem geändert werden. Die Teilnehmer wurden daher darauf hingewiesen, dass sie dieses Feld frei lassen können.

Die Teilnehmer haben bei der Passwortbestätigung alle zweimal das gleiche Passwort eingegeben, weshalb die Fehlermeldung, dass die Passwörter nicht übereinstimmen, nicht auftrat.

Entgegen den Erwartungen waren alle drei Durchläufe relativ kurz. Die Teilnehmer klickten sich schnell durch die Use Cases, wobei die ersten beiden Durchläufe ca. zweieinhalb bis drei Minuten dauerten. Der letzte Durchlauf war mit viereinhalb Minuten etwas länger, da der Teilnehmer zwischen den Use-Cases seine Gedanken äußerte. Durch die kürzeren Studien kann es gegebenenfalls vorkommen, dass es keine klar erkennbaren Ausschläge in den Werten gibt, da die Abstände der Use-Cases zu gering sind. Außerdem werden dadurch viel weniger Daten gesammelt, die ausgewertet werden können.

Kapitel 6

Auswertung und Diskussion

Im folgenden Kapitel werden die in der Nutzerstudie gesammelten Daten manuell ausgewertet und analysiert, indem die Diagramme der Parameter einzeln, manuell und dabei in Verbindung mit den Besonderheiten der Use-Cases betrachtet werden. Dabei wird der Verlauf der Studie sowie die Graphen und Werte für jeden Teilnehmer einzeln kurz beschrieben, erläutert und anschließend ein Fazit gezogen. In der folgenden Tabelle sind die Besonderheiten der Use-Cases aufgelistet, welche Emotionen bei den Teilnehmer hervorrufen sollten und wo man Erklärungsbedarf vermuten könnte.

Tabelle 6.1: Auslöser für Emotionen

Nr.	Erklärungsbedarf oder Unvorhersehbarkeit	Art
A	Nicht bestandene Prüfung	Unvorhersehbarkeit
B	Fehlermeldung bei Prüfunganmeldung	Erklärungsbedarf
C	Adressänderung nicht gespeichert	Erklärungsbedarf
D	Eingabe zweier unterschiedlicher Passwörter	Erklärungsbedarf
E	Altes Passwort nicht bekannt	Unvorhersehbarkeit

6.1 Auswertung der Teilnehmerdaten

6.1.1 Teilnehmer 1

Verhalten

Für Use-Case 1 hat der erste Teilnehmer zunächst auf Bescheinigungen und Studienverlauf geklickt, um anschließend dort zur Notenübersicht zu gelangen. Dort war die Note für Logik und Formale Systeme nicht hervorgehoben und als bestanden eingetragen.

Vom Home-Menü aus navigierte der Teilnehmer für Use-Case 2 über die Prüfungsübersicht zur Prüfungsanmeldung, wählte die richtige Prüfung aus und bestätigte die Anmeldung. Nach erneutem Bestätigen erhielt er die Fehlermeldung und versuchte nochmal, die Prüfung anzumelden, mit dem selbem Ergebnis. Daraufhin bekam er den Hinweis, mit dem nächsten Use-Case fortzufahren.

Der Teilnehmer hat die Reihenfolge der Use-Cases vertauscht und änderte zunächst sein Passwort ohne Auffälligkeiten.

Use-Case 3 verlief wie geplant. Es gab keine Fehlermeldungen, aber die Anschrift wurde nicht übernommen. Der Teilnehmer wiederholte den Use-Case. Er merkte an, dass es schwer erkennbar sei, dass man auf das Feld mit der Adresse zum Ändern anklicken musste.

Die Fragen beantwortete der Teilnehmer damit, dass er nur bei Use-Case 1 eine Fehlermeldung bekommen hätte, und die Fehler sonst kaum ersichtlich waren und er es deshalb erneut versucht hätte.

Der Teilnehmer hatte also mit Use-Case 1 und 4 wenig bis keine Probleme und somit auch keinen Erklärungsbedarf. Bei den anderen Use-Cases hingegen hätten dem Teilnehmer Erklärungen geholfen und es gab Erklärungsbedarf.

EDA

In Abbildung 6.1 ist der Verlauf der EDA-Kurve über die gesamte Studie gezeigt. Dabei sind die Zeiträume der Use-Cases und die Zeitpunkte aus Tabelle 6 markiert. Zu erkennen ist, dass der EDA-Wert zu Anfang der Studie extrem hoch war. Dies könnte entweder mit der Aufregung beziehungsweise Nervosität zusammenhängen, wie Boucsein [2] bereits feststellte oder ein Messfehler durch falsches Befestigen der Uhr sein. Nach Use-Case 1 ist ein Ausschlag zu erkennen, obwohl an der Stelle keine Probleme oder Erklärungsbedarf vorhanden waren. Während des ersten Versuchs und der Fehlermeldung von Use-Case 2 sinkt der EDA-Wert des Teilnehmers auf ein Minimum. Nach der Fehlermeldung des zweiten Schritts ist erneut ein kurzer Ausschlag zu sehen (Marker B1 und B2). Unmittelbar vor Eingabe des Passworts ist ein enormer Anstieg (E) zu erkennen. Danach fiel der Wert direkt wieder auf den vorherigen Wert und die Studie wurde beendet.

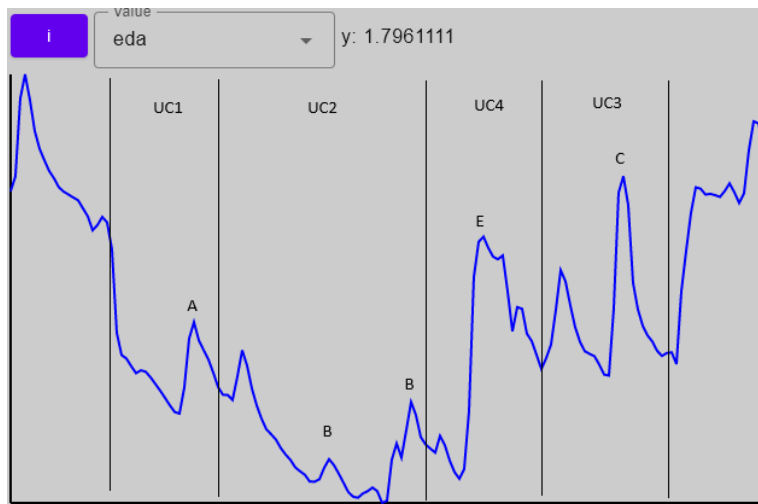


Abbildung 6.1: EDA-Graph Teilnehmer 1

BVP

Der Graph des BVP-Werts (Abbildung 6.2) zeigt eine immer ähnliche Frequenz, mit immer wieder stärkeren Ausschlägen in den Amplituden. Zwischendurch gibt es immer wieder kurze Phasen, in denen Amplitude und Frequenz sehr gering sind. Diese stimmen ungefähr mit den Zeitpunkten der starken Ausschläge des EDA-Werts überein. Weiter ist zu erkennen, dass wenn der EDA-Wert fällt, die Frequenz und Amplitude des BVP-Werts erhöht sind.

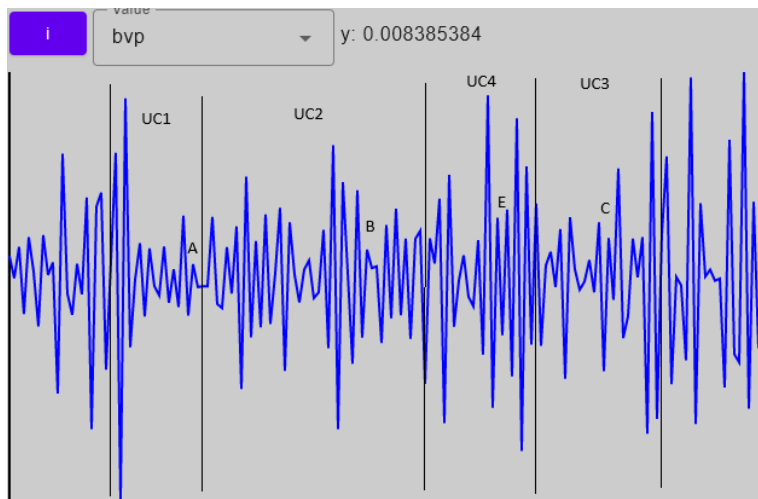


Abbildung 6.2: BVP-Graph Teilnehmer 1

Herzrate (Puls)

Zu erkennen ist, dass der Puls (Abbildung 6.3) im Durchschnitt während der Studie von circa 120 Schlägen pro Minute (bpm) auf circa 100 bpm gesunken ist. Bei genauer Betrachtung gibt es allerdings viele Ausschläge nach unten, teilweise bis zu einem Wert von 30 bpm. Diese Ausschläge hängen höchstwahrscheinlich mit Messungenauigkeiten, beispielsweise durch Bewegungen, zusammen, wie Peper et al. [18] bereits beschrieben. Diese könnten mit aufwendigen Algorithmen und Filtertechniken entfernt werden.

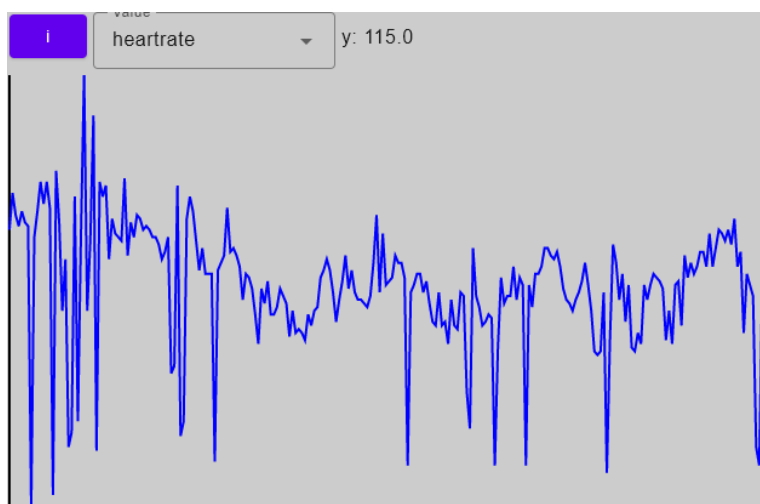


Abbildung 6.3: Puls-Graph Teilnehmer 1

Temperatur

Die Hauttemperatur des Teilnehmers stieg während der Studie stetig von 31,8 auf 31,9 Grad Celsius. Auch hier waren dennoch keine spontanen Ausschläge zu beobachten.

Zusammenfassung

Bei Teilnehmer 1 ist somit festzustellen, dass es Ausschläge in den EDA-Werten gibt, welche klar erkennbar zu den in der Tabelle 6 gezeigten Momenten während der Studie passen. Diese sind beispielsweise bei Marker A, wo der Teilnehmer erkennt, dass die Prüfung nicht bestanden wurde. Zu diesem Zeitpunkt ist er vermutlich erstaunt, was diesen Ausschlag begründen würde. Ein weiteres Beispiel wäre bei Marker E. Dort stellt der Teilnehmer fest, dass er das alte Passwort nicht kennt, was ihn wahrscheinlich ziemlich verunsichert. Beim ersten Marker B ist kaum ein Ausschlag zu erkennen. Das könnte daran liegen, dass an dieser Stelle eine Erklärung vorhanden ist und der Teilnehmer dadurch nicht so sehr verunsichert wurde. Beim zweiten Marker B hingegen ist der Ausschlag wieder stärker, weil eine erneute Fehlermeldung den Teilnehmer wahrscheinlich doch verunsichert hat. Es ist außerdem zu erkennen, dass fast immer dort, wo der EDA-Wert nach diesen Ausschlägen am Sinken ist, die Amplitude und die Frequenz des BVP-Werts

erhöht sind. Aus Puls und Temperatur lassen sich in dieser Studie keine Rückschlüsse zu den Use-Cases und den Emotionen des Teilnehmers ziehen.

6.1.2 Teilnehmer 2

Verhalten

Teilnehmer 2 kam ein paar Minuten zu spät und war deshalb leicht außer Atem und gestresst. Nach Ausfüllen der Datenschutzerklärung startete er direkt mit der Studie. Er klickte sich direkt zur Notenübersicht, wo das Ergebnis “nicht bestanden” angezeigt wurde.

Bei Use-Case 2 meldete auch er die richtige Prüfung an und bekam die Fehlermeldung. Als Reaktion darauf merkte er dann, dass dies vielleicht so gewollt ist und fragte, ob er weitermachen sollte. Der Teilnehmer hat das Feld zum Ändern der Anschrift relativ schnell gefunden und bestätigte die Änderung. Nachdem er feststellte, dass die Änderung nicht übernommen wurde, wiederholte er den Vorgang. Auch hier merkte er an, dass dies wahrscheinlich gewollt war und ging zum nächsten Use-Case weiter. Das Ändern des Passworts verlief problemlos.

Zu den Fragen sagte er, dass er die Fehlermeldungen größtenteils nicht richtig gelesen hätte, sich aber dennoch mehr Fehlermeldungen gewünscht hätte, insbesondere beim Ändern der persönlichen Daten.

Bei Teilnehmer 2 sind also zwei Dinge interessant. Zum einen können die Werte anders beziehungsweise verfälscht sein gegenüber den anderen Teilnehmern, da dieser außer Atem und gestresst in die Studie ging. Zum anderen kann der Bedarf an Erklärung bei diesem Teilnehmer auch anders sein, da er geantwortet hatte, dass er die Fehlermeldung nicht richtig gelesen hätte.

EDA

Der Verlauf des EDA-Werts von Teilnehmer 2 ist in Abbildung 6.4 abgebildet. Dabei sind wie bei Teilnehmer 1 die Zeitpunkte aus der Tabelle 6 markiert. Es ist zu erkennen, dass entgegen der Erwartung der Wert zunächst gering ist, obwohl der Teilnehmer bereits gestresst zur Studie kam. Ein Anstieg während und nach Use-Case 1 ist kaum erkennbar. Zum Start, nach Erhalt der Fehlermeldung (Marker B1) und bei erneutem Versuch (B2) sind jeweils signifikante Anstiege und Ausschläge des Werts jeweils um ein bis zwei Sekunden zeitversetzt zu erkennen. Auch nachdem keine Änderung der Anschrift (C) erkennbar war, ist ein klarer Anstieg zu erkennen. Bei Marker E ist wie bei Teilnehmer 1 ein extremer Anstieg zu erkennen, was sich wahrscheinlich auch hier auf den Mangel an Informationen zum alten Passwort und damit verbundener Unsicherheit begründen lässt.

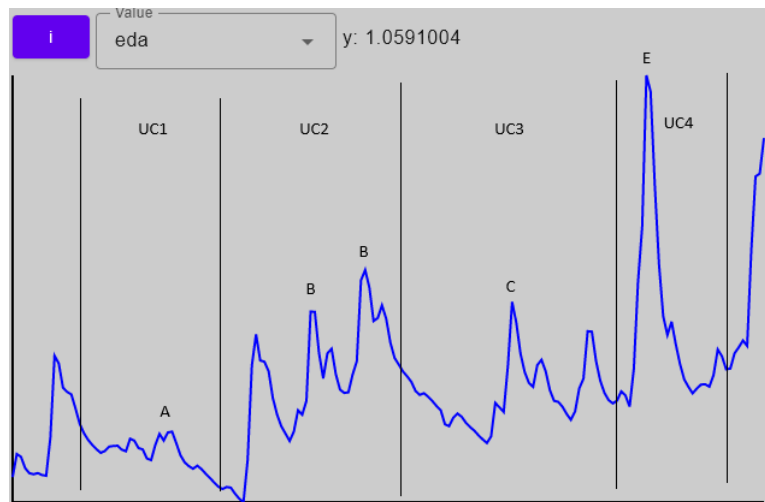


Abbildung 6.4: EDA-Graph Teilnehmer 2

BVP

Der BVP-Wert (Abbildung 6.5) von Teilnehmer 2 verläuft im ersten Drittel der Studie mit sehr niedriger Frequenz und Amplitude. Danach sind die Amplituden und Frequenzen zeitweise sehr hoch, mit kurzen Aussetzern, aber es ist nicht klar erkennbar, wie die Werte mit den Use-Cases zusammenhängen.

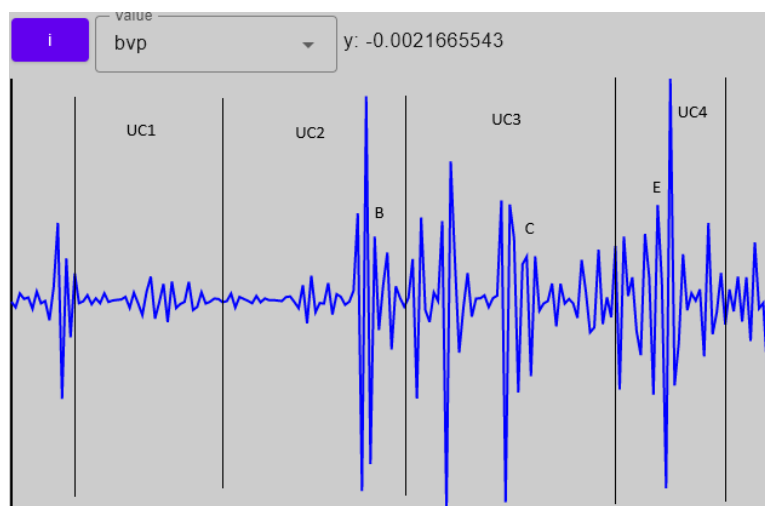


Abbildung 6.5: BVP-Graph Teilnehmer 2

Herzrate (Puls)

Der Puls von Teilnehmer 2 liegt während der Studie im Durchschnitt zwischen 80 bpm und 90 bpm. Auch hier gibt es wieder viele kurze Sprünge und hohe Amplituden von 40 bpm bis zu 120. Ein klarer Verlauf ist allerdings schwierig abzulesen.

Temperatur

Die Hauttemperatur von Teilnehmer 2 stieg während der Studie stetig von 30,8 auf 31,2 Grad Celsius, mit einem nahezu linearen Verlauf.

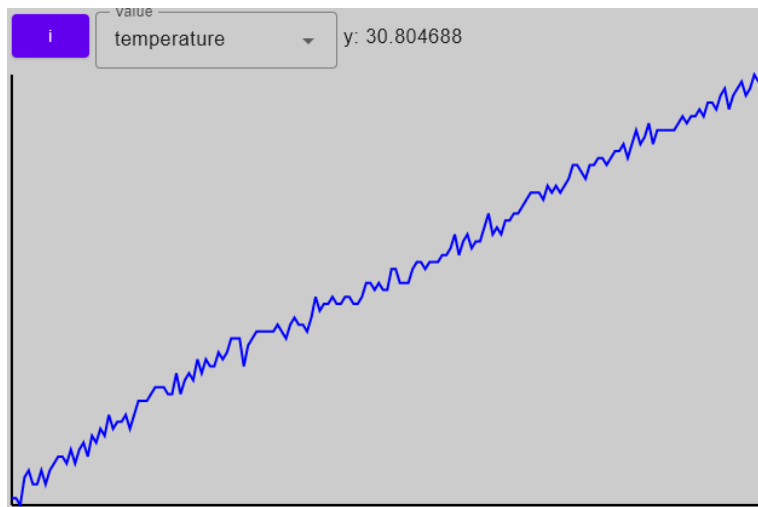


Abbildung 6.6: Temperatur-Graph Teilnehmer 2

Zusammenfassung

Zur Studie von Teilnehmer 2 ist also zu sagen, dass es auch in diesem Fall klar erkennbare Ausschläge beim EDA-Wert gibt, welche zeitlich zu Ereignissen während der Studie passen. Allerdings passen diesmal die BVP-Werte nicht dazu. Dies ist höchstwahrscheinlich auf Messfehler zurückzuführen, da im ersten Drittel nahezu kein Ausschlag zu erkennen ist, was bei einer Person mit erhöhtem Puls unwahrscheinlich ist, da auch im Ruhezustand oder beim Sport deutliche Amplituden erkennbar sein sollten [18]. Dies belegt auch der Puls, da dieser sich aus den BVP-Werten errechnet und der Verlauf bei diesem Teilnehmer viel sprunghafter und schwieriger erkennbar ist. Zur Temperatur sei gesagt, dass sie keinerlei Aussagekraft über den Verlauf der Use-Cases hat und somit den Erklärungsbedarf zulässt.

6.1.3 Teilnehmer 3**Verhalten**

Dieser Teilnehmer hat während der gesamten Studie laut seine Gedanken und Überlegungen ausgesprochen, wodurch dieser Durchlauf ein wenig länger als die anderen beiden dauerte (fünf Minuten gegenüber drei Minuten). Auch

er fand die Notenübersicht im Bereich Prüfungsübersicht und das Ergebnis der Prüfung. Dabei musste er kurz nach der richtigen Prüfung aus der Liste suchen. Bei Use-Case 2 navigierte er zunächst zu den angemeldeten Prüfungen, bis er feststellte, dass er dort keine weiteren anmelden kann. Danach fand er die Prüfungsanmeldung, meldete die richtige Prüfung an, bekam die Fehlermeldung und fragte dann, ob er einfach weiter machen solle. Beim Ändern der Anschrift merkte er an, dass es einmal “Anschrift” und einmal “Adresse” heie und dass die nderung nicht bernommen wurde. Dies habe ihn verwirrt. Das ndern des Passworts verlief auch bei diesem Teilnehmer ohne Probleme.

Die Fragen beantwortete er damit, dass die Fehlermeldung beim Anmelden der Prfung ausreichend erklrt gewesen sei und dass es sich beim ndern der Adresse er nicht sicher war, ob dies so beabsichtigt war und er sonst auf jeden Fall einer Erklrung bentigt htte.

Da der Teilnehmer sich oft verlickt hat und relativ lange brauchte, den fr den Use-Case relevanten Bereich zu finden, wren fr diesen Teilnehmer bei der Navigation wahrscheinlich mehr Erklrungen ntig gewesen. Somit knnte zu diesen Zeitpunkten mit Ausschlgen der Werte zu rechnen sein, da es Verunsicherung hervorrufen kann, wenn man das Gesuchte nicht direkt findet und sich verlickt.

EDA

Aus dem Verlauf der EDA-Werte 6.7 lassen sich bei diesem Teilnehmer wenig Aussagen treffen. Der Graph macht dabei groe Sprnge und Amplituden sind nicht klar erkennbar. Dies lsst vermuten, dass ein Fehler bei der Durchfhrung passiert sein muss. Beispielsweise knnten die Kontakte und Sensoren der Uhr nicht richtig auf dem Arm angelegen haben, sodass die Messungen fehlerhaft waren.

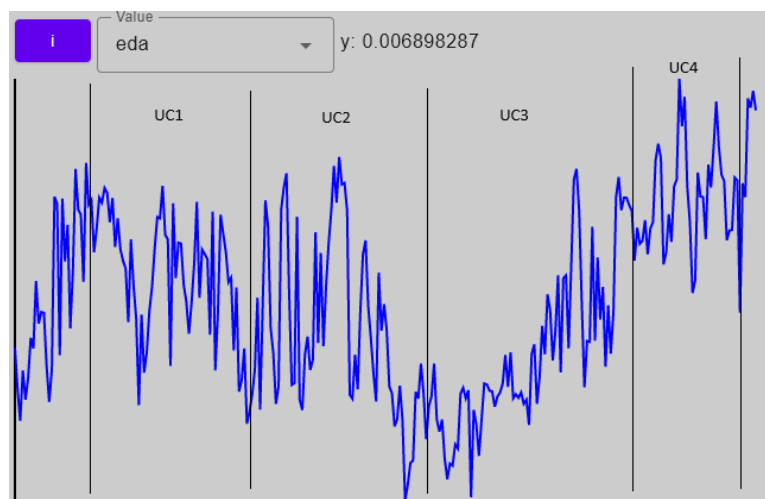


Abbildung 6.7: EDA-Graph Teilnehmer 3

BVP

Auch der BVP-Wert 6.8 lässt wenig Interpretation bezüglich der Use-Cases zu, da auch dieser scheinbar nicht richtig aufgenommen wurde. Dabei sind während der gesamten Studie Frequenz und Amplitude nahezu Null. Lediglich am Anfang und am Ende sind einige Ausschläge zu erkennen.

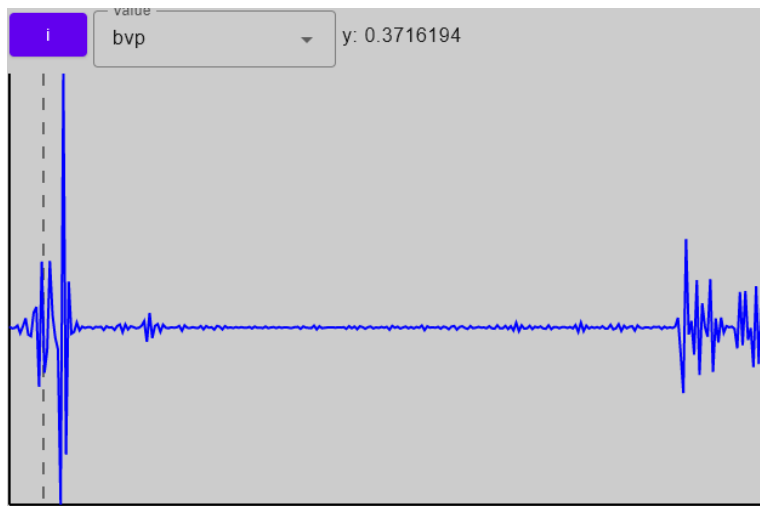


Abbildung 6.8: BVP-Graph Teilnehmer 3

Herzrate (Puls)

Der Puls 6.9 hingegen sieht ähnlich wie bei den anderen Teilnehmern aus und ist dabei ungefähr bei einem Wert von 90 bpm während der gesamten Studie.

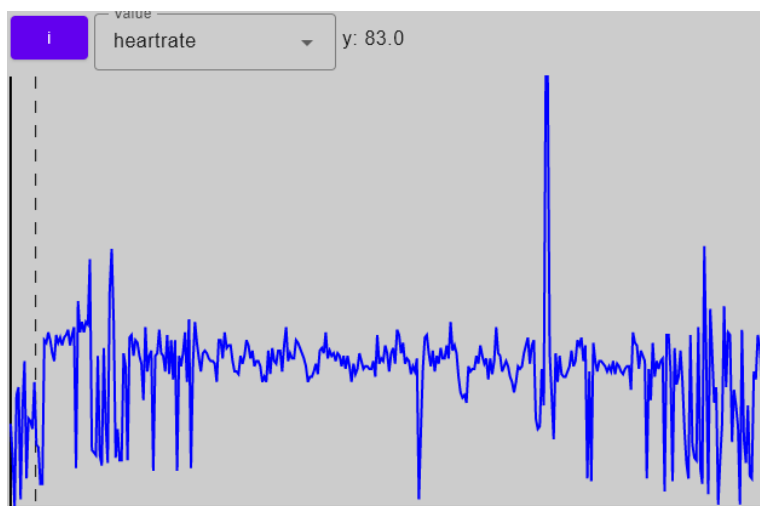


Abbildung 6.9: Puls-Graph Teilnehmer 3

Temperatur

Auch die Temperatur steigt wie bei den anderen Teilnehmern linear, in diesem Fall von 29 Grad Celsius auf 29,4 Grad Celsius.

Zusammenfassung

Zusammenfassend lässt sich zu der Studie von Teilnehmer 3 sagen, dass vermutlich ein Fehler bei der Durchführung passiert ist und somit keine Rückschlüsse in Hinsicht auf Erklärungsbedarf gezogen werden können. Für die Zukunft lässt sich allerdings sagen, dass bei weiteren Studien auf den korrekten Sitz der Uhr am Handgelenk genauer geachtet werden sollte.

6.2 Beantwortung der Forschungsfragen

6.2.1 Beantwortung von Forschungsfrage 1

Welche Parameter der Empatica Watch eignen sich zur Auswertung des Erklärungsbedarfs?

Die Ergebnisse der Studie lassen vermuten, dass es möglich ist, mit den Daten, die die Uhr sammelt, festzustellen, ob Teilnehmer Probleme bei den Use-Cases hatten. Besonders gut eignet sich dafür der EDA-Wert. Mit einer kurzen Verzögerungszeit von ein bis zwei Sekunden kann mithilfe dieses Werts ungefähr erkannt werden, an welchen Stellen der Teilnehmer Schwierigkeiten mit der Software und den Use-Cases hatte. Dies wurde bereits durch andere Studien bestätigt, wie in Kapitel 2.1 bereits erläutert. Auch kann anhand der Stärke des Ausschlags ungefähr bestimmt werden, wie stark die Emotionen des Teilnehmers sind. Ein gutes Beispiel dafür ist der Ausschlag von Use-Case 4 bei Teilnehmer 1 und 2, da dieser signifikant höher ist als die anderen Ausschläge. Somit kann die elektrodermale Aktivität dazu genutzt werden, um in solchen Studien die Emotionen einer Person zu bestimmen und daraus dann gegebenenfalls den Erklärungsbedarf abzuleiten. Der BVP-Wert ergänzt die Ergebnisse, welche aus den EDA-Werten gezogen werden konnten. Es ist erkennbar, dass die Frequenz und die Amplitude erhöht sind, wenn ein Ausschlag des EDA-Werts wieder sinkt. Dadurch lässt sich vermuten, dass der BVP-Wert eine höhere Verzögerungszeit hat als der EDA-Wert, bevor eine Emotionsänderung messbar wird. Allerdings ist es schwierig, diese Stellen in dem Graphen des BVP-Werts unabhängig von den EDA-Werten zu deuten. Somit lässt sich sagen, dass es denkbar ist, dass der BVP-Wert dazu geeignet ist, Erklärungsbedarf zu ermitteln, es aber deutlich einfacher ist, diesen parallel mit dem EDA-Wert zu analysieren und als Bestätigung der Deutungen des EDA-Werts zu betrachten. Zu der Herzrate beziehungsweise dem Puls lässt sich sagen, dass es bei den aufgezeichneten Daten eine große Varianz und große Sprünge gab. Deshalb könnte es nötig sein, die Werte zu filtern oder andere Berechnungen anzustellen, sodass der Verlauf klarer erkennbar ist. Dennoch lässt sich aus den Ergebnissen der Studie

anhand der groben Bestimmung des Pulses nur sagen, ob ein Teilnehmer während der Studie erhöhten Puls hatte und aufgeregt beziehungsweise nervös war oder nicht. Dadurch lassen sich aber eventuell die Daten von Teilnehmer aus Studien rausfiltern, weil diese durch die Nervosität weniger Aussagekraft haben. Erklärungsbedarf lässt sich aus dem Puls somit schlecht bis gar nicht ableiten, da keine kurzfristigen Änderungen erkennbar sind. Bei der Temperatur ist dies ähnlich: Es lässt sich lediglich erkennen, dass die Hauttemperatur der Teilnehmer während der Studie stieg. Dies wird damit zusammenhängen, dass sich der Sensor der Uhr nur langsam aufwärmt. Demnach ist auch die Temperatur für solche kurzweiligen Studien ungeeignet, da es keine Ausschläge und Sprünge gibt, die Rückschlüsse auf Emotionen zulassen. Anzumerken ist außerdem, dass bei einer Studie darauf geachtet werden muss, dass die Uhr korrekt am Teilnehmer befestigt ist, da sonst sehr schnell Messfehler und Ungenauigkeiten entstehen, wie bei Teilnehmer 2 und 3 erkennbar.

Abschließend lässt sich sagen, dass die Werte EDA und BVP am vielversprechendsten sind in Hinsicht auf das Ermitteln von Emotionen. Allerdings lässt sich nicht mit Sicherheit sagen, ob diese Rückschlüsse auf den Erklärungsbedarf liefern oder es anderweitige Emotionen sind. Um dies mit Sicherheit sagen zu können, wären weitere Studien notwendig.

6.2.2 Beantwortung von Forschungsfrage 2

Wie können die Daten der Empatica Watch zur Auswertung nach Erklärungsbedarf vereinfacht visualisiert werden?

Die Auswertung der Nutzerstudie hat gezeigt, dass die Daten mithilfe der entwickelten Software so dargestellt werden können, dass eine einfache, synchrone und parallele Auswertung möglich ist. Das Einlesen der Daten in die Software ist dabei per einfachem Knopfdruck nach Verschieben in den passenden Ordner möglich. Die Werte der einzelnen Parameter werden dabei so angepasst, dass es jeweils einen Wert pro Sekunde gibt und diese dann als Graph dargestellt werden. Diese Graphen werden dann nebeneinander beziehungsweise untereinander angezeigt und parallel dazu wird ein Video der Studie eingebettet. Dadurch kann eine synchrone Auswertung erfolgen, welche sowohl in den Graphen als auch im Video den gleichen Zeitpunkt einer Nutzerstudie zeigt. Diese Synchronität und Parallelität haben die vergleichsweise einfache und schnelle manuelle Auswertung ermöglicht. Wie Boucsein [2] in seinem Buch bereits erwähnte, ist es einfacher, solche Daten per Hand zu analysieren, als dies eine Software erledigen zu lassen, welche komplexe Berechnungen anstellen müsste. Somit kann mithilfe der für diese Arbeit entwickelten Software eine schnelle Erstanalyse der Daten der Empatica Watch ermöglicht werden, indem durch ein erstes Betrachten der Daten einige interessante Ausschläge erkennbar sind. Im Nachhinein können immer noch anderweitig genaue mathematische Analysen der Daten

stattfinden.

Die Ergebnisse der Auswertung mithilfe der Software sind allerdings durch den kleinen Rahmen der Nutzerstudie, also die geringe Anzahl an Teilnehmern, sowie die Kürze der Softwaretests limitiert. Dadurch kann nicht genau bestimmt werden, wie sich die Software, Arbeitsabläufe und Auswertung bei längerfristigen Studien mit höherer Teilnehmeranzahl verhalten würden.

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

An dieser Stelle seien nun die wichtigsten Ergebnisse dieser Arbeit noch einmal zusammengefasst.

Um untersuchen zu können, wie anhand von Bioparametern der Bedarf nach Erklärbarkeit beim Benutzen einer Software ermittelt werden kann, wurde im Zuge dieser Arbeit eine Software entwickelt. Diese Software sollte dazu dienen, die von einer smarten Armbanduhr gesammelten Daten, welche in binary Format vorliegen, möglichst einfach visuell darzustellen. Dabei sollte parallel zur Darstellung der Daten ein Video eingebettet werden, welches den Ablauf eines Softwaretests parallel zu den Daten zeigt. Der wichtige Punkt dabei war es, die Daten mit dem Video zu synchronisieren.

Nach der Entwicklung der Software wurde eine Nutzerstudie durchgeführt, bei welcher drei Teilnehmer eine ihnen unbekannt Software mit einigen Unvorhersehbarkeiten testen sollten, während ihre Bioparameter mit der Uhr aufgezeichnet wurden. Die gesammelten Daten wurden anschließend mithilfe der Software ausgewertet und es wurde untersucht, ob die gesammelten Daten Rückschlüsse auf den Erklärungsbedarf liefern.

Das Ergebnis der Studie war, dass zwei der insgesamt acht Parameter vielversprechende Ergebnisse in Hinsicht von Erklärungsbedarf aufwiesen. Diese beiden Werte sind die Elektrodermale Aktivität (EDA) beziehungsweise die elektrische Hautleitfähigkeit und der Blut Volumen Puls (BVP). Außerdem stellte sich die entwickelte Software als sehr geeignet zum Auswerten der Daten heraus.

7.2 Ausblick

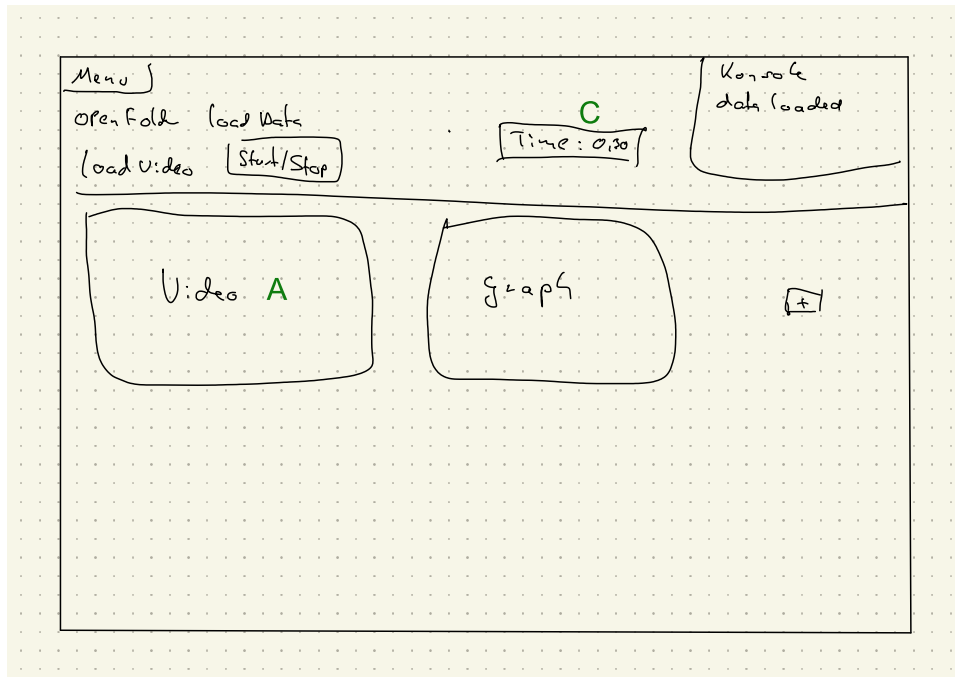
Anhand der Ergebnisse der Studie zeigt sich, dass sich mithilfe der Bioparameter und Software deuten lässt, an welchen Stellen ein Nutzer Probleme mit einer Software hat. Nun könnte in weiteren Studien untersucht werden, ob und wie sich mehr beziehungsweise andere Erklärungen auf die Ergebnisse der Daten auswirken. So könnte bestimmt werden, ob durch alternative Erklärungen die Ausschläge in den Daten minimiert oder sogar beseitigt werden können. Auch kann untersucht werden, wie sich die Daten bei unterschiedlichen Personengruppen ändern.

Wie in der Einleitung 1 bereits erwähnt, soll in zukünftigen Studien und Arbeiten untersucht werden, ob es möglich ist, mithilfe dieser Parameter und Daten der Uhr Erklärungsbedarf in Echtzeit zu ermitteln und einem Nutzer direkt eine Erklärung zu liefern, wenn dieser eine benötigt.

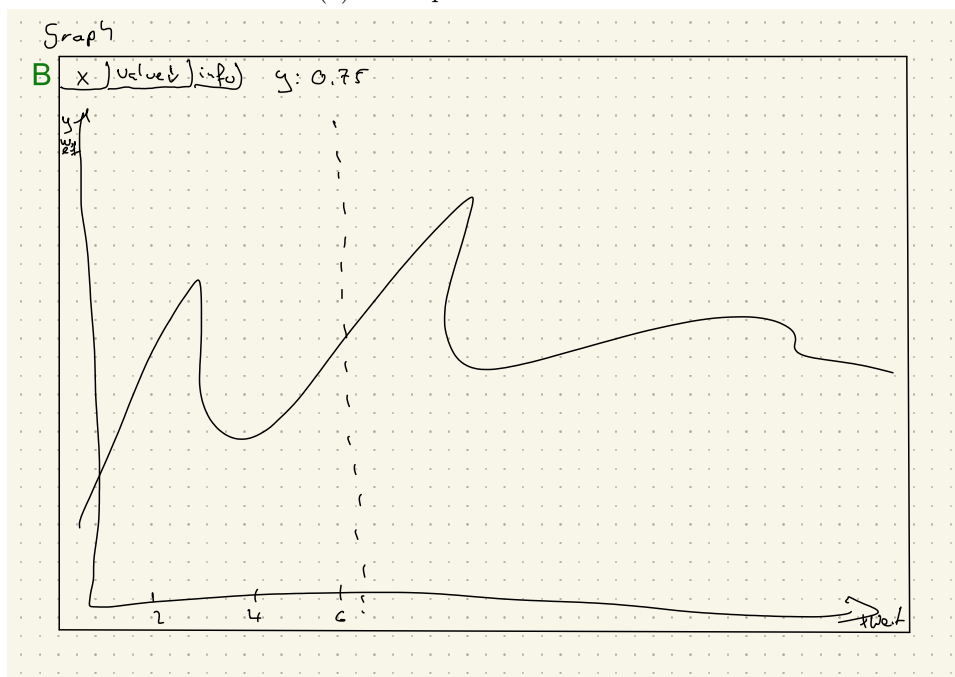
Weiter könnte untersucht werden, ob eine andere Methode der Berechnung oder das Filtern der Daten ein besseres Ergebnis für den Verlauf des Puls-Graphen liefert. Mit einem klar erkennbaren Verlauf könnte dann überprüft werden, ob sich dieser Wert vielleicht doch zur Erkennung von Erklärungsbedarf eignet.

Anhang A

Konzeptentwürfe



(a) Konzeptentwurf 2 Seite 1



(b) Konzeptentwurf 2 Seite 2

Abbildung A.1: Konzeptentwurf 2

Literaturverzeichnis

- [1] Apache. Interface GenericRecord. <https://avro.apache.org/docs/1.10.2/api/java/org/apache/avro/generic/GenericRecord.html>. letzter Zugriff: September 2023.
- [2] W. Boucsein. *Electrodermal activity*. Springer Science & Business Media, 2012.
- [3] L. Chazette, W. Brunotte, and T. Speith. Exploring explainability: A definition, a model, and a knowledge catalogue. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 197–208, 2021.
- [4] L. Chazette and K. Schneider. Explainability as a non-functional requirement: challenges and recommendations. *Requirements Engineering*, 25(4):493–514, 2020.
- [5] G. Cosoli, G. Iadarola, A. Poli, and S. Spinsante. Learning classifiers for analysis of blood volume pulse signals in iot-enabled systems. In *2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4. 0&IoT)*, pages 307–312. IEEE, 2021.
- [6] G. De Haan and A. Van Leest. Improved motion robustness of remote-ppg by using the blood volume pulse signature. *Physiological measurement*, 35(9):1913, 2014.
- [7] J. Droste, H. Deters, J. Puglisi, and J. Klünder. Designing end-user personas for explainability requirements using mixed methods research. In *2023 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, 2023.
- [8] Empatica. Embrace Care - A dedicated software ecosystem for seamless EmbracePlus data transfer and access. <https://www.empatica.com/en-eu/care/>. letzter Zugriff: September 2023.
- [9] Empatica. Embrace Plus - The world’s most advanced smartwatch for continuous health monitoring. <https://www.empatica.com/en-eu/embraceplus/>. letzter Zugriff: August 2023.

- [10] E. Eren and T. S. Navruz. Stress detection with deep learning using bvp and eda signals. In *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–7, 2022.
- [11] M. Fechner. Konzept und implementierung einer komponente zur untersuchung des erklärungsbedarfs von software. Bachelorarbeit, Leibniz Universität Hannover, 2023.
- [12] D. Girardi, F. Lanubile, N. Novielli, and A. Serebrenik. Emotions and perceived productivity of software developers at the workplace. *IEEE Transactions on Software Engineering*, 48(9):3326–3341, 2022.
- [13] JetBrains. Compose Multiplatform Develop stunning shared UIs for Android, iOS, desktop, and web. <https://www.jetbrains.com/lp/compose-multiplatform/>. letzter Zugriff: August 2023.
- [14] JetBrains. Data classes. <https://kotlinlang.org/docs/data-classes.html>. letzter Zugriff: August 2023.
- [15] A. Kushki, J. Fairley, S. Merja, G. King, and T. Chau. Comparison of blood volume pulse and skin conductance responses to mental and affective stimuli at different anatomical sites. *Physiological Measurement*, 2011.
- [16] M. A. Köhl, K. Baum, M. Langer, D. Oster, T. Speith, and D. Bohlender. Explainability as a non-functional requirement. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 363–368, 2019.
- [17] S. C. Müller and T. Fritz. Using (bio)metrics to predict code quality online. In *Proceedings of the 38th International Conference on Software Engineering, ICSE '16*, page 452–463, New York, NY, USA, 2016. Association for Computing Machinery.
- [18] E. Peper, R. Harvey, I.-M. Lin, H. Tylova, and D. Moss. Is there more to blood volume pulse than heart rate variability, respiratory sinus arrhythmia, and cardiorespiratory synchrony? *Biofeedback*, 35:54–61, 01 2007.
- [19] D. Prasanna. *Dependency injection: design patterns using spring and guice*. Simon and Schuster, 2009.
- [20] W. Prokasy. *Electrodermal activity in psychological research*. Elsevier, 2012.
- [21] H. Y. Yang, E. Tempero, and H. Melton. An empirical study into use of dependency injection in java. In *19th Australian Conference on Software Engineering (aswec 2008)*, pages 239–247. IEEE, 2008.